DAWN: Energy Efficient Data Aggregation in WSN with Mobile Sinks

ShaoJie Tang* Jing Yuan† XiangYang Li*¶ Yunhao Liu‡ GuiHai Chen† Ming Gu§ JiZhong Zhao‖ Guojun Dai¶

*Department of Computer Science, Illinois Institute of Technology, Chicago, IL, USA
†State Key Laboratory of Novel Software Technology, NanJing University, NanJing, China
‡Department of Computer Science and Engineering, HKUST, China
§Key Laboratory of Information System Security, Tsinghua University, China
¶Institute of Computer Application Technology, Hangzhou Dianzi University, China
‖ Department of Computer Science and Engineering Xi'An JiaoTong University

***Abstract*—The benefits of using mobile sink to prolong sensor network lifetime have been well recognized. However, few provably theoretical results remain are developed due to the complexity caused by time-dependent network topology. In this work, we investigate the optimum routing strategy for the static sensor network. We further propose a number of motion stratifies for the mobile sink(s) to gather real time data from static sensor network, with the objective to maximize the network lifetime. Specially, we consider a more realistic model where the moving speed and path for mobile sinks are constrained. Our extensive experiments show that our scheme can significantly prolong entire network lifetime and reduce delivery delay.**

***Index Terms*—Sensor networks, mobility, detection, delay.**

I. Introduction

In this paper, we consider a large-scale system, which consists of sensor nodes that are homogeneous and highly energy-constrained. Further, due to possibly harsh terrains, replacing batteries on hundreds of nodes is difficult. In our application [8], sensors are deployed in a forest and we need to collect environment information (*e.g.* temperature, humidity and optical intensity) from all sensors to the mobile sinks. The basic requirement and key challenge in such data gathering is conserving the sensor energies, *i.e.* to maximize their lifetime. In the past decade, a lot of research has been directed and a wide variety energy-aware routing schemes are conducted in the context of sensor networks [2]–[5], [9]. The benefits of using mobile sinks to prolong sensor network lifetime have been well recognized [4], [5]. These approaches aim to alleviate the data aggregation burden from sensor nodes near the sink to other sensor nodes in the network, such that it is possible to extend the network lifetime significantly.

Indeed, although the practical feasibility of using a mobile sink is still considered far-fetched a few years ago, such capabilities are nowadays reality, thanks to recent breakthrough in unmanned autonomous vehicle (UAV) competition by DARPA's Grand Challenge program [6] and advances in customized robotics for sensors [7]. In our application [8], some sink nodes are built into the devices used by forest patrol guards, which can also serve as mobile base stations. GreenOrbs [8] is one of a wide variety of surveillance and dissemination applications that span large geographic areas. As a mobile sink moves in close proximity to sensors, data is transferred to the mobile sink for later depositing at the destination, *i.e.* the data processing center. Transmitting data over these much shorter distances leads to substantial power savings at sensors.

Although the potential benefits of using mobile sinks to prolong sensor network lifetime is significant, the theoretical model of this problem remains difficult. Two components are tightly coupled here. First, physical network topology may vary at different time instances, with the mobile sink being at different positions. In short, the location of the mobile sink is not deterministic but a function of time. Further, when mobile sink changes its location, the flow routing behavior may also change accordingly. Thus, in order to maximizE the network lifetime, we need take both sink location (time-dependent) and flow routing into account. Due to these difficulties, most of existing solutions either remain heuristic at best (*e.g.* [4], [5]) or ignore the relatively low moving speed of mobile sinks [9]. Although a provably $(1-\epsilon)$ optimal solution to network lifetime performance is proposed in [9], the moving region and moving speeds of the mobile sinks are not considered. In reality, there likely exist natural obstacles in a large monitoring field, such as stones, bushes and lakes. To avoid these obstacles, the mobile sink has to move within some constrained region. In our scenario, static sensor nodes are deployed in a large-scale forest and we are given several deterministic roads for human walking. This implies that the *possible* moving paths for mobile sinks are restricted to those walking road in the map. Further, the typical moving speed of a regular mobile sensor is far slower than the propagation speed of wireless signals. For such variable and slow moving speed may have significant impact on energy dissipation performance, it can not be simply ignored in network lifetime evaluation.

Taken constrained moving path into account, we study the network lifetime performance limit with mobile sinks. We formulate an optimization problem with flow routing and sink movement subject to multiple constraints,*e.g.*, interference, moving speed. With the objective to maximize the life time of static sensor network, we design a routing strategy for the static sensor network, and further specify the total time durations and corresponding locations for the mobile sink to be present. As another contribution, we design a number of motion strategies with constant approximation for the mobile sink, under the constraint that the mobile sink can only move along a pre-given road network.

A. Prior Work

As pointed out in [4] [5], network lifetime can be substantially increased if the optimization space can be expanded to include movement of mobile sinks during the course of sensor network operation. Relevant work in the area of mobile sink for network lifetime problems include [4], [5], [9]–[12]. In [10]–[12], the locations of the mobile sink are constrained on a set of *pre-determined* locations. Younis *et al.* [5] show that network lifetime can be increased through mobile sink. In [4], Luo and Hubaux propose to minimize the maximum load on a node among all the nodes in the network, which can be regarded as an equivalent problem to maximize network lifetime. The results in [4], [5] are *heuristic*, and thus do not provide any theoretical bound on network lifetime performance. Shi *et al.* design algorithms for the joint flow routing and mobile sink location problem, with approximation ratio $(1-\epsilon)$ regarding to the life time. However, they fail to take the moving speed into account which is shown to be critical in real time implementation.

B. Our contributions

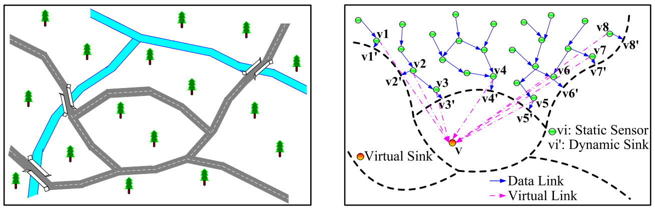
With the objective to maximize the life time of static sensor network, we focus on designing dynamic routing strategies for the static sensor network by taking mobile sinks into account. Accordingly, we design a number of motion strategies for the mobile sinks. Our detailed contributions are as follows:

This is the first work to exploit restricted sink mobility in *constrained* moving region for data gathering, to handle the natural obstacles problem in reality. We extensively investigate and evaluate the energy consumption and delay performance for data aggregation and collection with *multiple mobile sinks*. Our protocols are novel in realistic models with *restricted mobility*, which is resulted from different topographic environments in large-scale applications. We also propose several efficient routing schemes for data aggregation, to minimize the *delivery delay* while maximizing the network lifetime. In order to minimize the maximum energy consumption for k mobile sinks, we further propose and evaluate efficient schemes to find near optimal k sub-tours with *constant approximation*.

