ISSN (Online): 2349-7084
GLOBAL IMPACT FACTOR 0.238
ISRA JIF 0.351

INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING IN RESEARCH TRENDS
VOLUME 1, ISSUE 6, DECEMBER 2014, PP 436-446

Mobile Transmission Using Rigorous Data for Wireless Sensor Networks

[1] K.MANIMALA, [2] B.RANJITH

[1] M.Tech Research Scholar, Priyadarshini Institute of Technology and Science for Women
[2] HOD-CSE, Priyadarshini Institute of Technology and Science for Women

Abstract: - Wireless Sensor Networks (WSNs) are progressively more used in data-intensive applications such as micro-climate monitor, exactitude agriculture, and audio/video supervision. A key dispute faced by data-intensive WSNs is to convey all the data generated within an application's duration to the base station regardless of the fact that antenna nodes have restricted power supplies. We recommend using low-cost not reusable mobile transmissions to concentrate the energy expenditure of data-intensive WSNs. Our advance differs from preceding work in two main aspects. First, it does not require complex motion planning of mobile nodes, so it can be implemented on a number of low-cost mobile sensor platforms. Second, we incorporate the energy exploitation due to both mobility and wireless transmissions into a holistic optimization framework. Our framework consists of three main algorithms. The first algorithm computes an optimal routing tree pretentious no nodes can move. The second algorithm improves the topology of the routing tree by tightfistedly adding new nodes exploiting mobility of the newly added nodes. The third algorithm improves the routing tree by relocating its nodes without changing its topology. This iterative algorithm converges on the optimal position for each node given the restriction that the routing tree topology does not change. We present professional distributed implementations for each algorithm that require only inadequate, confined to a small area bringing together. Because we do not automatically calculate an optimal topology, our ending routing tree is not automatically optimal. However, our simulation results show that our algorithms significantly surpass the best presented solutions.

Index Terms: Wireless sensor networks, energy optimization, mobile nodes, wireless steering

1 INTRODUCTION

WSNs have been deployed in a variety of data intensive Applications counting micro-climate and habitation monitoring [1], exactitude gardening, and audio/video supervision [2]. A controlled-size WSN can assemble up to 1 Gb/year from a genetic territory [3]. Due to the inadequate storage capability of transmitter nodes, most data must be transmitted to the base station for archiving and exploration. However, sensor nodes must activate on inadequate power equipment such as batteries or small solar panels. Therefore, a key attempt faced by data-intensive WSNs is to moderate the energy utilization of sensor nodes so that all the data generated within the lifetime of the application can be transmitted to the base station.

Several dissimilar approaches have been predictable to significantly contemplate the energy cost of WSNs by using the mobility of nodes. A robotic unit may move around the network and accumulate data from moribund nodes through one-hop or multi-hop transmissions [4], [5], [6], [7], [8]. The mobile node may serve as the base station or a "data mule" that transports data between static nodes and the base station [9], [10], [11]. Mobile nodes may also be used as transmissions [12] that promote data from resource nodes to the base station. Several faction techniques for mobile transmissions have been studied in [12], [13].

Although the effectiveness of mobility in energy conservation is demonstrated by previous studies, the following key issues have not been cooperatively addressed. First, the pressure group cost of mobile nodes is not accounted for in the total network energy consumption. Instead, mobile nodes are often unspecified to have replenished an able energy supply [7] which is not constantly practicable due to the constraints of the physical environment. Second, complex motion preparation of mobile nodes is often unspecified in accessible solutions which introduces significant design complication and industrialized costs.

In [7], [8], [14], [15], mobile nodes need to frequently calculate most favorable motion paths and change their location, their point of reference and/or speed of association. Such capabilities are usually not supported by obtainable low-cost mobile sensor platforms. For instance, Robomote [16] nodes are designed using 8-bit CPUs and small batteries that only last for about 25 minutes in full motion.In this paper, we use low-cost not reusable mobile transmissions to reduce the total energy spending of data-intensive WSNs. Different from mobile base station or data mules, mobile transmissions do not transport data; as a substitute, they move to different places and then remain motionless to promote data along the paths from the sources to the base station. Thus, the communication delays can be considerably concentrated compared with using mobile sinks or data mules. Moreover, each mobile node performs a single relocation unlike other approaches which require repeated relocations.

