Lexicographically Optimal Routing for Wireless Sensor Networks with Multiple Sinks

Vahid Shah-Mansouri, *Student Member, IEEE*, A. Hamed Mohsenian-Rad, *Student Member, IEEE*, and Vincent W.S. Wong, *Senior Member, IEEE*

***Abstract*— In wireless sensor networks (WSNs), the field information (e.g., temperature, humidity, airflow) is acquired via several battery-equipped wireless sensors and is relayed towards a sink node. As the size of the WSNs increases, it becomes inefficient (in terms of power consumption) to gather all information in a single sink. To tackle this problem, one can increase the number of sinks. The set of sensor nodes sending data to sink k is called commodity k. In this paper, we formulate the lexicographically optimal commodity lifetime routing problem. A stepwise centralized algorithm, called the lexicographically optimal commodity lifetime (LOCL) algorithm, is proposed which can obtain the optimal routing solution and lead to lexicographical fairness among commodity lifetimes. We then show that under certain assumptions, the lexicographical optimality among commodity lifetimes can be achieved by providing lexicographical optimality among node lifetimes. This motivates us to propose our second algorithm, called the lexicographically optimal node lifetime (LONL) algorithm, which suitable for practical implementation. Simulation results show that our proposed LOCL and LONL algorithms increase the normalized commodity and node lifetimes compared to the maximum lifetime with multiple sinks (MLMS) [1] and lexicographical max-min fair (LMM) [2] routing algorithms.**

Keywords: multiple sinks, wireless sensor networks, lexicographical optimality, routing flow.

I. Introduction

Recent advances in low power integrated circuits have sped up the development of various types of low cost wireless sensors, which are the building blocks of the wireless sensor networks (WSNs). In WSNs, each sensor node has the capability to sense the environment (e.g., temperature, pressure, light, acoustic) and process the data. In general, WSNs have an ad hoc topology and each node is capable of relaying the data towards the sink [3]. Since most of the sensor nodes are battery powered, one of the design objectives is to prolong the *lifetime* of the network [4]. There are various ways to define the lifetime of a WSN. It can be defined as the time at which the first node runs out of energy [5]. This time is equivalent to the time at which the first routing path is disconnected [6]. In [7], the lifetime is defined as the time at which the maximum number of times a certain data collection function can be carried out. In [8], [9], it is defined as the time at which a region within the WSN is not covered by any nodes.

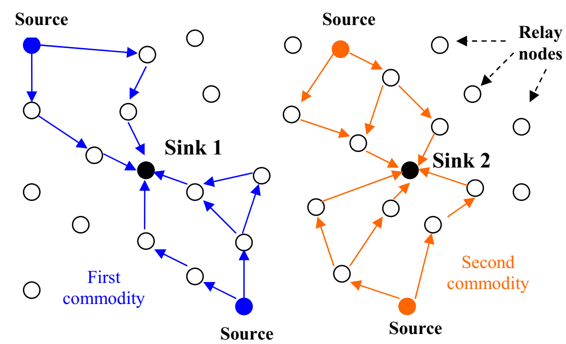


Fig. 1. A sample wireless sensor network with two sinks.

We now summarize some of the related works on lifetime maximization in WSNs. The maximum lifetime routing problem is formulated as a linear programming (LP) problem in [5], [10], and [11]. In [12], a distributed algorithm based on dual decomposition is proposed to solve the linear maximum lifetime problem. In [13], the maximum lifetime problem is extended by considering a variable-length TDMA (time division multiple access) frame in the MAC (medium access control) layer. In [14], the utility-lifetime tradeoff in maximum lifetime problem is studied by considering the source rates as variables in the system. In [15], the same problem is studied by considering the scheduling constraints of the link data rates. In [1], the lifetime problem is formulated as an LP problem for multicommodity networks. In [2], an iterative algorithm is proposed to obtain a lexicographic max-min node lifetime solution. In [16], an iterative centralized algorithm is proposed to find a Pareto-optimal routing solution for WSNs.

As the size of the network increases, it becomes inefficient (in terms of power consumption) and sometimes impossible (in terms of network capacity) to gather all information in a single sink node. To tackle this problem, one can increase the number of sinks [17]–[22]. A sample WSN with two sinks is illustrated in Fig. 1, with each source node sending data to the nearest sink. WSNs with multiple sinks have recently received increasing attention. In [17], a multi-sink WSN architecture is proposed where the network is partitioned into clusters. All the sources in a cluster are assigned to send data to the sink designated to that particular cluster. A multi-drain sensor network is considered in [18]. Data from each source are logged in two distinct drains in order to increase the resiliency of drain failure. In [19], the upper and lower bounds for the

Manuscript received October 10, 2007; revised March 30, 2008; accepted June 14, 2008. This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada. Part of this paper was presented at the *IEEE International Conference on Communications (ICC)*, Beijing, China, May 2008. The review of this paper was coordinated by Prof. Mohsen Guizani.

The authors are with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC V6T 1Z4, Canada, e-mail: {vahids, hamed, vincentw}@ece.ubc.ca.

optimal solution of the multiple sink problem are obtained. It is shown that the bounds are tight for networks with a large number of nodes. In [23], we studied the problem of fair resource allocation among different commodities in multiple sinks WSNs.

The set of nodes sending data to a particular sink is called a commodity. The lifetimes of commodities in a network with multiple sinks are not independent. By changing the routing flows, we can increase or decrease the lifetime for each commodity. One important issue is how to obtain a routing solution which provides fairness among different commodity lifetimes. This is the main focus in this work. To the best of our knowledge, there is no prior study on fair lifetime maximization for multi-commodity WSNs with multiple sinks. Our proposed algorithms are based on lexicographic optimization (see Section II-A) to achieve fairness. The contributions of this paper are as follows:

- We consider the commodity lifetime problem in a WSN with multiple sinks. We formulate this problem as a sequence of optimization problems.
- We propose a stepwise centralized algorithm, called the lexicographically optimal commodity lifetime (LOCL) algorithm, which lexicographically maximizes the lifetime of commodities. By using the LOCL algorithm, the average normalized commodity lifetime increases by 100% and 35% compared to maximum lifetime with multiple sinks (MLMS) [1] and lexicographical max-min fair (LMM) [2] algorithms when there are four sinks in the network, respectively.
- We show that, under certain assumptions, a lexicographically optimal commodity lifetime problem can be reduced to a lexicographically optimal node lifetime problem. This problem can be solved in polynomial time. We propose a distributed algorithm, called the lexicographically optimal node lifetime (LONL) algorithm, to solve this problem using dual decomposition technique. The LONL algorithm is simpler for implementation and more interesting for practical deployment.

The rest of this paper is organized as follows: In Section II, we present the problem formulation and describe our proposed LOCL algorithm. In Section III, we propose the LONL algorithm and provide its distributed implementation using the dual decomposition technique. The performance comparisons between LOCL, LONL, MLMS, and LMM algorithms are presented in Section IV. Conclusions are given in Section V.