II. Preliminaries And Problem Formulation

A. Network Model and Assumptions

We consider a set of sensor nodes $\mathcal{N}$ deployed over a two-dimensional area. The locations of the sensors are fixed and known *a priori*. We assume that each node generates one data packet per timeslot to be transmitted to the static sink, and each packet has size k bits. The information from all the sensors needs to be aggregated at each timeslot, then sent to the sinks. There is a set of mobile sinks deployed in the sensor network. Data gathered by each static sink can be directly transmitted toward dynamic sinks as they move in close proximity. Further, each sensor i has a battery with finite, non-replenishable energy E_i. Based on the consideration of constrained moving region, we assume that there exists a road map, M, which indicates the feasible paths for mobile sinks to move along, as the gray strips showed in Fig. 1. Mobile sink(s) are assumed to be capable of short-range wireless communication and can exchange data as they pass by sensors and access points as a result of their motion.



(a) Road Map (b) Data Gathering

Fig. 1. Forest Monitoring

Fig. 1 shows that the monitoring field is partitioned into several subregions by walking roads in the map. In our model, we specify a point located outside the subregion as a *virtual sink* for data gathering. In this way, sensors located within the same subregion will deliver their data to a set of nodes near the walking path bounding that subregion. These nodes can be regarded as *dynamic sinks* since the set may vary during the course of data gathering. While moving along the paths, mobile sinks can collect data from dynamic sinks. We also assume that each mobile sink has an appropriate moving speed, which takes a typical value ranging from 0.1m/s to 2m/s.

Now we discuss the energy consumption for data transmission and reception. Whenever a sensor transmits or receives a data packet it consumes some energy from its battery. In this paper, the energy model for sensor nodes is based on the first order radio model [1]. A sensor consumes $\epsilon_{elec} = 50nJ/bit$ to run the transmitter or receiver circuity and $\epsilon_{amp} = 100pJ/bit/m^2$ for the transmitter amplifier. Thus, the energy consumed by a sensor i in receiving a k-bit data packet is given by, $R_{X_i} = \epsilon_{elec} \times k$. The energy consumed in transmitting a data packet to sensor j is given by, when $d_{i,j}$ is the distance between nodes i, j, $T_{X_{i,j}} = \epsilon_{elec} \times k + \epsilon_{amp} \times {d_{i,j}}^2 \times k$.

We define the *lifetime* T of the system to be the number of timeslots or periodic data readings from sensors until the first sensor is drained of its energy. A *data gathering schedule* specifies, for each timeslot, how the data packets from all the sensors are collected and transmitted to the dynamic sinks. Observe that a schedule can be considered as a collection of T directed trees, each rooted at the sink and spanning all the sensors. In other words, a schedule has one tree for each timeslot. Clearly, the network lifetime is intrinsically connected to the data gathering schedule. Our objective is to find an optimal schedule such that the network lifetime is maximized. While at the same time, we intensively investigate the maximal data rate that can be supported under the condition of maximizing the network lifetime.

B. Problem Formulation and Approaches

The focus of this paper is to investigate how to optimally move a set of mobile sinks to collect real time data (may be aggregated) in wireless sensor networks so that the network lifetime can be maximized. During the stage of data delivery to static sink nodes, we propose efficient link scheduling schemes in different cases of transmissions, *e.g.* with and without aggregation, respectively. Our objective is to minimize delivery delay or maximize data rate while minimizing the energy consumptions. In the stage of data collection with mobile sinks, we adopt a tour splitting method to get near optimal traveling tours for mobile sinks with constant approximation.

1) Maximum Lifetime Data Aggregation: Data aggregation performs in-network fusion of data packets, from different sensors enroute to the sink, in an attempt to minimize the number and size of data transmissions and thus save energies. When the data from different sensors are highly correlated, aggregation can be performed to alleviate the burden of highly constrained power. We assume that an intermediate sensor can aggregate multiple incoming packets into a single packet as an output.

Definition 1: MLDA is then, given a set of sensors and a static sink, together with their locations and the energy E_i of each sensor, find a data gathering schedule, where sensors can aggregate incoming data packets, with maximum lifetime.

Consider a schedule $\mathcal{S}$ with lifetime $\mathtt{T}$ timeslots. We denote by $f_{i,j}$ the total number of packets that node i transmits to node j in $\mathcal{S}$. Since any valid schedule must respect the energy constraints at each sensor, it follows that for each sensor $i = 1, 2, \cdots, n$,

$$\sum_{j=1}^{n+1} f_{i,j} \cdot T_{X_{i,j}} + \sum_{j=1}^{n} f_{j,i} \cdot R_{X_i} \leq E_i \qquad (1)$$

Recall that each sensor, for each one of the $\mathtt{T}$ timeslots, generates one data packet that needs to be collected, possible aggregated, and eventually transmitted to the virtual sink.

The schedule $\mathcal{S}$ induces a flow network $G = (V, E)$. The flow network G is a directed graph having as nodes all the sensors and the sink, and having edges (i, j) with capacity $\mathtt{f}_{i,j}$ whenever $\mathtt{f}_{i,j} > 0$. A necessary condition for a schedule to have lifetime $\mathtt{T}$ is that each node in the induced flow network can push flow $\mathtt{T}$ to the sink.

Next we consider the problem of finding a flow network G with maximum $\mathtt{T}$. In our model, each sensor push flow T to dynamic sinks, while respecting the energy constraints in (1) at all sensors. Clearly what needs to be found are the capacities of the edges of G. We call such a flow network G a *feasible* flow network with lifetime $\mathtt{T}$. A feasible flow network with maximum lifetime is called an *optimal feasible* flow network.

As Fig. 1 illustrated, the dashed lines indicate the walking paths for mobile sinks to move along according to the road map. Here we define a set of static sink candidates as $S = \{v_1, v_2, \cdots, v_n\}$, including the nodes nearby the road and are likely to become static sinks. Data generated from each sensor is first delivered to the static sinks, stored there and then relayed to mobile sinks as they pass by. Based on previous energy consumption function, it is intuitive that the optimal positions chosen for mobile sinks to stop for data collection should be those points which belong to the road and have the shortest Euclidian distance to appropriate static sink v_i at the same time, denoted by v_i'. Specifying an additional node V outside the subregion as a virtual "sink", we add one edge to each (v_i, V) pair and set d_i, the Euclidian distance between v_i and v_i', as the weight of the edge denoted by (v_i, V). Based on this reconstructed graph, we solve the following linear programming problem to get optimal data routing tree and sink moving paths. The collected aggregated data will be transmitted from static sinks to a mobile sink through edge (v_i, v_i') illustrated in Fig. 1 when the mobile sink passes v_i'.