Our approach is provoked by the current state of mobile antenna platform knowledge. On the one hand, frequent low-cost mobile sensor prototypes such as Robomote [16], Khepera [17], and FIRA [18] are now available. Their industrialized cost is similar to that of representative stationary antenna proposals. As a result, they can be especially organized in a network and used in a not reusable manner. Our approach takes benefit of this competence by presuming that we have a large number of mobile transmission nodes. On the other hand, due to low manufacturing cost, accessible mobile sensor platforms are naturally powered by batteries and only accomplished of restricted mobility. Consistent with this constraint, our advance only needs one-shot substitution to designated positions after deployment. Compared with our approach, presented mobility approaches typically assume a small number of powerful mobile nodes, which does not exploit the availability of many low-cost mobile nodes.

Work Contribution

We make the following assistance in this paper.

(1) We originate the problem of Optimal Mobile Transmission Configuration (OMRC) in data-intensive WSNs. Our objective of energy management is holistic in that the total energy infatuated by both mobility of transmissions and wireless transmissions is decreased, which is in distinction to accessible mobility approaches that only diminish the transmission energy consumption. The exchange in energy consumption between mobility and transmission is oppressed by configuring the positions of mobile transmissions.

(2) We study the effect of the preliminary configuration on the final result. We compare different initial tree building strategies and recommend an optimal tree manufacture approach for static nodes with no mobility.

(3) We develop two algorithms that iteratively refine the configuration of mobile transmissions. The first improves the tree topology by adding new nodes. It is not assured to find the optimal topology. The second improves the routing tree by relocating nodes without changing the tree topology. It converges to the optimal node positions for the given topology. Our algorithms have efficient distributed implementations that require only limited, localized synchronization.

(4) We conduct extensive simulations based on realistic energy models obtained from existing mobile and static sensor platforms. Our results show that our algorithms can reduce energy consumption by up to 45% compared to the best existing solutions.

2. RELATED WORK

We review three different approaches, mobile base stations, data mules, and mobile transmissions that use mobility to reduce energy expenditure in wireless sensor networks. A mobile base station moves around the network and accumulates data from the nodes. In some work, all nodes are always performing multiple hop transmissions to the base station, and the purpose is to rotate which nodes are close to the base station in order to control the transmission load [4], [5], [6]. In other work, nodes only transmit to the base station when it is close to them (or a neighbor). The goal is to calculate a mobility path to collect data from visited nodes before those nodes suffer buffer overflows [7], [8], [14], [15]. In [8], [19], [20], 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. These approaches incur high latencies due to the low to moderate speed, e.g. 0.1-1 m/s [14], [16], of mobile base stations.

Data mules are similar to the second form of mobile base stations [9], [10], [11]. They pick up data from the sensors and transport it to the sink. In [21], the standard theoretical model for energy consumption in mobile ad hoc networks considers the distances of communication partners and the amount of data transmission, namely the flow cost model of [17].The quadratic increase of this model is motivated by the path loss in radio communications, which can be approximated for a fixed scenario by O $(d\alpha)$, where d is the distance and α is the path loss exponent. This approximation has been established by extensive Tests in several real environments leading to different Path loss exponents for different environments. In [19] and [13], an additive constant is added to this term to take into account the signal processing before sending and after receiving messages.

In the third approach, the network consists of mobile communicate nodes along with static base station and data sources. Transmission nodes do not transport data; in-stead, they move to different locations to decrease the transmission costs. We use the mobile transmission approach in this work. Goldenberg et al. [13] showed that an iterative mobility algorithm where each transmission node moves to the midpoint of its neighbors converges on the optimal solution for a single routing path. However, they do not account for the cost of moving the communicate nodes. In [22], mobile nodes decide to move only when moving is useful, but the only position considered is the midpoint of neighbors.

Unlike mobile base stations and data mules, our OMRC problem considers the energy consumption of both mobility and transmission. Our approach also relocates each mobile transmission only once immediately after deployment. Unlike previous mobile transmission schemes [13] and [22], we consider all possible locations as possible target locations for a mobile node instead of just the midpoint of its neighbors. Mobility has been comprehensively studied in sensor network and robotics applications which consider only mobility costs but not communication costs. For example, in [23], the authors propose approximation algorithms to minimize maximum and total movement of the mobile nodes such that the network becomes connected. In [24], the authors propose an optimal algorithm to bridge the gap between two static nodes by moving nearby mobile nodes along the line connecting the static points while also minimizing the total/maximum distance moved. In [25], [26], the authors suggest algorithms to find motion paths for robots to investigate the area and perform a certain task while taking into contemplation the energy available at each robot. These problems ignore announcement costs which add an increased difficulty to OMRC, and accordingly their results are not applicable.
Our OMRC problem is somewhat similar to a number of graph theory problems such as the Steiner tree problem [27], [28], [29] and the facility location problem [30], [31]. However, because the OMRC cost function is fundamentally different from the cost function for these other problems, existing solutions to these problems cannot be applied directly and do not provide good solutions to OMRC. For example, there is no obvious way to include mobility costs in the Steiner tree problem.