II. Lexicographically Optimal Commodity Lifetime (LOCL) Algorithm

In this section, we first introduce the notations. The lifetime of the commodity and the lexicographically ordered commodity lifetime vectors are defined. We then present the stepwise LOCL algorithm.

A. System Model

Consider a WSN where the sensor nodes are randomly scattered over the coverage area. Each sensor node has limited energy. Nodes can cooperate to relay data towards the sinks. The commodities and the nodes can be considered as the *users* and *resources* of the system, respectively. Each sink uses the system resources to gather information within the sensing area. The source nodes in a commodity utilize the energy of the intermediate nodes in the network for a multi-hop transmission towards the sink. Since each node has battery powered, resources in the system are limited. Thus, a fair resource allocation is needed. Our objective is to obtain a routing flow which fairly shares the system resources among commodities. The lexicographic optimality is used as the notion of fairness among the lifetimes of the commodities.

Let $\mathcal{V}$ denote the set of sensor nodes and $\mathcal{C}$ denote the set of sinks collecting information from the network. The data rate generated by source node i to sink $k \in \mathcal{C}$ is denoted by S_i^k. The total data rate generated by source node i is denoted by S_i and is equal to $\sum_{k\in\mathcal{C}} S_i^k$. Let $\mathcal{N}_i$ be the set of neighbors of node $i \in \mathcal{V}$. Let x_{ij}^k denote the data rate of commodity $k \in \mathcal{C}$ transmitted from node i to node $j \in \mathcal{N}_i$. The aggregate data rate for the unidirectional logical link from node i to $j \in \mathcal{N}_i$ is denoted by x_{ij} and is equal to $\sum_{k\in\mathcal{C}} x_{ij}^k$. For notation simplicity, we stack up all x_{ij} and denote it as vector $\mathbf{x}$.

Let p_{ij} denote the power consumed in node $i \in \mathcal{V}$ for transmission of one bit information to node $j \in \mathcal{N}_i$. The maximum data rate between nodes i and j is denoted by R_{ij} depending on the capacity of the link and the MAC protocol being used. Let E_i represent the initial energy of node i. The *lifetime of sensor node* i, $T_i(\mathbf{x})$, is defined as follows:

$$T_i(\mathbf{x}) = \frac{E_i}{\sum_{j\in\mathcal{N}_i} p_{ij} \sum_{k\in\mathcal{C}} x_{ij}^k}. \quad (1)$$

The lifetime of commodity k is defined as the time at which the first path is disconnected between one of the sources and sink $k \in \mathcal{C}$. This time is equal to the time at which the first node carrying information for sink k runs out of its energy. The lifetime of commodity $k \in \mathcal{C}$ under data flow vector $\mathbf{x}$ is defined as follows:

$$T^k(\mathbf{x}) = \min\left\{ T_i(\mathbf{x}) \;\;|\;\; i \in \mathcal{V} \text{ and } \sum_{j\in\mathcal{N}_i} x_{ij}^k > 0 \right\}. \quad (2)$$

A vector $\mathbf{T} = (T^1, T^2, \cdots, T^{|\mathcal{C}|})$ is called a *lexicographically ordered* commodity lifetime vector if $T^1 \leq T^2 \leq \cdots \leq T^{|\mathcal{C}|}$. That is, all the elements in the vector are sorted (or arranged) in ascending order. The commodity lifetime vector $\hat{\mathbf{T}}$ is *lexicographically greater* than vector $\tilde{\mathbf{T}}$ if and only if there exists i such that $\hat{T}^i > \tilde{T}^i$, and for all $j < i$ we have $\hat{T}^i = \tilde{T}^j$. As an example, if $\hat{\mathbf{T}} = (2,2,5,8)$ and $\tilde{\mathbf{T}} = (2,2,4,9)$, then $\hat{\mathbf{T}}$ is lexicographically greater than $\tilde{\mathbf{T}}$. Two vectors are *lexicographically equal* if all of the elements of these two vectors are equal.

A vector is *lexicographically optimal* in a set if it is the lexicographically greatest vector in the set. As an example, for the set $\{(1,3,7,8),(2,2,5,8),(2,2,4,9),(2,2,3,10)\}$, the vector $(2,2,5,8)$ is the lexicographically optimal vector. Note that for any compact set of $\mathbb{R}^m$, there exists only one lexicographically optimal vector [24].

B. Problem Formulation

In the LOCL algorithm, the objective is to determine the routing paths and flow rates (i.e., data flow vector $\mathbf{x}$) which lead to the lexicographically optimal feasible lifetime vector. The LOCL algorithm is a stepwise algorithm. In the first step, the minimum commodity lifetime in the network is maximized. This step may have an infinite number of optimal routing solutions. In the second step, among the solutions from the first step, a solution is chosen which maximizes the second minimum commodity lifetime. The second step may also have an infinite number of optimal solutions.

In general, in step n of the LOCL algorithm, among the solutions from step $(n-1)$, the solution is chosen which maximizes the nth minimum commodity lifetime. In other words, in step n, the nth minimum commodity lifetime is maximized while all lower commodity lifetimes are being maximized. The routing solution in the last step lexicographically maximizes the lifetime of all commodities. Note that the number of steps is equal to or less than the number of commodities. The optimal routing solution in the last step is the lexicographically optimal commodity lifetime routing (LOCLR) solution:

Definition 1. A routing flow is LOCLR if the vector of lifetimes of commodities under this routing flow is the lexicographically greatest feasible commodity lifetime vector.

LOCLR is the solution of the LOCL algorithm or equivalently the optimal solution of the last step of this algorithm. The last step of the LOCL algorithm may have more than one (i.e., an infinite number of) optimal solutions.

Proposition 1. The lexicographically optimal commodity lifetime vector is unique but the LOCLR solution is not necessarily unique.

As we mentioned earlier, the lexicographically optimal commodity lifetime vector in a set is unique [24]. Results in [2] showed that in a lifetime maximization problem, it is possible to have an infinite number of optimal solutions. An example can be found in [25]. This example can be extended for the networks with multiple sinks or commodities.