2) Maximum Lifetime Data Routing: Data aggregation, while being a useful paradigm, is not applicable in all sensing environments. Imagine a scenario where the data being transmitted by the nodes are completely different (no redundancy) *e.g.* streams from video sensors. In such situations, it might not be feasible to fuse data packets from different sensors into a single data packet, in any meaningful way. Thereby larger number and size of data packets drain the sensor energies much faster than the energy consumption in above data aggregation scenario. Here data routing in sensor networks is modeled as the maximum network flow problem with energy constraints on sensors. The data gathering schedule specifies for each timeslot how to get and route data to the sink. A solution with integer programming is presented.

3) Sink Mobility: In order to optimally move mobile sinks to collect real time data in a sensor network so that the network lifetime can be maximized, we study the network lifetime performance when mobile sinks are employed.

Selective Patrol: As a first step, we consider the case when the forest patrolmen (mobile sinks) are ordered to walk along several roads that are necessary for (aggregated) data collection. This implies that only those roads with static sinks nearby for some timeslot will be selected. Intuitively, mobile sinks will move along the selected roads and stop at the static sinks to collect data packets.

Given a specified set of static sinks for each timeslot (we can get this from the solution of above two subproblems), we construct an undirected weighted graph $G' = (V', E')$ to build data gathering tours for mobile sinks. Here V' consists of all the static sinks, *i.e.* $V' = \{v_i, 1 \leq i \leq k\}$. Since G is connected, there exists an edge for each pair of vertices v_i and v_j $(v_i, v_j \in V')$, with weight w defined as the period for a mobile sink to move from v_i to v_j along the shortest path.

For the computation of w, we take the speed s of mobile sinks into consideration. Since s may vary in different topographic environment, w is a computational result of comprehensive impact of complex road conditions between each pair of static sinks. We assume the shortest path between v_i and v_j consists of a sequence of edges $\{e_1, e_2, \cdots, e_m\}$, where e_i denotes a segment of walking path between two intersections in the road map with length l_i. For each edge e_i, it is assigned a speed s_i to represent the average moving speed on that segment of road. Then, the weight of the edge between v_i

and v_j in G' is given by,

$$w_{ij} = \frac{l_1}{v_1} + \frac{l_2}{v_2} + \cdots + \frac{l_m}{v_m}.$$

Based on this reconstructed weighted graph G', we offer an efficient method to find near optimal k sub-tours with minimized maximum energy consumption for k mobile sensors.

Inclusive Patrol: We then studied the case when patrolmen are ordered to visit *all* the walking paths in the road map, performing abnormality surveillance besides collecting environment data from sensors. In this case, we model the data collection as a *k-Chinese Postman Problem* to find k split tours in G while minimizing the maximum tour time.

III. Data Aggregation towards Static Sinks

A. Finding a near-optimal feasible flow network

An optimal feasible flow network can be found using the following integer program with linear constraints. The integer program, in addition to the variables for the lifetime $\mathtt{T}$ and the edge capacities $\mathtt{f}_{i,j}$, uses the following variables: for each sensor $k = 1, 2, \cdots, n$, let $\pi_{i,j}^{(k)}$ be a flow variable indicating the flow that a sensor k sends to the virtual sink over the edge (i, j). The integer program is given by,

$$\max T$$

subject to the energy constraint (1) and the constraints below, for each $k = 1, 2, \cdots, n$,

$$\begin{cases} a) \quad \sum_{j=1}^{n} \pi_{j,i}^{(k)} = \sum_{j=1}^{(n+1)} \pi_{i,j}^{(k)} \\ b) \quad T + \sum_{j=1}^{n} \pi_{j,k}^{(k)} = \sum_{j=1}^{n+1} \pi_{k,j}^{(k)} \\ c) \quad 0 \leq \pi_{i,j}^{(k)} \leq f_{i,j} \\ d) \quad \sum_{i=1}^{n} \pi_{i,n+1}^{(k)} = T \end{cases} \tag{2}$$

where all the variables are required to take integer values. For each $k = 1, 2, \cdots, n$, constraint (2) consists of following restrictions: *a)*enforces the flow conservation principle at a sensor; *b)*ensures that T flow from sensor k reaches the virtual sink; *c)*ensures that the capacity constraints on the edges of the flow network are respected and *d)* ensures the sample rate constraints on each node are respected.

When all the variables are allowed to take fractional values, the linear relaxation of the above integer program can be computed in polynomial-time. Then, we can obtain an approximation for the admissible flow network as follows. First, we fix the edge capacities to the floor of their values obtained from the linear relaxation, and then solving the linear program subject to constraints (2).

B. Constructing routing trees from a feasible flow network:

Next, we discuss how to get a schedule from previous computed feasible flow network. Recall that a schedule is a collection of directed trees rooted at the virtual sink that span all the sensors, with one such tree for each timeslot. Each such tree specifies how data packets are gathered and transmitted to the virtual sink. We call these trees *aggregation trees*. An aggregation tree may be used for one or more timeslots; we indicate the number of timeslots f an aggregation tree is used by associating the value f with each one of its edges; we call f as the lifetime of the aggregation tree. Given a feasible flow network G with lifetime $\mathtt{T}$ and a directed tree A rooted at the sink with lifetime f, we define the (A, f)-reduction G' of G to be the flow network that results from G after reducing the capacities of all of its edges, that are also in A, by f. We call G' the (A, f)-reduced G. An (A, f)-reduction G' of G is *feasible* if the maximum flow from v to the sink in G' is $\geq T - f$ for each vertex v in G'. Note that A does not have to span all the vertices of G, and thus it is not necessarily an aggregation tree. Moreover, if A is an aggregation tree, with lifetime f, for a feasible flow network G with lifetime T, and the (A, f)-reduction of G is feasible, then the (A, f)-reduced flow network G' of G is a feasible flow network with lifetime $T - f$. Therefore, we can devise a simple iterative algorithm, to construct a schedule for a feasible flow network G with lifetime $\mathtt{T}$, provided we can find such an aggregation tree A.