3. PROBLEM DEFINITION
3.1Energy Consumption Models

Nodes consume energy during communication, computation, and movement, but communication and mobility energy consumption are the major cause of battery drainage. Radios consume considerable energy even in an idle listening state, but the idle hearing time of radios can be significantly condensed by a number of sleep scheduling protocols [32]. In this work, we focus on reducing the total energy expenditure due to transmissions and mobility. Such a holistic objective of energy conservation is motivated by the fact that mobile transmissions act the same as static forwarding nodes after movement.

For mobility, we consider wheeled sensor nodes with differential drives such as Khepera [17], Robomote [16] and FIRA [18]. This type of node usually has two wheels, each controlled by independent engines. We adopt the distance proportional energy consumption model which is appropriate for this kind of node [33]. The energy EM (d) consumed by moving a distance d is modeled as: **EM (d) = KD**
The value of the parameter k depends on the speed of the node. In general, there is an optimal speed at which k is lowest. In [33], 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 [33].To model the energy consumed through transmissions, we investigate the empirical results obtained by two radios CC2420 [34] and CC1000 [35] that are widely used on existing sensor network platforms. For CC2420, the authors of [36] studied the transmission power level needed for transmitting packets reliably (e.g., over 95% packet reception ratio) over different distances. Let ET (d) be the energy consumed to transmit reliably over distance d. It can be modeled as : **ET (d) = m (a + bd2)**
Where m is the number of bits transmitted and a and b are constants depending on the environment. We now discuss the instantiation of the above model for both CC2420 and CC1000 radio platforms. In an outdoor environment, for received signal strength of -80 dbm (which corresponds to a packet reception ratio higher than 95%), we obtain a = 0.6 × 10−7J/bit and b = 4 × 10−10Jm−2/bit from the measurements on CC2420 in [36]. This model is consistent with the theoretical analysis discussed in [37]. We also consider the energy needed by CC1000 to output the same levels. We get lower consumption parameters:

a = 0.3 × 10−7J/bit and b = 2 × 10−10Jm−2/bit.

We will see in Section 5 that we maintain this high packet reception ratio throughout our algorithm. We note that although the mobility parameter k is roughly 1010 times larger than the transmission parameter b, the transmissions move only once whereas large amounts of data are transmitted. For large enough data chunk sizes, the savings in energy transmission costs compensates for the energy expended to move the nodes resulting in a decrease in total energy consumed.

3.3 Problem Formulation

In our definitions, we assume that all movements are completed before any transmissions begin. We also assume there are no obstacles that affect mobility or transmissions. In this case, as we show in Section 4.2, the distance moved by a mobile transmission is no more than the distance between its starting position and its corresponding position in the evenly spaced configuration which often leads to a short delay in mobile transmission relocation. Furthermore, we assume that all mobile nodes know their locations either by GPS units mounted on them or a localization service in the network. We focus on the case where all nodes are in a 2-dimensional plane ℜ2, but the results apply to ℜ3 and other metric spaces.

Our problem can be explained as follows. Given a network containing one or more static source nodes that store data gathered by other nodes, a number of mobile transmission nodes and a static sink, we want to find a directed routing tree from the sources to the sink as well as the optimal positions of the mobile nodes in the tree in order to minimize the total energy consumed by transmitting data from the source(s) to the sink and the energy consumed by relocating the mobile transmissions. The source nodes in our problem formulation serve as storage points which cache the data gathered by other nodes and periodically transmit to the sink, in response to user queries. Such network architecture is consistent with the design of storage-centric sensor networks [38]. Our problem formulation also considers the initial positions of nodes and the amount of data that needs to be transmitted from each storage node to the sink. The formal definition of the problem is given below.

Definition 1: (Optimal Mobile Transmission Configuration):

Input Instance: S, a list of n nodes (s1, . . . , sn) in the network; O, a list of n locations (o1, . . . , on) where oi is the initial position of node si for 1 ≤ i ≤ n; Ssources, a subset of S representing the source nodes; r, a node in S, representing the single sink;
Msources = {Mi ∣ si ∈ Ssources}, a set of data chunk sizes for all sources in Ssources;

4 CENTRALIZED SOLUTION

4.1 Energy Optimization Framework

The Optimal Mobile Transmission Configuration (OMRC) problem is challenging because of the dependence of the solution on multiple factors such as the routing tree topology and the amount of data transferred through each link. For example, when transferring little data, the optimal configuration is to use only some transmission nodes at their original positions. As the amount of data transferred increases, three changes occur: the topology may change by adding new transmission nodes, the topology may change by changing which edges are used, and the transmission nodes may move closer together. In many cases, we may have restrictions such as no mobility for certain transmission nodes or we must use a fixed routing tree. These constraints affect the optimal configuration.