C. LOCL Algorithm: First Step

We now present the first step of the LOCL algorithm and describe how it can be converted to a linear mixed-integer programming problem. In the first step, the minimum commodity lifetime is maximized. The problem is as follows:

$$\begin{aligned}
&\text{maximize} && \min_{k\in\mathcal{C}} \left\{ T^k(\mathbf{x}) \right\} \\
&\text{subject to} && \sum_{j\in\mathcal{N}_i} (x_{ij}^k - x_{ji}^k) = S_i^k, \quad \forall\, i\in\mathcal{V},\ \forall\, k\in\mathcal{C} \\
& && \sum_{k\in\mathcal{C}} x_{ij}^k \le R_{ij}, \qquad \forall\, i\in\mathcal{V},\, \forall\, j\in\mathcal{N}_i \\
& && x_{ij}^k \ge 0, \quad \forall\, i\in\mathcal{V},\ \forall\, j\in\mathcal{N}_i,\ \forall\, k\in\mathcal{C}. \qquad (3)
\end{aligned}$$

The first set of constraints is the flow conservation for each commodity in all the nodes. The second set of constraints is the data rate limits on each link. To obtain a linear objective function, we introduce an auxiliary scalar variable t, which is a lower bound for the lifetime of minimum commodity:

$$t \le \min_{\mathbf{x},\ k\in\mathcal{C}} \left\{ T^k(\mathbf{x}) \right\} \quad \Rightarrow \quad t \le T^k(\mathbf{x}), \qquad \forall\, k\in\mathcal{C}. \quad (4)$$

The objective is to maximize t. Substituting (2) in (4) and (4) in (3), it becomes:

$$\begin{aligned}
&\text{maximize} && t \\
&\text{subject to} && t \le T^k, && \forall\, k\in\mathcal{C} \\
& && T^k = \min_{i\in\mathcal{V}} \Big\{ T_i \mid T_i = E_i / (\sum_{j\in\mathcal{N}_i} p_{ij} \sum_{m\in\mathcal{C}} x_{ij}^m) \\
& && \quad \text{and} \sum_{j\in\mathcal{N}_i} x_{ij}^k > 0 \Big\}, && \forall\, k\in\mathcal{C} \\
& && \sum_{j\in\mathcal{N}_i} (x_{ij}^k - x_{ji}^k) = S_i^k, && \forall\, i\in\mathcal{V},\ \forall\, k\in\mathcal{C} \\
& && \sum_{k\in\mathcal{C}} x_{ij}^k \le R_{ij}, && \forall\, i\in\mathcal{V},\, \forall\, j\in\mathcal{N}_i \\
& && x_{ij}^k \ge 0, \quad \forall\, i\in\mathcal{V},\, \forall\, j\in\mathcal{N}_i,\ \forall\, k\in\mathcal{C}. && (5)
\end{aligned}$$

The second constraint in problem (5) can be replaced by:

$$T^k \le \frac{E_i}{\sum_{j\in\mathcal{N}_i} \sum_{m\in\mathcal{C}} p_{ij} x_{ij}^m} \quad \text{if} \sum_{j\in\mathcal{N}_i} x_{ij}^k > 0,\ \forall\, i\in\mathcal{V}. \quad (6)$$

We replace T^k and t by their respective inverses: $q^k = 1/T^k$ and $q = 1/t$. The objective is changed from maximizing t to minimizing q. Problem (5) can now be written as:

$$\begin{aligned}
&\text{minimize} \quad q \\
&\text{subject to} \\
&q^k \le q, && \forall\, k\in\mathcal{C} \\
&\sum_{j\in\mathcal{N}_i} \sum_{m\in\mathcal{C}} p_{ij} x_{ij}^m \le E_i q^k, \text{if} \sum_{j\in\mathcal{N}_i} x_{ij}^k > 0, \forall i\in\mathcal{V}, \forall k\in\mathcal{C} \\
&\sum_{j\in\mathcal{N}_i} (x_{ij}^k - x_{ji}^k) = S_i^k, && \forall\, i\in\mathcal{V},\ \forall\, k\in\mathcal{C} \qquad (7) \\
&\sum_{k\in\mathcal{C}} x_{ij}^k \le R_{ij}, && \forall\, i\in\mathcal{V},\, \forall\, j\in\mathcal{N}_i \\
&x_{ij}^k \ge 0, && \forall\, i\in\mathcal{V},\ \forall\, j\in\mathcal{N}_i,\ \forall\, k\in\mathcal{C}.
\end{aligned}$$

The second constraint in problem (7) is conditional. To obtain a closed-form for this constraint, we multiply both sides of the inequality by $(\sum_{j\in\mathcal{N}_i} x_{ij}^k)$. The new constraint is:

$$\left(\sum_{j\in\mathcal{N}_i} x_{ij}^k \right) \left(\sum_{j\in\mathcal{N}_i} \sum_{m\in\mathcal{C}} p_{ij} x_{ij}^m \right) \le \left(\sum_{j\in\mathcal{N}_i} x_{ij}^k \right) E_i q^k, \quad \forall\, i\in\mathcal{V}, \quad k\in\mathcal{C}. \quad (8)$$

In (8), if $\sum_{j\in\mathcal{N}_i} x_{ij}^k = 0$, then both left hand side and right hand side become 0. If $\sum_{j\in\mathcal{N}_i} x_{ij}^k > 0$, then constraint (8) is equivalent to: $\sum_{j\in\mathcal{N}_i} \sum_{m\in\mathcal{C}} p_{ij} x_{ij}^m \le E_i q^k$. Therefore, constraint (8) is equivalent to the second constraint in (7). The constraint in (8) is a nonconvex constraint. We use a series of change of variables to replace this constraint with several linear constraints. We introduce an auxiliary boolean variable:

$$b_i^k = \begin{cases} 0, & \text{if } \sum_{j\in\mathcal{N}_i} x_{ij}^k = 0 \\ 1, & \text{if } \sum_{j\in\mathcal{N}_i} x_{ij}^k > 0, \end{cases} \quad (9)$$

where b_i^k is equal to 1 when node i carries information of commodity k. We can then map the new boolean variable to rate variables as follows:

$$\frac{\sum_{j\in\mathcal{N}_i} x_{ij}^k}{\sum_{j\in\mathcal{N}_i} R_{ij}} \leq b_i^k, \qquad \forall\, i\in\mathcal{V},\quad k\in\mathcal{C}. \tag{10}$$

Constraint (8) can now be written as follows:

$$b_i^k \sum_{j\in\mathcal{N}_i}\sum_{m\in\mathcal{C}} p_{ij}x_{ij}^m \leq b_i^k E_i q^k, \qquad \forall\, i\in\mathcal{V},\quad k\in\mathcal{C}. \tag{11}$$

This constraint is still nonlinear. We use a linearization technique and convert this constraint to a set of linear constraints. Details of the linearization technique can be found in Appendix A. We define two new variables:

$$\begin{aligned} \nu_i^k &= b_i^k \sum_{j\in\mathcal{N}_i}\sum_{m\in\mathcal{C}} p_{ij}x_{ij}^m, \quad \forall\, i\in\mathcal{V},\quad k\in\mathcal{C} \\ \gamma_i^k &= b_i^k E_i q^k, \qquad\qquad\qquad \forall\, i\in\mathcal{V},\quad k\in\mathcal{C}. \end{aligned} \tag{12}$$

Constraint (11) can be written as:

$$\nu_i^k \leq E_i \gamma_i^k, \qquad \forall\, i\in\mathcal{V},\quad k\in\mathcal{C}. \tag{13}$$