Algorithm 1 GetTree

```
Input: Flow Network $G$, Lifetime $\mathtt{T}$, Virtual Sink $t$
Output: $T$, $G$, $A$
$f := 1$;
Let $A = (V_o, E_o)$ where $V_o = t$ and $E_o = \emptyset$;
while $A$ does not span all the nodes of $G$ do
    for each edge $e = (i, j) \in G$ such that $i \notin V_o$ and $j \in V_o$ do
        Let $A'$ be $A$ together with the edge $e$;
        Let $G_r$ be the $(A', 1)$-reduction of $G$;
        if MAXFLOW$(v, t, G_r) \geq T - 1$ for all nodes $v$ of $G$ then
            $V_o := V_o \cup \{i\}, E_o := E_o \cup \{e\}$;
            Break;
Let $c_{min}$ be the minimum capacity of the edges in $A$;
Let $G_r$ be the $(A, c_{min})$-reduction of $G$;
if MAXFLOW$(v, t, G_r) \geq T - c_{min}$ for all nodes $v$ of $G$ then
    $f := c_{min}$;
Replace $G$ with the $(A, f)$-reduction of $G$;
return $f, G, A$
```

We use algorithm 1 to get an aggregation tree $\mathtt{A}$ with lifetime f from a feasible flow network G with lifetime $\mathtt{T} \geq f$. Throughout this routine, we maintain the invariant that $\mathtt{A}$ is a tree rooted at sink and the $(\mathtt{A}, f)$-reduction of G is feasible.Tree $\mathtt{A}$ is formed as follows. Initially $\mathtt{A}$ contains just the sink. While $aggtree$ does not span all the sensors, we find and add to $\mathtt{A}$ an edge $e = (i, j)$, where $i \notin A$ and $j \in A$, provided that the (A', f)-reduction of G is feasible-here A' is the tree A together with the edge e and f is the minimum of the capacities of the edges in A'. Given a flow network G and base station t such that each sensor s has a minimum $s - t$ cut of size $\geq T$ (*i.e.* the maximum flow s to t in G is $\geq T$), we can prove that it is always possible to find a sequence of aggregation trees, via the GetTree algorithm, that can be used to aggregate T data packets from each of the sensors. The proof of correctness is based on a powerful theorem in graph theory and is omitted due to lack of space. We refer to the approach described in this section, for finding a maximum lifetime schedule with data aggregation, as the MLDA approach.

C. Interference-free Tree Scheduling

In order to minimize the total data delivery delay for a given admissible flow network, we next study how to schedule the links with efficacy, *i.e.* which links should be activated (and sleep) for each timeslot.

Here we try to find an efficient link schedule based on previous aggregation trees to minimize the delivery delay, D, which is defined as the time during for aggregated data collection from each node to static sinks. With the knowledge of lifetime (workload) for each aggregation tree, we consider how to schedule different instances for different trees to avoid conflicts and minimize the transmission delay.

We use l_e, c_e to denote the load and the capacity of a link e respectively, and r_i denotes the interference range of v_i. [13] gives both a necessary and a sufficient condition on the link flows such that an interference-free link scheduling is feasible under various interference models. It follows that,

Theorem 1: Under the RTS/CTS model, any link flow l that permits an interference-free link scheduling must satisfy the constraint $\frac{l(e)}{c(e)} + \sum_{e' \in I \geq (e)} \frac{l(e')}{c(e')} \leq 2C_1$. If $\frac{l(e)}{c(e)} + \sum_{e' \in I \geq (e)} \frac{l(e')}{c(e')} \leq 1$, then link flow l permits an interference-free link scheduling. Here $C_\alpha \leq (6\alpha+1)^2 + 11$, and C_α is called α-hop interference number.

Theorem 2: Under the fPrIm interference model, any link flow l that permits an interference-free link scheduling must satisfy the following constraint $\frac{l(e)}{c(e)} + \sum_{e' \in I^{in}(e)} \frac{l(e')}{c(e')} \leq \lceil \frac{2\pi}{\arcsin \frac{\gamma-1}{2\gamma}} \rceil$. If $\frac{l(e)}{c(e)} + \sum_{e' \in I^{in}(e)} \frac{l(e')}{c(e')} \leq 1$, then any link flow l permits an interference-free link scheduling.

Further, a fast distributed weighted coloring schedule algorithm is proposed in [13].

Theorem 3: If $\hat{l}$ is the delivery delay supported by our schedule, compared with the one of the optimal schedule, l^*, it is satisfied that, $\hat{l}/l^* \leq O(\log(\varphi)+1)$, where φ denotes the ratio between max and min interference ranges.
The proof of above two theorems can be found in [13] and are omitted here due to space limit.

IV. Data Collection towards Static Sinks

A. Finding a near-optimal admissible flow network

The problem is to find an efficient schedule to collect and transmit the data to the base station, such that the system lifetime T is maximized. Here the *base station* is interpreted as the virtual sink we defined in Section II-A. Since no in-network aggregation is performed, our problem can be viewed as a maximum flow problem with energy constraints at the sensors, subject to integral flows. It can be solved by the following integer program with linear constraints:

$$\max T$$

Algorithm 2 Distributed Interference-free Tree Scheduling

```
Input: $G'$
Output: A valid coloring of links in $G'$

Node $v_i$ computes a subset, say $H_i$, of all communication
  links containing $v_i$ such that link $l_{i,j} \in H_i$ iff. $r_i > r_j$;
while node $v_i$ failed to obtain the channel do
    Node $v_i$ monitors the channel and competes for the
    channel;
for each link $l_{i,j} \in H_i$ do
    Color link $l_{i,j}$ with the first fit $w_{i,j}$ colors that are
    not used by any link that interferes or is interfered by
    $l_{i,j}$. Here, the assigned colors are not required to be
    continuous;
Broadcasts the message Color($i$,$j$,$k$) to each head of links
  that conflict with $l_{i,j}$.
```

subject to energy constraint 1 for each sensor and the flow conservation constraints

$$T + \sum_{j=1}^{n} f_{i,j} = \sum_{j=1}^{n+1} f_{j,i},\ i = 1, 2, \cdots, n,\ j = 1, 2, \cdots, n+1$$

where all variables T and $f_{i,j}$ are required to be non-negative integers. A near-optimal solution to this problem can be obtained as follows. First, solve the linear relaxation of the above integer program, by replacing the requirement that all the T and $f_{i,j}$ variables are non-negative integers, with the requirement that they are non-negative integers, with the requirement that they are non-negative real numbers. Second, compute a solution to the linear program that consists of the above two equations, by fixing the values of the $f_{i,j}$ variables to the floor of their values obtained in the previous step. The solution obtained in this second step is guaranteed to have integer values for all the variables, since it is a *max-flow problem* with integer capacities.