We illustrate how the optimal configuration depends on the amount of data to transfer using the example from Fig. 2a. When there is very little data to transfer, the optimal routing tree Ta depicted in Fig. 2a uses only some of the transmission nodes in their original positions. When the amount of data to transfer from s1 and s2 increases to 15 MB, the transmission nodes in tree Ta move to their corresponding positions in tree Tb of Fig. 2b but the topology does not change. When the amount of data to transfer from s1 and s2 is between 25 and 60 MB, the optimal routing tree has a different topology as shown in Fig. 2c. For even larger messages, new trees with even more nodes included are optimal. For example, when the amount of data to be transferred is between 100 and 150 MB, the optimal tree is depicted in Fig. 2d. No existing tree construction strategy handles all these cases. For example, the minimum spanning tree that includes all network nodes has two fundamental problems. It will typically include unneeded nodes, and it typically creates non-optimal topologies as it focuses only on the current location of nodes as opposed to where nodes may move to.We now present a centralized approach to solve OMRC that breaks the problem into three distinct steps: initial tree construction, node insertions, and tree optimization. For each step, we present an algorithm to solve the corresponding sub problem. Our algorithm for initial tree construction is optimal for the static environment where nodes cannot move. However, we can effectively apply the later algorithms if we must start with a different topology. Our greedy heuristic for improving the routing tree topology by adding nodes exploits the mobility of the newly added nodes. Our tree optimization algorithm improves the routing tree by relocating its nodes with-out changing its topology. This iterative algorithm converges on the optimal position for each node given the constraint that the routing tree topology is fixed. Our node insertion and tree optimization algorithms use the LocalP os algorithm we propose in Fig. 3 that optimally solves the simplest case (see Section 4.2) of the mobile transmission configuration problem where there is a single source, a single sink, and a single transmission node. Our approach is not guaranteed to produce an optimal configuration because we do not necessarily find the optimal topology, but our simulation results show that it performs well.

4.2 Base Case

Before presenting our algorithm for OMRC, we revisit the example of Section 3.2 as it represents the simplest possible base case of the problem in which the network consists of one source si−1, one mobile transmission node si and one sink si+1. In this section, we calculate the optimal position for the transmission node. We use the following notation. In $\mathfrak{R}^2$, let the original position of a node sj be oj = (pj, qj), and let uj = (xj, yj) its final position in configuration U. According to our energy models, the total transmission and movement energy cost incurred by the mobile transmission node si is

Ci (U) = k‖ui − oi‖ + am + b‖ui+1 − ui‖2m

We also define

Ci (U) = ci(U) + am + b‖ui − ui−1‖2m

4.3 Static Tree Construction

Different applications may apply different constraints on the routing tree. When only optimizing energy consumption, a shortest path strategy (as discussed below) yields an optimal routing tree given no mobility of nodes. However, in some applications, we do not have the freedom of selecting the routes. Instead, they are predetermined according to some other factors (such as delay, capacity, etc). In other less stringent cases, we may be able to update the given routes provided we keep the main structure of the tree. Depending on the route constraints dictated by the application, we start our solution at different phases of the algorithm. In the unrestricted case, we start at the first step of constructing the tree. When the given tree must be loosely preserved, we start with the transmission insertion step. Finally, with fixed routes.We apply directly our tree optimization algorithm. Our simulations (Section 8) show that our approach outperforms existing approaches for all these cases. We construct the tree for our starting configura-tion using a shortest path strategy. We first define a weight function w specific to our communication energy model. For each pair of nodes si and sj in the network, we define the weight of edge sisj as: w (si, sj) = a + b‖oi − oj ‖2 where oi and oj are the original positions of nodes si and sj and a and b are the energy parameters discussed in Section 3.1. We observe that using this weight Function, the optimal tree in a static environment coincides with the shortest path tree rooted at the sink. So we apply Dijkstra’s shortest path algorithm starting at the sink to all the source nodes to obtain our initial topology.

```
function LOCALPOS($o_i, u_i, u_{i-1}, u_{i+1}$)
    ▷ Consider case $s_i$ moves right
    valid ← FALSE;
    $x_i \leftarrow \frac{1}{2}(x_{i-1} + x_{i+1}) - Y_i$;
    if $x_i > p_i$ then
        valid ← TRUE;
    else
        ▷ Consider case $s_i$ moves left
        $x_i \leftarrow \frac{1}{2}(x_{i-1} + x_{i+1}) + Y_i$
        if $x_i < p_i$ then
            valid ← TRUE;
        end if
    end if
    ▷ Record if new position is different from previous one
    if valid then
        $y_i \leftarrow \frac{(x_{i-1}+x_{i+1}-2p_i)}{(y_{i-1}+y_{i+1}-2q_i)}(x_i - p_i) + q_i$;

        $u'_i = (x_i, y_i)$;

        if $\|u'_i - u_i\| >$ threshold then
            return ($u'_i$, TRUE);
        end if
    end if
    ▷ not beneficial to move, stay at original position
    return ($o_i$, FALSE);
end function
```

Fig. 3. Algorithm to compute the optimal position of a transmission node that receives data from a single node and transmits the data to a single node.