A set of constraints is added for each new variable and node i and commodity k:

$$\begin{gathered} 0 \leq \nu_i^k \leq \textstyle\sum_{j\in\mathcal{N}_i}\sum_{m\in\mathcal{C}} p_{ij}x_{ij}^m, \\ \textstyle\sum_{j\in\mathcal{N}_i}\sum_{m\in\mathcal{C}} p_{ij}x_{ij}^m - (1-b_i^k)P_i^{max} \leq \nu_i^k, \\ \nu_i^k \leq b_i^k P_i^{max}, \\ 0 \leq \gamma_i^k \leq q^k, \\ q^k - q_{max}^k(1-b_i^k) \leq \gamma_i^k \leq q_{max}^k b_i^k, \end{gathered} \tag{14}$$

where $P_i^{max} = \sum_{j\in\mathcal{N}_i} p_{ij}R_{ij}$, and q_{max}^k is a loose upper bound for q^k. We re-write problem (7) with new variables. In this problem, the variables are q, q^k, x_{ij}^k, x_{ij}^m, b_i^k, ν_i^k, and γ_i^k, while S_i^k, R_{ij}, E_i, p_{ij}, P_i^{max}, and q_{max}^k are the constants:

$$\begin{aligned} &\text{minimize} \quad q \\ &\text{subject to} \\ &q^k \leq q, && \forall\, k\in\mathcal{C} \\ &\sum_{j\in\mathcal{N}_i}(x_{ij}^k - x_{ji}^k) = S_i^k, && \forall\, i\in\mathcal{V},\quad k\in\mathcal{C} \\ &\frac{\sum_{j\in\mathcal{N}_i} x_{ij}^k}{\sum_{j\in\mathcal{N}_i} R_{ij}} \leq b_i^k, && \forall\, i\in\mathcal{V},\quad k\in\mathcal{C} \\ &\nu_i^k \leq E_i\gamma_i^k, && \forall\, i\in\mathcal{V},\quad k\in\mathcal{C} \\ &0 \leq \nu_i^k \leq \sum_{j\in\mathcal{N}_i}\sum_{m\in\mathcal{C}} p_{ij}x_{ij}^m, && \forall\, i\in\mathcal{V},\quad k\in\mathcal{C} \\ &\sum_{j\in\mathcal{N}_i} p_{ij}\sum_{m\in\mathcal{C}} x_{ij}^m - (1-b_i^k)P_i^{max} \leq \nu_i^k, && \forall\, i\in\mathcal{V},\ k\in\mathcal{C} \\ &\sum_{k\in\mathcal{C}} x_{ij}^k \leq R_{ij}, && \forall\, i\in\mathcal{V},\quad j\in\mathcal{N}_i \\ &\nu_i^k \leq b_i^k P_i^{max}, && \forall\, i\in\mathcal{V},\quad k\in\mathcal{C} \\ &0 \leq \gamma_i^k \leq q^k, && \forall\, i\in\mathcal{V},\quad k\in\mathcal{C} \\ &q^k - q_{max}^k(1-b_i^k) \leq \gamma_i^k \leq q_{max}^k b_i^k, && \forall\, i\in\mathcal{V},\quad k\in\mathcal{C} \\ &x_{ij}^k \geq 0,\ b_i^k\in\{0,1\}, && \forall\, i\in\mathcal{V},\quad j\in\mathcal{N}_i,\quad k\in\mathcal{C}. \end{aligned} \tag{15}$$

Because of the linearity of the objective function, it is possible to have an infinite number of optimal lifetime solutions. To obtain a unique set of commodities with minimum lifetime in this step and the subsequent steps, we use the regularization method. Details of the regularization method can be found in Appendix B. The regularization term used in problem (15) is the Euclidean norm of commodity lifetime vector which is $\sum_{k\in\mathcal{C}}\left(q^k\right)^2$. Therefore, the new objective of problem (15) is:

$$\text{minimize} \quad q + \delta\sum_{k\in\mathcal{C}}\left(q^k\right)^2, \tag{16}$$

where δ is the regularization parameter. The objective function is now quadratic. The work in [26] and [27] proved that when δ is less than a certain threshold, the optimal solutions of the regularized problem form a subset of the optimal solutions of the problem before regularization.

D. LOCL Algorithm: Subsequent Steps

The first step in the LOCL algorithm is a linear mixed-integer program (MIP). The feasible set in the second step of the LOCL algorithm is the optimal solution set of the first step. Similarly, the feasible set in the third step is the optimal solution of the second step, and so on. We call the mixed integer programming problem in step n as MIP-n. The following lemma characterizes the feasible region for the problem in step n:

Lemma 1. The feasible region of problem MIP-n is nonempty if there exists at least one path between each source and its associated sink.

The proof of this lemma is presented in Appendix C. This lemma guarantees that the feasible region in each step, which is the optimal solution set in the previous step, is nonempty.

Assume that the minimum lifetime in the first step (problem (15)) is $T^{1\star}$ and the optimal value (inverse of minimum commodity lifetime) is $q^{1\star}$. Let $\mathcal{P}_1$ be the set of commodities that the first constraint is active in problem (15). In the second step, the minimum lifetime among all the commodities except the members of $\mathcal{P}_1$ (i.e., $\mathcal{C}\backslash\mathcal{P}_1$) is maximized subject to the condition that the lifetime of the commodities in $\mathcal{P}_1$ is also being maximized. The problem in the second step is similar to problem (3) while the objective is modified to be as follows:

$$\text{maximize} \min_{k\in\mathcal{C}\backslash\mathcal{P}_1} \; T^k(\mathbf{x}).$$

Also, there is a constraint on the maximization of minimum lifetime. The constraint is:

$$T^l \geq T^{1\star}, \qquad \forall\, l\in\mathcal{P}_1.$$

The problem in the nth step can be formulated with similar changes in the objective function and by including the additional constraints. Let $T^{h\star}$ denote the maximum achievable value for the hth minimum commodity lifetime (obtained from the hth step). Let $\mathcal{P}_h$ denote the set of commodities that their lifetimes are equal to $T^{h\star}$ in the hth step. We have $\mathcal{P}_h = \{k \mid T^k = T^{h\star} \text{ and } k\in\mathcal{C}\backslash\bigcup_{l=1}^{h-1}\mathcal{P}_l\}$. The problem in the nth step is as follows:

$$\begin{aligned}
\text{maximize} \quad & \min_{k\in\mathcal{C}\setminus\bigcup_{h=1}^{n-1}\mathcal{P}_h} \left\{T^k(\mathbf{x})\right\} \\
\text{subject to} \quad & T^l \geq T^{h\star}, \qquad \forall\, l \in \mathcal{P}_h, \quad h=1,\ldots,n-1 \\
& \sum_{j\in\mathcal{N}_i} (x_{ij}^k - x_{ji}^k) = S_i^k, \quad \forall\, i\in\mathcal{V}, \quad k\in\mathcal{C} \\
& \sum_{k\in\mathcal{C}} x_{ij}^k \leq R_{ij}, \quad \forall\, i\in\mathcal{V}, \quad j\in\mathcal{N}_i \\
& x_{ij}^k \geq 0, \qquad \forall\, i\in\mathcal{V}, \quad j\in\mathcal{N}_i, \quad k\in\mathcal{C}.
\end{aligned} \tag{17}$$