Observe that this solution provides us readily with a schedule for collecting the data packets without aggregation from all the sensors, during the lifetime of the system. A simple way to construct such a schedule would be to take the flow network obtained from the solution, and push T data packets from each sensor on one or more paths (with available capacities) to the virtual sink. We define the *depth* of this schedule to be the maximum length of a path used by any sensor to transmit its data to the sink. This approximate solution provides a near-optimal system lifetime that is efficiently computable.

B. Interference-free Link Scheduling

Next, we try to find an efficient schedule based on previous feasible flow network. In order to maximize the data rate while minimizing data transmission delay, our interference-free link scheduling can be modeled as a Minimum Fractional Weighted Link Schedule (MFWLS) Problem, which is proved to be NP-Hard in [14]. Recall that we denote by $G' = (V', E')$ an undirected graph representing the conflict graph of the communication links in the original graph G conducted from the road map. A subset I of V' is an *independent set* (IS) of G' if no two nodes in I are adjacent. If I is an IS of G' but no proper superset of I is an IS of G', then I is called a *maximal IS* of G'. Any node ordering $\langle v_1, v_2, \cdots, v_n\rangle$ of V' induces a maximal IS I in the following first-fit manner: Initially, $I = v_1$. For $i = 2$ up to n, add v_i to I if v_i is not adjacent to any node in I. An IS of the largest size is called a *maximum IS* (MIS). Let $\mathcal{I}$ be the collection of all independent sets of G'. A *(fractional) link schedule* in N is a set $S = (I_j, \lambda_j) : 1 \leq j \leq k$ with $I_j \in \mathcal{I}$, and $j > 0$ for $1 \leq j \leq k$. The value $\sum_{j=1}^{k} \lambda_j$ is referred to as the *length* of S. The *independence number* of G', denoted by $\alpha(G')$, is defined to be $\max_{I\in\mathcal{I}} |I|$. Here weight $w(v)$ is interpreted as the total number of packets each link transmits, which we get from Section IV-A. For any $w > 0$, the *weighted independence number* of (G', w), denoted by $\alpha(G', w)$, is defined to be $max_{I\in\mathcal{I}}w(I)$. For any $w > 0$, a *fractional coloring* of (G', w) is a set of k pairs (I_j, λ_j) with each $I_j \in \mathcal{I}$ and $\lambda_j > 0$ for $1 \leq j \leq k$ satisfying that for each $v \in V'$,

$$\sum_{1\leq j\leq k, v\in I_j} \lambda_j = w(v).$$

Here k and $\sum_{j=1}^{k} \lambda_j$ are referred to as the number and total weight of the coloring respectively. The *fractional chromatic number* $\mathcal{X}_f(G', w)$ of (G', w) is defined as the minimum weight of all fractional colorings of (G', w). In our specific case, $\mathcal{X}_f(G', w)$ denotes the minimum number of total packets transmitted in a single timeslot following a schedule S conducted from the MFWLS Problem. We have that $\mathcal{X}_f(G', w) \geq \frac{w(V')}{\alpha(G')}$. MFWLS is with the goal of finding a fractional coloring of (G', w) with total color weight equal to $\mathcal{X}_f(G', w)$, *i.e.* maximizing the data rate of the admissible flow network while minimizing the total data delivery delay. In order to get such an efficient link scheduling scheme, we introduce a first-fit schedule algorithm for approximation [15]. Consider a vertex ordering $\langle v_1, v_2, \cdots, v_n\rangle$ of V'. For any $U \subseteq V'$, the *first-fit MIS* of U is the MIS of $G[U]$ selected in the first manner in the ordering $\langle v_1, v_2, \cdots, v_n\rangle$. The idea of Algorithm 3 is to iteratively pick up a first-fit MIS of remaining nodes with positive weight, and then assign to this MIS a color with a weight (delay) to saturate at least one node. This ensures that at least one node gets its flow demand satisfied and stops moving onto the subsequent iteration. As a result, the number of colors is bounded by n, and the running time is $O(n^2)$. Lemma 4 gives a bound on the total color weight of the output coloring.

Lemma 4: The coloring output by Algorithm 3 uses at most n colors of total weight at most $max_{1\leq i\leq n}w(V_i')$, where V_i' consists of v_i and all its neighbors in $v_1, v_2, \cdots, v_{i-1}$ for each $1 \leq i \leq n$.
By Lemma 4, we have following theorem to show the total delay of our link schedule mechanism.

Theorem 5: If we denote by D the delivery delay for data collection followed by our scheme, we have $D \leq \max\{w(V_i') : 1 \leq i \leq n\}$.

Algorithm 3 LinkSchedule

Input: Conflict Graph G' of the Admissible Flow Network, Weight w, An Ordering $\langle v_1, v_2, \cdots, v_n\rangle$ of V'
Output: A fractional weighted coloring Π of (G', w).

```
$Pi := \emptyset$;
$U := \{v \in V' : w(v) > 0\}$;
while $U \neq \emptyset$ do
    $I$ :=the first-fit MIS of $U$;
    $\lambda := \min_{v\in I} w(v)$
    add $(I, \lambda)$ to $Pi$;
    for each $v \in U$ do
        $w(v) := w(v) - \lambda$;
        if $w(v) = 0$ then
            remove $v$ from $U$;
output $\Pi$;
```

V. Sink Motion Strategy

Here we consider two scenarios: (1)all walking paths should be visited, which we call Forest Patrolman Problem (FPP) and (2)not all walking paths need to be visited, in other words, we only choose roads that consist of edges forming shortest paths between all the virtual static sink nodes.

A. Forest Patrolman Problem

In Forest Patrolman Problem (FPP), mobile sinks should visit all the walking paths in the roadmap to perform abnormality surveillance besides collecting environment data from sensors. To minimize the maximum path length of mobile sinks, we can model FPP as min-max k-Chinese postman problem (MM k-CPP). Given a road network, the Chinese postman problem (CPP) is to find the shortest postman tour covering all the roads in the network. Given an undirected edge-weighted graph and a distinguished depot node, the MM k-CPP consists of finding $k > 1$ tours, starting and ending at the depot node, such that each edge is traversed by at least one tour and the length of the longest tour is minimized. For the min-max k-Chinese postman problem (MM k-CPP) the aim is to minimize the length of the longest of the k tours. This kind of objective is preferable when customers have to be served as early as possible. Furthermore, tours will be enforced to be more balanced resulting in a fair scheduling of tours. Although the CPP and the k-CPP are polynomially solvable, the MM k-CPP is shown to be *NP*-hard by a reduction from the k-partition problem [16], which is also NP-hard. It is solvable if the degree of each node is even. Otherwise, it has a 3/2 approximation. Hence, we must rely on heuristics producing approximate solutions. Here we introduce a tabu search algorithm recently proposed in [17] which outperforms all known heuristics.