4.4 Node Insertion

We improve the routing tree by greedily adding nodes to the routing tree exploiting the mobility of the inserted nodes. For each node sout that is not in the tree and each tree edge sisj, we compute the reduction (or increase) in the total cost along with the optimal position of sout if sout joins the tree such that data is routed from si to sout to sj instead of directly from si to sj using the LocalPos algorithm described in Fig. 3. We repeatedly insert the outside node with the highest reduction value modifying the topology to include the selected node at its optimal position, though the node will not actually move until the completion of the tree optimization phase. After each node insertion occurs, we compute the reduction in total cost and optimal position for each remaining outside node for the two newly added edges (and remove this information for the edge that no longer exists in the tree). At the end of this step, the topology of the routing tree is fixed and its mobile nodes can start the tree optimization phase to relocate to their optimal positions.

5 TREE OPTIMIZATION

In this section, we consider the sub problem of finding the optimal positions of transmission nodes for a routing tree given that the topology is rigid. We assume the topology is a directed tree in which the leaves are sources and the root is the sink. We also assume that separate messages cannot be compressed or merged; that is, if two distinct messages of lengths m1 and m2 use the same link (si, sj) on the path from a source to a sink, the total number of bits that must traverse link (si, sj) is m1 + m2.First, we extend the base case solution of Section 4.2 to handle multiple flows passing through a mobile transmission node. Then, we propose an iterative algorithm that uses the solution for this base case to compute the new positions of the transmission nodes in the routing tree. We also show that this algorithm converges to the optimal solution for the given tree given the topology is fixed.

5.1Extended Base Case

Before we describe our optimal algorithm for this problem, we expand the solution to the base case existing in Section 4.2 to the more general multiple flow traffic patterns. The network now consists of multiple sources, one transmission node and one sink such that data is transmitted from each source to the transmission node and then to the sink. We modify our solution as follows. Let si be the mobile transmission node, S (si) the set of source nodes transmitting to si and sdi the sink collecting nodes from si. The cost incurred by si in this configuration U is:

ci(U) = k‖ui − oi‖ + ami + bmi‖ud − ui‖2

where mi is the total amount of data that si transmits to sdi. Similar to the single source base case, we define X

Ci(U) = ci(U) + aml + b‖ui − ul‖2ml
sl∈S(si)

```
Procedure OPTIMALPOSITIONS ($U^0$)
converged ← false;
j ← 0; repeat
Anymove ← false; j ← j + 1;
⊳ Start an even iteration followed by an odd iteration for idx = 2 to 3 do
For i = idx to n by 2 do
($u^j_i$ , moved) ← LOCALPOS($o_i$, S($s_i$), $s^d_i$);
anymove ← anymove OR moved
End for end for
Converged ← NOT anymove until converged
End procedure
```

5.2 Optimization Algorithm

We propose a simple iterative approach to compute the optimal position ui for each node si. We define the following notations. Let uji = (xji , yij) be the position of node si after the jth iteration of our algorithm for j ≥ 0 and U j = (uj1, . . . , ujn) the computed configuration of nodes s1 through sn after j iterations. We define u0i = oi. Note that the mobile transmission nodes do not move until the final positions are computed.

Our algorithm starts by an odd/even labeling step followed by a weighting step. To obtain consistent labels for nodes, we start the labeling process from the root using a breadth first traversal of the tree. The root gets labeled as even. Each of its children gets labeled as odd. Each subsequent child is then given the opposite label of its parent. We define mi, the weight of a node si, to be the sum of message lengths over all paths passing through si. This computation starts from the sources or leaves of our routing tree. Initially, we know mi = Mi for each source leaf node si. For each intermediate node si, we compute its weight as the sum of the weights of its children.