By letting $q^{h\star} = 1/T^{h\star}$, the same series of changes that are applied to problem (3) can be applied to problem (17). The mixed integer programming problem in step n (i.e., MIP-n) is as follows:

$$\begin{aligned}
& \text{minimize} \quad q \\
& \text{subject to} \\
& q^k \leq q, \qquad \forall\, k\in\mathcal{C}\setminus\bigcup_{h=1}^{n-1}\mathcal{P}_h \\
& q^l \leq q^{h\star}, \qquad \forall\, l\in\mathcal{P}_h, \quad h=1,\ldots,n-1 \\
& \sum_{j\in\mathcal{N}_i}(x_{ij}^k - x_{ji}^k) = S_i^k, \qquad \forall\, i\in\mathcal{V}, \quad k\in\mathcal{C} \\
& \frac{\sum_{j\in\mathcal{N}_i} x_{ij}^k}{\sum_{j\in\mathcal{N}_i} R_{ij}} \leq b_i^k, \qquad \forall\, i\in\mathcal{V}, \quad k\in\mathcal{C} \\
& \nu_i^k \leq E_i \gamma_i^k, \qquad \forall\, i\in\mathcal{V}, \quad k\in\mathcal{C} \\
& 0 \leq \nu_i^k \leq \sum_{j\in\mathcal{N}_i}\sum_{m\in\mathcal{C}} p_{ij} x_{ij}^m, \quad \forall\, i\in\mathcal{V}, \quad k\in\mathcal{C} \\
& \sum_{j\in\mathcal{N}_i} p_{ij} \sum_{m\in\mathcal{C}} x_{ij}^m - (1-b_i^k)P_i^{max} \leq \nu_i^k, \ \forall\, i\in\mathcal{V}, \ k\in\mathcal{C} \\
& \sum_{k\in\mathcal{C}} x_{ij}^k \leq R_{ij}, \qquad \forall\, i\in\mathcal{V}, \quad j\in\mathcal{N}_i \\
& \nu_i^k \leq b_i^k P_i^{max}, \qquad \forall\, i\in\mathcal{V}, \quad k\in\mathcal{C} \\
& 0 \leq \gamma_i^k \leq q^k, \qquad \forall\, i\in\mathcal{V}, \quad k\in\mathcal{C} \\
& q^k - q_{max}^k(1-b_i^k) \leq \gamma_i^k \leq q_{max}^k b_i^k, \quad \forall\, i\in\mathcal{V}, \quad k\in\mathcal{C} \\
& x_{ij}^k \geq 0, \ b_i^k \in \{0,1\}, \qquad \forall\, i\in\mathcal{V}, \quad j\in\mathcal{N}_i, \quad k\in\mathcal{C}.
\end{aligned} \tag{18}$$

Algorithm 1 shows the stepwise LOCL algorithm to determine the LOCLR solution.

Algorithm 1 Stepwise LOCL Algorithm (Centralized)

1: Set $n = 1$
2: **While** $\bigcup_{h=1}^{n}\mathcal{P}_h \neq \mathcal{C}$
3: Solve MIP-n.
4: Set $q^{n\star}$:= Optimal value of MIP-n.
5: Set $\mathcal{P}_n := \{k \mid q^k = q^{n\star} \text{ and } k\in\mathcal{C}\setminus\bigcup_{h=1}^{n-1}\mathcal{P}_h\}$.
6: Set $n := n+1$.
7: **End**

The MIP-1 is problem (15). The MIP-n (for $n > 1$) is problem (18). The number of steps in the stepwise LOCL algorithm is less than or equal to the number of commodities.

Algorithm 1 is NP-hard in general. There are efficient commercial software (such as CPLEX [28] and MOSEK [29]) to solve linear mixed-integer problems. Most of them use branch-and-bound algorithm [30]. The linear mixed-integer problem in the n^{th} step of Algorithm 1 has $|\mathcal{C}||\mathcal{V}|$ binary variables, $|\mathcal{C}|(2|\mathcal{V}|+|\mathcal{L}|)$ real variables, $|\mathcal{C}|(8|\mathcal{V}|+n)$ inequality constraints, and $|\mathcal{C}||\mathcal{V}|$ equality constraints, where $\mathcal{L}$ is the set of links in the network. Notice that the computational complexity of a linear mixed-integer problem depends only on the number of its integer (in our case binary) variables, but not the number of real variables [31]. Thus, problem (18) can easily be solved in practice for small-scale and medium-scale WSNs. Next, we propose our second algorithm which is particularly useful for practical deployments of large scale WSNs.

III. Lexicographically Optimal Node Lifetime (LONL) Algorithm

In (2), we defined the lifetime of a commodity k to be the time at which the first node carrying information in commodity k runs out of its energy. In a WSNs with a large number of sensor nodes, if the sensor nodes and the sinks are uniformly distributed in the coverage area and all the sensor nodes have the same amount of initial energy, then the first node which runs out of its energy in commodity k is most likely to be one of those that is the neighbor of sink k. In the first stage, some sensor nodes that are closer to the sinks will most likely run out of energy. These nodes, which are in different commodities and most likely neighbors of the sinks, usually carry information belonging to their commodities. Therefore, their lifetimes are independent. If the above assumptions are valid, then the lexicographically optimal commodity lifetime problem can be reduced to a lexicographical optimal node lifetime problem. In this section, we show how to solve the latter in a *distributed* manner. The proposed algorithm is called the *lexicographically optimal node lifetime* (LONL) algorithm. Compared to the LOCL algorithm in the previous section, the LONL algorithm is simpler for implementation and more practical for deployment.

A. System Model

Given the list of nodes' lifetimes from equation (1), we can determine the corresponding *lexicographically ordered node lifetime vector* such that $T_1 \leq T_2 \leq \cdots \leq T_{|\mathcal{V}|}$. The concepts of *lexicographical ordering* and *lexicographically greater than* are the same as what we defined in Section II-A.

Definition 2. A routing is called *Lexicographically Optimal Node Lifetime Routing* (LONLR) if the node lifetime vector under this routing flow is the lexicographically greatest among all feasible node lifetime vectors.

We call the inverse of the lifetime of each node the *normalized power consumption* of the node. Let g_i denote the normalized power consumption of node i. The normalized power consumption of node i under data flow vector $\mathbf{x} = \{x_{ij}^k\}$ is as follows:

$$g_i(\mathbf{x}) = \frac{\sum_{j\in\mathcal{N}_i} p_{ij} \sum_{k\in\mathcal{C}} x_{ij}^k}{E_i}. \tag{19}$$

For simplicity, the vector $\mathbf{g} = \{g_i\}$ is used to denote the normalized power consumptions of all the nodes.