First we should construct an undirected weighted graph $G' = (V', E')$ from the original roadmap. Graph G' includes all the paths in the roadmap as edges and intersections as vertex. The weight $w : E' \rightarrow R^+$ for each edge which we interpret as the distance (along the road) between two intersections v_i and v_j incident to this edge. A distinguished depot node $v_0 \in V'$ and a fixed number $k > 1$ of postmen. Our aim is to find k closed walks (tours) where each tour starts and ends at the depot node and each edge $e \in E$ is covered by at least one tour. To minimize the length of the longest of the k tours. A feasible solution, called $k-postmantour$, is a set C of k closed walks, $C = C_1, \cdots, C_k$, such that each tour C_i contains the depot node v_0, all edges $e \in E'$ are covered by at least one tour C_i and each postman is involved.

Note that in our mobile sink routing model, we do not constrain all the mobile sinks to start from a single position which is so called *depot node*. In our more realistic model, mobile sinks can start to move from any position along each possible road. To release that constraint, we introduce a virtual node V_0 outside the roadmap as a virtual depot node and $w(v_i', V_0)$ is set to be zero for $i = 1, \cdots, n$. In this way, by deleting the virtual node V_0 from Algorithm 4's output, we can get a near optimal solution for MM k-CPP without explicitly setting a depot node.

For an edge set or a walk F, let F denote the cardinality of the edge set and the number of edges contained in the walk, respectively. We extend the weight function w to walks $F = (e_1, \cdots, e_p)$ by defining $w(F) = \sum_{i=1}^{p} w(e_i)$. Now, for a k-postman tour C, we denote the maximum weight attained by a single tour C_i as $w_{max}(C)$, *i.e.*,

$$w_{max}(C) = \max_{i=1,\cdots,k} w(C_i).$$

The objective of the MM k-CPP is to find a k-postman tour C^* which minimizes w_{max} among all feasible k-postman tours, *i.e.*,

$$w_{max}(C^*) = \min\{w_{max}(C) | C \text{ is a } k\text{-postman tour}\}.$$

We denote by $SP(v_i, v_j)$ the set of edges on the shortest path between nodes $v_i, v_j \in V$. The distance of the shortest path between v_i and v_j is given by $w(SP(v_i, v_j))$. We will use *Dijkstra's algorithm* [18] and the *Floyd-Warshall* [19] algorithm to compute single-pair and all pairs shortest paths, respectively, in our implementation.

First we try to find an optimal solution for MM 1-CPP. The Chinese postman problem is to find the minimum length postman tour of a connected graph. If there is an Euler tour in the graph, then it solves the Chinese postman problem. Thus, whenever every node of a connected graph is incident to an even number of edges, the Chinese postman problem reduces to simply finding an Euler tour, which is known to exist in such a graph. On the other hand, given any postman tour of G,every edge e is in the tour at least once, but perhaps more than once. Let $1 + x_e$ be the number of times edge e is in the tour. Let G' be formed from G by putting x_e additional copies of edge e in G'. That is, where G had one copy of edge e, G' has $(1 + x_e)$ copies of edge e. Then the postman tour of G becomes an Euler tour of G'. In the graph G', every node is incident to an even number of edges. In this way, finding the numbers x_e of an optimum postman tour is equivalent to the problem of finding an integer $x_e \geq 0$ for every edge e of G such that $\sum w_e x_e$ is minimized subject to the following constraints,

$$\min \sum w_e x_e$$
$$\begin{cases} x_e \text{ is integer, } e \in E; \\ x_e \geq 0, \; e \in E; \\ \sum_{e \in E} a_{ne} x_e - 2l_n = b_n, \; n \in N; \end{cases} \tag{3}$$

Here the integer x_e represents the number of extra times (according to the original once) the edge e is traversed. In terms of the patrolman, x_e is the number of times he must traverse an edge without performing surveillance. Let the *node-edge incidence matrix* (a_{ne}), $n \in N$ and $e \in E$, be defined as a_{ne} equals to 1 if edge e meets node n and 0 otherwise. The variables l_n can be thought of as adjoining loops to the graph at each node, where a *loop* is an edge with two ends meeting the same node. The above Linear Programming problem is a special case of the general matching problem [20]. We can reduced the problem to an equivalent 1-matching problem [21], [22] to get an optimal 1-route $R = (V_0, e_{i1}, v_{i2}, \cdots, v_{im}, e_{im}, V_0)$. Let $L = w(R)$ and let $R_{v_{in}}$ denote the path $(v_0, e_{i1}, v_{i2}, \cdots, v_{in})$, with $n \leq m$. We denote the cost of a shortest path from a vertex v to u by $w(v, u)$. For each j, $1 \leq j \leq k$, $L_j = j/k(L - 2w_{max}) + w_{max}$.

Algorithm 4 MM k-postmen

```
Input: $G'$, $V_0$
Output: $C_1, C_2, \cdots, C_k$
Find an optimal 1-route $R$ where $V_0$ is the start vertex;
for each j, $1 \leq j \leq k$ do
    find the last vertex $v_{l'(j)}$ such that $w(R_{v_{l'(j)}}) \leq L_j$;
Let $r_j = L_j - w(R_{v_{l'(j)}})$;
for each j, $1 \leq j \leq k$ do
    if $r_j + w(v_{l'(j)}, v_0) \leq w(v_{l'(j),v_{l'(j)+1}}) - r_j + w(v_{l'(j)+1}, v_0)$ then
        $v_{l(j)} = v_{l'(j)}$;
    else
        $v_{l(j)} = v_{l'(j)+1}$;
Let $C_1 = (v_0, e_{i1}, v_{i2}, \cdots, v_{l(1)}), C_2 = (v_{l(1)}, \cdots, v_{l(2)}), \cdots, C_k = (v_{l(k-1)}, \cdots, v_0)$;
Build the $k$-route by connecting $v_0$ to both the initial and terminal vertices of the $C_j$'s with shortest paths to transform $C_j$ into a subroute.
```

Theorem 6: If $\hat{C}_k$ is the cost of the largest of the k subtours generated by Algorithm 4, and C_k^* is the cost of the largest subtour in an optimal solution of k-CPP, then the Algorithm 4 produces $\hat{C}_k$ in $O(n^3)$ time such that [23]

$$\hat{C}_k / C_k^* \leq 2 - 1/k.$$

After the virtual node V_0 is simply eliminated from above tours, we get k patrol paths for k mobile sinks, with the initial location at the first vertex of each tour.