Once each node gets a weight and a label, we start our iterative scheme. In odd iterations j, the algorithm computes a position uji for each odd-labeled node si that minimizes Ci (U j) assuming that uji−1 = uji−−11 and uji+1= uji+1−1; that is, node si's even numbered neighboring nodes remain in place in configuration U j. In even-numbered iterations, the controller does the same for even-labeled nodes. The algorithm behaves this way because the optimization of uji requires a fixed location for the child nodes and the parent of si. By alternating between optimizing for odd and even labeled nodes, the algorithm guarantees that the node si is always making progress towards the optimal position ui. Our iterative algorithm is shown in Fig. 4

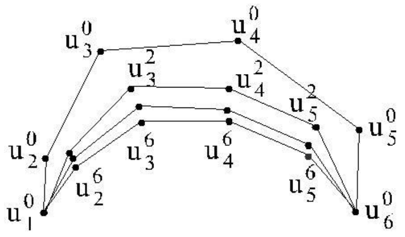

Fig. 1. Convergence of iterative approach to the optimal solution.

Each line shows the configuration obtained after 2 iterations. The optimal configuration is reached after 6 iterations.Fig. 1 shows an example of an optimal configuration for a simple tree with one source node. Nodes start at configuration U 0. In the first iteration, odd nodes (s3 and s5) moved to their new positions (u13, u 15) computed based on the current location of their (even) neighbors (u02, u04, u06). In the second iteration, only even nodes (s2 and s4) moved to their new positions (u22, u24) computed based on the current location of their (odd) neighbors (u11, u13, u 15). Since s3 and s5 did not move, their position at the end of this iteration remains the same, so u13 = u23 and u15 = u25. In this example, nodes did two more sets of iterations, and finally converged to the optimal solution shown by configuration U 6.

Even though configurations change with every iteration, nodes only move after the final positions have been computed. So each node follows a straight line to its final destination. As the data size increases, nodes in the optimal configuration get more evenly spaced. In fact, in any given configuration, the maximum distance traveled by a node is bounded by the distance between its starting position and its final position in the evenly spaced configuration.

The above example shows another property of our algorithm. When a node si moves and its neighbors (si−1 and si+1) remain in place, it moves in the direction of the midpoint of si−1si+1. This results in a reduction in the length of one of the transmission links. The other may increase in length but will never exceed the new length of the first link. This remains valid for multiple children case. So in any configuration U i+1, the length of the largest link is at most the length of the largest link in the previous configuration U i. So if we start with a route along good quality links, this quality will be preserved in the optimal configuration (and throughout intermediate configurations).

6. DISTRIBUTED ALGORITHMS

Our solutions to the three sub problems assume a centralized scheme in which one node has full knowledge of the network including which nodes are on the transmission paths to each source, the original physical position oi of each node si, and the total message length m to be sent from each source. Whereas the centralized algorithm computes the optimal static tree and the optimal position of each node in the restructured tree, it incurs prohibitively high overhead in large-scale networks. We now present a distributed and decentralized version of each of our algorithm. We modify the first phase, the tree construction phase, to use a fully distributed routing algorithm. We pick greedy geographic routing since it does not require global knowledge of the network although any algorithm with such property can be used.

After a routing tree is constructed, the tree restructuring phase begins. Network nodes outside the tree broadcast their availability (as NODE IN RANGE message) to tree nodes within their communication range and wait for responses for a period of time Tw.Similarly, tree nodes enter a listening phase Tu. During that period, tree nodes receive messages of different types (NODE IN RANGE, OFFER, . . .). Each tree node that receives one or more NODE IN RANGE message responds to the sender by giving it its location information and its parent's location information. Each non-tree node so that receives location information from a tree node si during Tw computes the reduction in cost if it joins the tree as parent of si and adds si to a list of candidates. At the end of Tw, the non-tree node selects from the candidate list the node that results in the largest reduction and sends it an offer. It also sends the tree node with the second largest reduction a POTENTIAL OFFER message. At the end of Tu, each tree node vt that collected one or more offers and potential offers operates as follows. If vt’s best potential offer exceeds its best offer by a certain threshold B and vt has not already waited R rounds, vt waits rather than accepting its best offer in the hopes that its best potential offer will become an actual offer in another round. By waiting, it sends everyone a REJECT OFFER, restarts the listening phase, and records that it has waited another round. Otherwise, vt accepts its best offer by responding to its sender p with an ACCEPT OFFER message and to the remaining nodes with a REJECT OFFER message. It then updates its parent in the tree to p, resets Tu and starts the listening phase again.

A non-tree node p that receives an ACCEPT OFFER message moves to the corresponding local optimal location and joins the tree. It becomes a tree node and enters the listening phase. On the other hand, if p does not receive an ACCEPT OFFER, p repeats the process by broadcasting its availability again and resetting Tw. We note that values in p’s candidate list cannot be reused to extend offers to old tree nodes since those tree nodes could have a new parent at this point in time. When the second phase ends, any remaining non-tree nodes stop processing whereas tree nodes enter the tree optimization phase. Fig. 6 shows the algorithm executed by each tree node.