Definition 3. A routing flow is *Lexicographically Optimal Normalized Power Consumption Routing* (LONPR) flow if the normalized power consumption vector under this routing flow is the *lexicographically smallest* feasible normalized power consumption vector.

Using the concept of LONPR, we can express the following lemma:

Lemma 2. The nodes' lifetimes vectors under LONLR flow (Definition 2) and LONPR flow (Definition 3) are lexicographically equal.

The proof of this lemma is provided in Appendix D. It shows that the ith minimum lifetime is equal to the inverse of the ith maximum normalized power consumption for all values of $i = 1, \ldots, |\mathcal{V}|$.

We now formulate a convex optimization problem, which leads to the LONPR solution. Consider the following problem:

$$\begin{aligned} \text{minimize} \quad & \sum_{i\in\mathcal{V}} f(g_i) \\ \text{subject to} \quad & \sum_{j\in\mathcal{N}_i} (x_{ij}^k - x_{ji}^k) = S_i^k, \quad \forall\, i \in \mathcal{V}, \quad k \in \mathcal{C} \\ & \sum_{j\in\mathcal{N}_i} p_{ij} \sum_{k\in\mathcal{C}} x_{ij}^k = E_i g_i, \quad \forall\, i \in \mathcal{V} \\ & \sum_{k\in\mathcal{C}} x_{ij}^k \leq R_{ij}, \quad \forall\, i \in \mathcal{V}, \quad j \in \mathcal{N}_i \\ & x_{ij}^k \geq 0, \quad \forall\, i \in \mathcal{V}, \quad j \in \mathcal{N}_i, \quad k \in \mathcal{C}, \end{aligned} \tag{20}$$

where the function $f(g_i)$ is a convex function. Problem (20) has $|\mathcal{V}|(|\mathcal{C}|+1)$ (real) variables, $|\mathcal{L}|(|\mathcal{C}|+1)$ inequality constraints, and $|\mathcal{V}|(|\mathcal{C}|+1)$ equality constraints. Since all the constraints are linear and the objective function is convex, the optimization problem in (20) is a convex programming problem. There are several schemes that can be used to solve convex optimization problems. They include the *gradient projection methods*, *interior point method*, and *primal-dual method* [32]. Note that the convex programming algorithms have polynomial complexity. That is, the run time is no greater than a polynomial function of the problem size $(2|\mathcal{V}|+|\mathcal{L}|)(|\mathcal{C}|+1)$. The theorem below relates problem (20) and the LONPR solution.

Theorem 1. If $h(x)$ is a differentiable increasing convex function, then the routing solution of problem (20) with $f(x) = \big(h(x)\big)^{\gamma}$ approaches the LONPR solution as $\gamma \to \infty$.

The proof of this theorem is presented in Appendix E. It can be shown that the LONPR flow solution is not necessarily unique. However, all of the solutions lead to a unique lexicographically ordered node lifetime vector. Based on Theorem 1, the optimal solution of problem (20) is the LONPR solution. Based on Lemma 2, the LONPR and LONLR flow solutions lead to the same node lifetime vector. Therefore, the solution of problem (20) is indeed the LONLR solution. Problem (20) is a tractable convex problem which can be solved in polynomial time. We now propose a distributed solution for this problem.

B. Distributed Implementation

In this section, we present a distributed algorithm to solve problem in (20). The technique that we use is *dual decomposition* [32], which has also been used in [12], [33]. Problem (20) is referred as the primal problem. We first introduce Lagrange multipliers λ_i^k for the first equality constraints in (20). The other constraints are local constraints in each node and do not need to be relaxed. The Lagrangian function $L(\mathbf{g}, \mathbf{x}, \boldsymbol{\lambda})$ is:

$$\begin{aligned} & L(\mathbf{g}, \mathbf{x}, \boldsymbol{\lambda}) \\ & = \sum_{i\in\mathcal{V}} f(g_i) + \sum_{i\in\mathcal{V}} \sum_{k\in\mathcal{C}} \lambda_i^k \left(\sum_{j\in\mathcal{N}_i} \left(x_{ij}^k - x_{ji}^k\right) - S_i \right) \\ & = \sum_{i\in\mathcal{V}} \left(f(g_i) + \sum_{j\in\mathcal{N}_i} \sum_{k\in\mathcal{C}} x_{ij}^k \left(\lambda_i^k - \lambda_j^k\right) - \sum_{k\in\mathcal{C}} \lambda_i^k S_i \right). \end{aligned}$$

From the Lagrangian, the dual function and the dual problem can be defined. A subgradient algorithm [32] can be used to solve the dual problem. The subgradient algorithm is an iterative algorithm. In iteration τ, given $\lambda_i^k(\tau)$, $\lambda_j^k(\tau)$ for $j \in \mathcal{N}_i$, and the local information (p_{ij}, E_i, R_{ij}), each node i updates $x_{ij}^k(\tau)$ and $g_i^k(\tau)$ by solving the following problem:

$$\begin{aligned} \text{minimize} \quad & f(g_i(\tau)) + \sum_{j\in\mathcal{N}_i} \sum_{k\in\mathcal{C}} x_{ij}^k(\tau) \left(\lambda_i^k(\tau) - \lambda_j^k(\tau)\right) \\ & - \sum_{k\in\mathcal{C}} \lambda_i^k(\tau) S_i \\ \text{subject to} \quad & \sum_{j\in\mathcal{N}_i} p_{ij} \sum_{k\in\mathcal{C}} x_{ij}^k(\tau) = E_i g_i(\tau), \\ & \sum_{k\in\mathcal{C}} x_{ij}^k(\tau) \leq R_{ij}, \quad \forall\, j \in \mathcal{N}_i \\ & x_{ij}^k(\tau) \geq 0, \quad \forall\, j \in \mathcal{N}_i, \quad k \in \mathcal{C}. \end{aligned} \tag{21}$$

Algorithm 2 shows the distributed LONL algorithm performed in node i.

Algorithm 2 LONL Algorithm in Node i (Distributed)

Set $\tau := 1$
While not converged
  Solve problem (21) to obtain $g_i(\tau)$ and $x_{ij}^k(\tau)$ for $j \in \mathcal{N}_i$, $k \in \mathcal{C}$.
  Exchange the values of $x_{ij}^k(\tau)$ with neighboring nodes $j \in \mathcal{N}_i$.
  Set $\lambda_i^k(\tau+1) := \lambda_i^k(\tau) - \mu(\tau) \Big(S_i - \sum_{j\in\mathcal{N}_i} \big(x_{ij}^k(\tau) - x_{ji}^k(\tau) \big) \Big)$.
  Exchange the values of $\lambda_i^k(\tau+1)$ with neighboring nodes $j \in \mathcal{N}_i$.
  $\tau := \tau + 1$.
End

In the above algorithm, $\mu(\tau)$ is a positive diminishing step size and is chosen as $\frac{1}{\tau+1}$.