B. Partial Patrol Problem

Recall that we set the shortest distance (along the walking path) between v'_i and v'_j as the weight of the edge denoted by $e_{i',j'}$ for $i' \neq j'$. Different from $ForestPatrolmanProblem$, mobile sinks visit only partial of the walking paths in the roadmap to collect aggregated data. Here we should reconstruct an undirected weighted graph $G' = (V', E')$ from the original roadmap. Graph G' includes all the paths in the roadmap as edges and intersections as vertex. Based on this reconstructed graph, we get a complete weighted graph G'. A shortest moving path should be found for mobile sinks to visit the vertices efficiently.

Here we assume there exist k mobile sinks to collect the aggregated data. After we get a Minimum Spanning Tree t for above weighted tree, we try to split t into k subtrees $t_1, t_2, \cdots, t_k$ with the goal of minimizing the maximum weight of these subtrees $w(t_i)$ for $i = 0, 1, \cdots, k$. Clearly, we will get C_l^{k-1} forests to make a choice. For each forest f_j, there exist a set of $k-1$ edges, s_j, which is cut from t and split t into k components (subtrees) $t_{j1}, t_{j2}, \cdots, t_{jk}$. $w_{max}(f_j)$ denotes the weight of the subtree with a maximum weight from the k subtrees in f_j. We calculate $w_{max}(f_j)$ of each forest f_j for $j = 1, 2, \cdots, C_l^{k-1}$. Thus, the forest f_m with the minimum $w_{max}(f)$ is exactly the one we need for constructing final k tours. We denote by C_i a Euler tour generated for each subtree t_{mi} in f_m such that we get k final tours $C_1, C_2, \cdots, C_k$ for k mobile sinks.

Algorithm 5 Splitting Tour

```
Input: $G'$, $k$
Output: $C_1, C_2, \cdots, C_k$
Find a Minimum Spanning Tree $t$ with totally $l$ edges;
Enumerate $k-1$ edges to be cut from $t$ such that we get $C_l^{k-1}$ forests which forms a set $F = (f_1, f_2, \cdots)$;
for each forest $f_j$, $1 \leq j \leq C_l^{k-1}$ do
    Calculate the weight of each components as $w(t_{ji})$ for $i = 0, 1, \cdots, k$;
    Find the $w_{max}(t_j) = \max(w(t_{ji}))$;
Find the forest $f_m$ with $\min w_{max}(t_j)$;
for each component $t_i$ do
    construct a Euler tour $C_i$ based on $t_i$.
```

Theorem 7: If $w(\hat{f})$ is the total weight of the k subtours generated by Algorithm 5, and $w(f^*)$ is the total length of the optimal k tours, it is satisfied that $w(\hat{f}) \leq 2w(f^*)$.

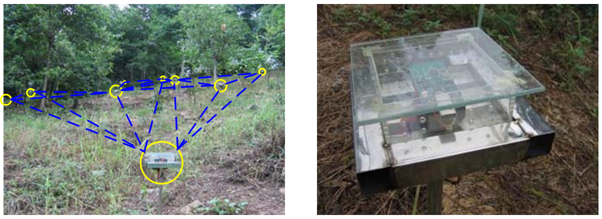
Proof: Proof is omitted here due to space limit. ■

VI. Experiment Results

We conducted extensive experiments to study the performance of our data aggregation and data collection schemes in terms of lifetime and energy consumption.

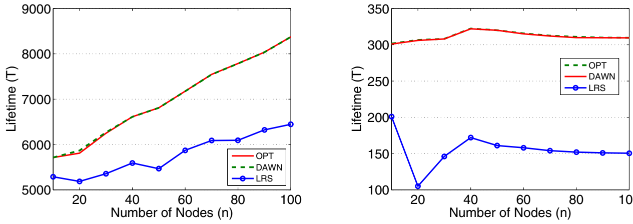
A. Testbed

For the experimental results presented in this section, we consider a network of sensors deployed in *GreenOrbs* [8], a testbed built by Zhejiang Forestry University for forest surveillance. *GreenOrbs* contains of hundreds of wireless sensors for light, radiation, temperature, and humidity. The purpose of the testbed is primarily for gathering environment conditions where rare and endangered plants have been living for centuries in Tian Mu Mountain, as showed in Fig. 2. In the past, thousands of guards were employed to patrol in the huge mountain everyday, which costs a large amount of time and labor cost. Currently, *GreenOrbs* takes over the task for environment data gathering and it has been working over months. We use real data traces gathered from *GreenOrbs* testbed to evaluate the performance of our routing schemes in terms of network lifetime.



(a) Sensor Testbed (b) Encapsulated Sensor

Fig. 2. GreenOrbs Testbed composed of 120 nodes.

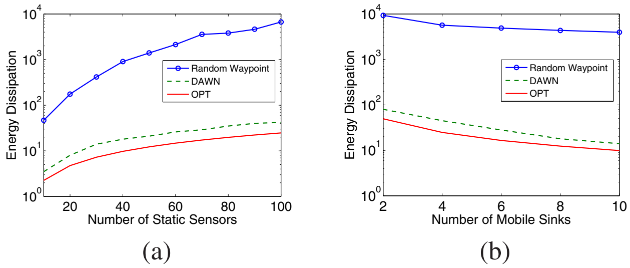


(a) Data Aggregation (b) Data Collection

Fig. 3. Lifetime of Static Sensors for Forest Monitoring

B. Lifetime Evaluation

Stationary Sensor Network: Our first set of experiments evaluate the basic performance of our routing scheduling schemes in terms of the network lifetime. The energy model for the sensors is based on the first order radio model described in Section II-A. We compare the data gathering scheduling given by *DAWN* algorithms with that obtained from a chain-based hierarchical protocol proposed in [24]. In each experiment, we measure the network lifetime T, *i.e.* the number of timeslots or periodic data readings from sensors until the first sensor is drained of its energy. Recall that the (integral) solution given by *DAWN* algorithm is an approximation of the optimal(fractional) solution. In order to estimate the quality of approximation, we also measure the system lifetime given by the optimal fractional solutions(denoted as *OPT*) for our data gathering problems. As shown in Fig. 3, the lifetime of the schedule given by the *DAWN* algorithms always significantly outperforms that given by the *LRS* protocol, both for data aggregation and data collection. In case of data collection,



(a) (b)

Fig. 4. Energy Dissipation of Mobile Sinks for Data Collection: (a) varied number of static sensors with 4 mobile sinks (b) varied number of mobile sinks with 100 static sensors

the *DAWN* algorithm performs 1.06 to 1.72 times better than the *LRS* protocol in terms of system lifetime.