Giving tree nodes the ability to wait before accepting an offer increases the chances of using mobile transmission nodes to their full potential. For example, consider a scenario where several mobile transmission nodes can greatly improve the capacities of several tree links but are all closest to one specific link. They will all send offers to the same tree node while the rest of the tree nodes in their proximity will receive modest offers from more distant mobile nodes. If the tree nodes cannot wait, they will be forced to accept a reserved offer and the mobile nodes will either remain unused or they will help more distant tree nodes where their impact is reduced since they use up more energy to get to their new location.

The centralized tree optimization algorithm can be transformed into a distributed algorithm in a natural way. The key observation is that computing each uji for node si only depends on the current position of si’s neighbors in the tree (children and parent), nodes that si normally communicates with for data transfers. Thus, si can perform this computation. The distributed implementation proceeds as follows. First, there is a setup process where the sender s1 sends a discover message that ends with the receiver sn; the two purposes of this message are (1) to assign a label of odd or even to each node si and (2) for each node si to learn the current positions of its neighbors. A node si sends its current position to node sj when acknowledging receipt of the discover message. Second, there is a distributed process by which the nodes compute their transmission positions. We make each iteration of the basic algorithm a “round”, though there does not need to be explicit synchronization. In odd rounds, each odd node computes its locally optimal position and transmits this new position to its neighbors. In even rounds, each even node does the same.

6.1 Centralized Algorithm

We first show the benefit of exploiting the mobility of transmission nodes by computing the average static energy consumption ratio of TREE+INS+FO for all data chunk sizes for each of our three tree building strategies PB, HB, and GG as shown in Fig. 7. For all three initial tree strategies, we see that the average static energy consumption ratio drops quickly as the data chunk size increases. For HB and GG, the average static energy consumption ratio starts out higher than 100% because P B(I), the optimal tree for the static case, is roughly 37% lower than HB(I) and GG(I) for any of our input instances. Even given this initial disadvantage of a poor starting tree from an energy consumption perspective, we see that the average static energy consumption ratios of HB+INS+FO and GG+INS+FO drop below 100% for data chunk sizes of 12 MB and 15 MB, respectively. As the data chunk size increases further, HB+INS+FO and GG+INS+FO achieve average static energy consumption ratios of 75% and 60% for data chunk sizes of 60 MB and 150MB, respectively. The results for PB+INS+FO are even better because we start with the optimal tree for the static case. Thus, the average static energy consumption ratio for PB+INS+FO is always below 100% and reaches 55% for 150 MB.We now evaluate the benefit achieved by our optimizations FO, INS, and INS+FO for each of our tree building strategies PB, HB, and GG. We note that in this set of simulations, we used our centralized improvement schemes with the distributed tree building approach GG. The purpose is to test the limits of our optimizations given a non-optimal starting tree. A fully distributed setup is studied later in this section.

6.2 Distributed Algorithm

We now evaluate how well our distributed implementation works. Our initial tree is the greedy geographic tree GG. We consider four optimizations: the centralized implementation of INS+FO, the distributed implementation of just FO, the distributed implementation of just INS, and the distributed implementation of INS followed by the distributed implementation of FO. For the distributed implementation of INS, we set parameter B to 10% (a potential offer must be 10% better than the best actual offer to cause a node to wait). Fig. 12 shows the average reduction ratio of each of these optimizations. The average reduction ratio for distributed INS+FO starts at 20% for small data chunk sizes, reaches 30% for data chunk sizes around 20MB, and exceeds 40% for data chunk sizes larger than 75MB. The gap between the average reduction ratio for centralized INS+FO and distributed INS+FO starts at roughly 5% for small data chunk sizes and increases to roughly 15% for large data chunk sizes. This gap is due to the lack of global information when performing the insertion step. Expensive links in the tree that do not have nearby transmission nodes are not able to communicate with further but available transmission nodes whose help is only offered to cheaper but nearby links. This problem is exacerbated as the data chunk size increases. We varied values for B between 10% and 50% and for R between 1 and 3. For all combinations of B and R that we tested, we obtained similar results to those of Fig. 12. As in the centralized case, distributed INS is more effective than distributed FO. However, doing both distributed optimizations does result in roughly a 10% improvement compared to only doing the distributed INS optimization for most data chunk sizes.