IV. Performance Evaluation

In this section, we evaluate the performance of our proposed LOCL and LONL algorithms. We assume a deterministic path loss model. The power consumed for transmission of one bit from node i to node j (i.e., p_{ij}) is $\eta_1 + \eta_2 d_{ij}^4$, where d is the physical distance. We choose $\eta_1 = 1$ and $\eta_2 = 0.1$. The regularization coefficient is set to $\delta = 10^{-5}$. Each source generates 1 kbps of information. The initial energy of each source node E_S is 3 J while the initial energy of each intermediate node i (i.e., E_i) is 1 J. Nodes use TDMA to access the shared communication channel. The time slots are assigned equally and statically to the different links, which have interference with each other. The maximum link rate R_{ij} is 250 kbps for all logical links.

A. Performance of LOCL Algorithm

In the first experiment, we show the performance of employing multiple sinks. There are 30 sensor nodes randomly deployed in a 50m $\times$ 50m square field. The transmission range of each node is 10m. Eight nodes are randomly chosen as the source nodes. Each source sends data to its (physically) closest sink. Fig. 2(a) shows the normalized minimum commodity lifetime in the network versus the number of sinks. Each sink is located in one of the four corners in the field. Results are averaged over 100 simulation runs. The values are normalized with respect to the lifetime of the network with only one sink. Simulation results depict that the lifetime increases almost linearly as the number of sinks is increased. It can be seen that the minimum commodity lifetime in the network when there are four sinks is almost 500% more compared to the network with one sink. The benefit of increasing the number of sinks is evident. Fig. 2(b) shows the number of steps performed in the LOCL algorithm versus the number of sinks in the network. When there are four sinks in the network, the average number of steps is about 3.5.

Next, we investigate the performance of the LOCL algorithm with a different number of sources. The other simulation settings are the same as in the previous experiment. Fig. 3 shows the average normalized commodity lifetime for the commodities under different number of sources. When the number of sources increases, the lifetime of each commodity decreases.

In this experiment, we assume that there are four different types of sources in the network, two sensors of each type. For example, there can be two temperature sensors, two pressure sensors, two humidity meter sensors, and two airflow sensors. One sink is assigned to gather information for each data type. Sensors are then required to send data to the corresponding sink. Fig. 4 shows the normalized lifetime of the commodities. We compare the results when there are different types of sensors and the case that each source sends data to the nearest sink. When there are different types of sources and all sources are randomly deployed in the sensor area, there are more common nodes among commodities. The correlation between different commodities increases. Also, the data packets are traversed over longer paths. Therefore, the lifetime of commodities decreases compared to the case where each node sends data to the nearest sink.

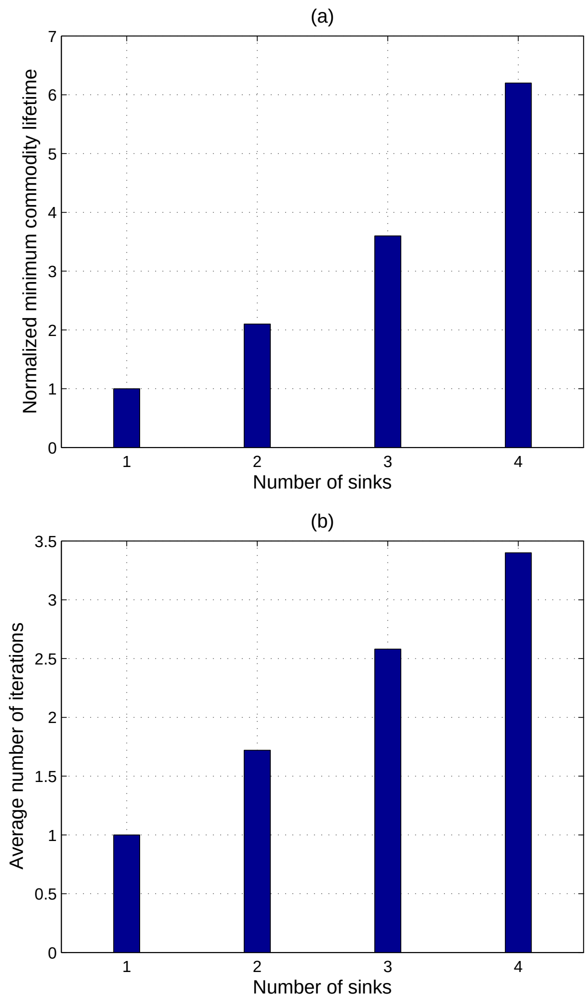


Fig. 2. LOCL algorithm for network with different number of sinks: (a) Minimum commodity lifetime; (b) Average number of iterations in LOCL algorithm.

B. Performance Comparisons between LOCL, LONL, MLMS, and LMM Algorithms

To perform the simulation for the LONL algorithm, we need to choose function $h(g_i)$ and parameter γ. We set $h(g_i) = g_i$. To choose γ, we set up an experiment. The network topology is similar to the network used in the first experiment with two sinks. We increment the value of γ starting from 1 and solve problem (20). Fig. 5 shows the average normalized minimum lifetime in the network for 100 simulation runs versus different values of γ. It can be seen that when $\gamma > 8$, the difference between consecutive lifetime values is less than 1%. We choose $\gamma = 10$ for the subsequent simulation runs. The objective function in problem (20) is chosen as $f(g_i) = g_i^{\gamma}$.

To compare our LOCL and LONL algorithms with the existing algorithms in the literature, we implemented the algorithms proposed in [1] and [2]. We solve the corresponding mixed integer programming problems by using the MOSEK [29] optimization toolbox. There are four sinks and eight sources in the network. In [1], the problem of network lifetime

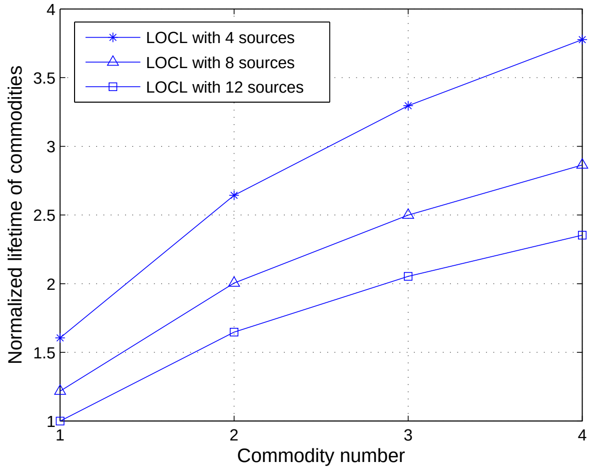


Fig. 3. Lifetime of commodities for LOCL algorithm with different number of sources.