Mobile Sink: In this set of experiments, we evaluate the performance of our mobile sink motion scheduling algorithms in terms of the maximum energy dissipation.

1) Network Constructions: In order to construct a network Graph G' defined in Section V, we calculate the weight (energy consumption) of each edge in G as a function of moving speed s. We select a square region in *GreenOrbs* testbed with size $500m \times 500m$. Obtained the road map for the selected region with relevant topographic information, sink moving speed can be estimated using fuzzy logic control schemes. To minimize the maximum energy dissipation for k mobile sinks, our scheduling methods try to evenly assign a minimum weighted Euler Circuit to k mobile sinks.

2) Energy Dissipation: Fig. 4 summarizes our main results. Note that the numerical results for each sensor in Fig. 4 represent how many units of energy is dissipated by the mobile sink with maximum weight for each experiment. In the real life implementation, the unit of energy consumption can be scaled accordingly. As shown in Fig. 4, the lifetime (energy dissipation) under the scheduling of *DAWN* is always significantly better than that under the Random Waypoint Scheduling. We also notice that the performance of *DAWN* is within a constant approximation compared with the optimum solution.

VII. Conclusion

In this paper, we study how to optimally schedule the (aggregated) packet routing and move a set of mobile sinks to collect the data in wireless sensor networks so that the network lifetime can be maximized. We present a scheduling strategy for the stationary sensor network in order to maximize the network lifetime. We also design a number of motion strategies for mobile sinks to minimize the energy consumption under different data collection requirements. We investigate the relationship between the system lifetime and different parameters in network planning.

VIII. Acknowledgement

The research of authors are partially supported by NSF CNS-0832120, National Natural Science Foundation of China under Grant No.60828003, No.60773042, No. 60873262, the Natural Science Foundation of Zhejiang Province under Grant No.$Z1080979$, program for Zhejiang Provincial Key Innovative Research Team, program for Zhejiang Provincial Overseas High-Level Talents (One-hundred Talents Program), National Basic Research Program of China (973 Program) under grant No. $2010CB328100$ and $2010CB334707$, and Tsinghua National Laboratory for Information Science and Technology (TNList). The work is partly supported by China NSF grants (60721002, 60825205) and China 973 project (2006CB303000).

References

[1] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy efficient Communication Protocols for Wireless Microsensor Networks," in *HICSS*, 2000.

[2] W. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for Information Dessemination in Wireless Sensor Networks," in *ACM Mobicom* , 1999.

[3] J. Chang and L. Tassiulas, "Energy Conserving Routing in Wireless Ad-hoc Networks," in *IEEE Infocom*, 2000.

[4] J. Luo and J. Hubaux, "Joint mobility and routing for lifetime elongation in wireless sensor networks," in *IEEE INFOCOM*, vol. 3, 2005.

[5] M. Younis, M. Bangad, and K. Akkaya, "Base-station repositioning for optimized performance of sensor networks," in *VTC 2003*.

[6] *http://www.darpa.mil/grandchallenge/index.html*. "DARPAR Grand Challenge 2005,"

[7] G. Sibley, M. Rahimi, and G. Sukhatme, "Robomote: A tiny mobile robot platform for large-scale ad-hoc sensor networks," in *IEEE Int. Conf. on Robotics and Automation*, 2002.

[8] M. L.F., H. Y., L. Y.H., Z. J.Z., T. S.J., and L. X.Y., "Canopy Closure Estimates with GreenOrbs: Long-Term Large-Scale Sensing in the Forest," in *ACM SenSys'09*.

[9] Y. Shi and Y. Hou, "Theoretical results on base station movement problem for sensor network," in *IEEE INFOCOM*, 2008, pp. 1–5.

[10] S. Basagni, A. Carosi, E. Melachrinoudis, C. Petrioli, and Z. Wang, "A new MILP formulation and distributed protocols for wireless sensor networks lifetime maximization," in *IEEE ICC*.

[11] S. Gandham, M. Dawande, R. Prakash, and S. Venkatesan, "Energy efficient schemes for wireless sensor networks with multiple mobile base stations," in *IEEE GLOBECOM'03*, vol. 1, 2003.

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

[13] W. Wang, Y. Wang, X. Li, W. Song, and O. Frieder, "Efficient Interference-Aware TDMA Link Scheduling for Static Wireless Networks," in *ACM MobiCom*. 2006

[14] L. C. and Y. M., "On the hardness of approximating minimization problems," in *Journal of ACM)*, pp. 41(5):960–981.

[15] W. P.J., "Multiflows in Multihop Wireless Networks," in *MobiHoc'09)*.

[16] G. Frederickson, M. Hecht, and C. Kim, "Approximation algorithms for some routing problems," in *SIAM Journal on Computing*, 1978.

[17] D. Ahr and G. Reinelt, "A tabu search algorithm for the min-max k-Chinese postman problem," in *Computers and Operations Research*.

[18] E. Dijkstra, "A note on two problems in connetion with graphs," in *Numerische Mathematik*.

[19] R. Floyd, "Algorithm 97: shortest path," in *Communications of the ACM*, 1962, p. 5(6):345.

[20] J. Edmonds and E. Johnson, "Matching: a well-solved class of integer linear programs," in *Combinatorial structures and their applications*.

[21] J. Edmonds, "Paths, trees and flowers," in *Canadian Journal of Mathematics 17*.

[22] J. Edmonds, "Maximum matching and a polyhedron with 0, 1-vertices," in *Journal of Research of the National Bureau of Standards Section B, 1, 2*, 1965, pp. 125–130.

[23] J. Edmonds and E. Johnson, "MATCHING, EULER TOURS AND THE CHINESE POSTMAN," in *Mathematical Programming 5*, 1973.

[24] L. S., R. C.S., and S. K., "Data Gathering in Sensor Networks using the Energy-Delay Metric," in *Proc. of IPDPS Workshop on Issues in Wireless Networks and Mobile Computing*, 2001.