Similar to the centralized implementation, we observe a slow reduction in the improvement ratio as the number of sources increases for k = 2 and 4 (Fig. 11). For cheaper mobility cost (k = 1), the difference in improvement ratios increases at a faster rate and reaches 9% as the number of sources increases from 2 to 20. This is because when mobility is cheaper, in an optimal setting, nodes can move over longer distances to help exclusive links. However, as we mentioned earlier, in a distributed setting, mobile nodes are not aware of those distant expensive edges. Moreover, as the number of sources increases, the number of mobile nodes available to help decreases. Both factors combined make the distributed implementation slightly less effective for a high number of sources.

7. CONCLUSION

In this paper, we proposed a holistic approach to minimize the total energy consumed by both mobility of transmissions and wireless transmissions. Most previous work ignored the energy consumed by moving mobile transmissions. When we model both sources of energy consumption, the optimal position of a node that receives data from one or multiple neighbors and transmits it to a single parent is not the midpoint of its neighbors; instead, it converges to this position as the amount of data transmitted goes to infinity. Ideally, we start with the optimal initial routing tree in a static environment where no nodes can move. However, our approach can work with less optimal initial configurations including one generated using only local information such as greedy geographic routing. Our approach improves the initial configuration using two iterative schemes. The first inserts new nodes into the tree. The second computes the optimal positions of transmission nodes in the tree given a fixed topology. This algorithm is appropriate for a variety of data-intensive wire-less sensor networks. It allows some nodes to move while others do not because any local improvement for a given mobile transmission is a global improvement. This allows us to potentially extend our approach to handle additional constraints on individual nodes such as low energy levels or mobility restrictions due to application requirements. Our approach can be implemented in a centralized or distributed fashion. Our simulations show it substantially reduces the energy consumption by up to 45%.

REFERENCES

[1] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and
D. Culler, "An analysis of a large scale habitat monitoring application," in SenSys, 2004.

[2] L. Luo, Q. Cao, C. Huang, T. F. Abdelzaher, J. A. Stankovic, and M. Ward, "Enviromic: Towards cooperative storage and retrieval in audio sensor networks," in ICDCS, 2007, p. 34.

[3] D. Ganesan, B. Greenstein, D. Perelyubskiy, D. Estrin, and
J. Heidemann, "An evaluation of multi-resolution storage for sensor networks," in SenSys, 2003.

[4] S. R. Gandham, M. Dawande, R. Prakash, and S. Venkatesan, "Energy efficient schemes for wireless sensor networks with multiple mobile base stations," in Globecom, 2003.

[5] J. Luo and J.-P. Hubaux, "Joint mobility and routing for life-time elongation in wireless sensor networks," in INFOCOM, 2005.

[6] Z. M. Wang, S. Basagni, E. Melachrinoudis, and C. Petrioli, "Exploiting sink mobility for maximizing sensor networks lifetime," in HICSS, 2005.

[7] A. Kansal, D. D. Jea, D. Estrin, and M. B. Srivastava, "Con-trollably mobile infrastructure for low energy embedded net-works," IEEE Transactions on Mobile Computing, vol. 5, pp. 958–973, 2006.

[8] 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.

[9] D. Jea, A. A. Somasundara, and M. B. Srivastava, "Multiple controlled mobile elements (data mules) for data collection in sensor networks," in DCOSS, 2005.

[10] R. Shah, S. Roy, S. Jain, and W. Brunette, "Data mules: Modeling a three-tier architecture for sparse sensor networks," in IEEE SNPA Workshop, 2003.

[11] 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, pp. 327–339, 2006.

[12] W. Wang, V. Srinivasan, and K.-C. Chua, "Using mobile re-lays to prolong the lifetime of wireless sensor networks," in MobiCom, 2005.

[13] D. K. Goldenberg, J. Lin, and A. S. Morse, "Towards mobility as a network control primitive," in MobiHoc, 2004, pp. 163–174.

[14] A. A. Somasundara, A. Ramamoorthy, and M. B. Srivastava, "Mobile element scheduling with dynamic deadlines," IEEE Transactions on Mobile Computing, vol. 6, pp. 395–410, 2007.

[15] Y. Gu, D. Bozdag, and E. Ekici, "Mobile element based differentiated message delivery in wireless sensor networks," in

[16] K. Dantu, M. Rahimi, H. Shah, S. Babel, A. Dhariwal, and G. S. Sukhatme, "Robomote: enabling mobility in sensor networks," in IPSN, 2005.

[17] http://www.k-team.com/robots/khepera/index.html.

[18] J.-H. Kim, D.-H. Kim, Y.-J. Kim and K.-T. Seow, Soccer Robotics. Springer, 2004.

[19] 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, pp. 1430–1443, 2008.

[20] "Rendezvous planning in mobility-assisted wireless sensor networks," in RTSS '07: Proceedings of the 28th IEEE Inter-national Real-Time Systems Symposium, 2007, pp. 311–320.