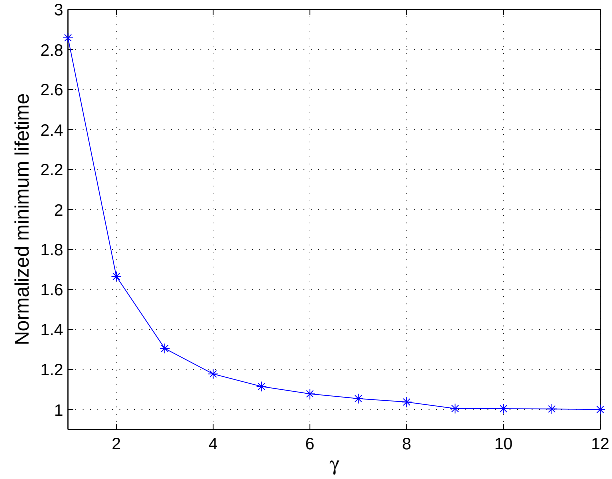


Fig. 5. Average normalized minimum lifetime in LONL algorithm versus γ.

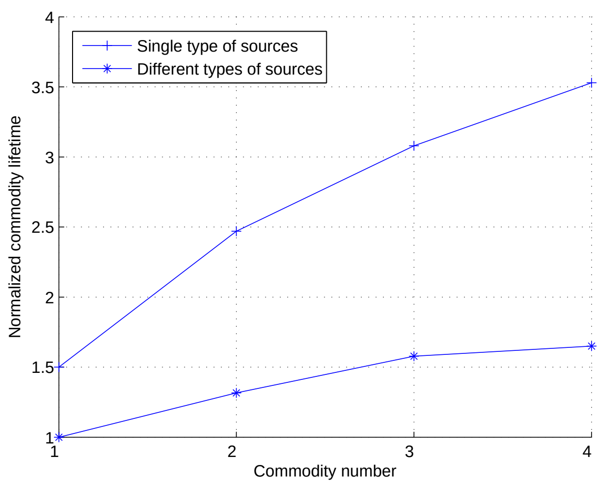


Fig. 4. Lifetime of commodities for LOCL algorithm when there are single and different types of sinks.

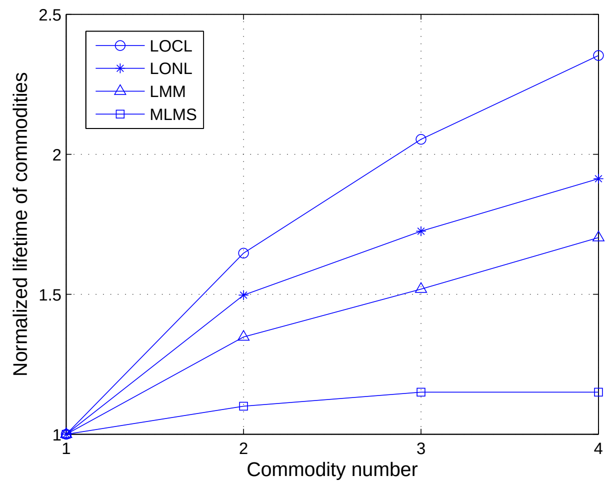


Fig. 6. Lifetime of commodities for LOCL, LONL, LMM, and MLMS algorithms in a network with four sinks.

maximization is modeled as a concurrent multi-commodity flow optimization problem. The problem can be extended to consider the multi-sink case. We modify this algorithm and assume that each source sends data to the closest sink. Note that this makes the comparison to be fair as in all other cases for multi-sink WSNs. When there is one sink, this problem is equal to the maximum lifetime routing problem proposed in [5]. We call the modified algorithm the maximum Lifetime routing for network with Multiple Sinks (MLMS).

A lexicographical max-min fair (LMM) algorithm is proposed in [2]. This algorithm determines a schedule for the routing flows. We extend this algorithm for WSNs with multiple sinks. We compare the results of LOCL with the first routing flow of LMM method. Notice that the LOCL algorithm provides one routing flow but not a schedule of routing flows for the network. Since both LOCL and LONL algorithms provide one routing flow, we compare these algorithms with the first routing flow in the schedule of LMM.

Fig. 6 compares the lifetime of commodities between LOCL, LONL, MLMS, and LMM algorithms. Results show that for the minimum commodity lifetime, all four algorithms provide the same lifetime. For the other commodities, the LOCL algorithm provides a higher lifetime. The difference between LOCL and LONL algorithms is due to the fact that after the first node runs out of its energy, the LOCL algorithm tries to maximize the next commodity lifetime while the LONL algorithm tries to maximize the second node lifetime. The second node does not necessarily belong to the second commodity. This happens for the subsequent commodities and leads to different results for LOCL and LONL algorithms. The difference increases as the commodity number increases.

The lifetimes for the commodities obtained from the MLMS algorithm are almost equal because this algorithm only maximizes the minimum lifetime in the network. For the LMM algorithm, the first routing flow from the schedule is compared.

Fig. 7 compares the lifetime of nodes under LONL, MLMS, and LMM algorithms. The LONL algorithm maximizes the

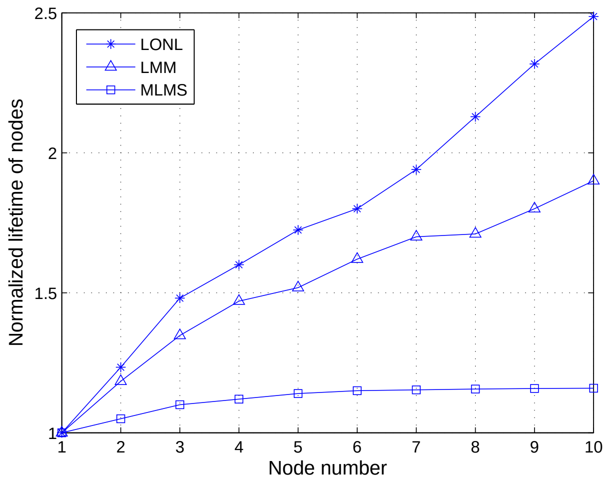


Fig. 7. Lifetime of nodes for LONL, LMM, and MLMS algorithms for a network with four sinks.

lifetime of all the nodes in the network and consequently has better performance. The LMM algorithm has a better performance compared to the MLMS algorithm. In the MLMS algorithm, only the minimum lifetime is maximized and therefore the lifetimes of other nodes are almost equal.

V. Conclusions

Multiple sinks can be employed in a WSN to increase the lifetime of the network. In this paper, we formulated two algorithms to fairly share the network resources among various commodities and nodes in the system using the concept of lexicographical fairness. We first proposed the centralized LOCL algorithm to obtain the exact lexicographically optimal solution. LOCL is a stepwise algorithm. In each step, a linear mixed-integer programming problem is being solved. Our second algorithm, called the LONL algorithm, is distributed. It can obtain the optimal solution under certain assumptions, and the sub-optimal solutions in general. The LONL algorithm is easier for implementation and more practical for deployments. Simulation results show that both LOCL and LONL algorithms have better performance compared to some existing schemes.

There are several directions for future work. The algorithm can be extended to enable the source nodes to select the appropriate sink. Alternatively, the problem can be formulated such that the optimal location for the sinks is determined while the network lifetime is being maximized.

Appendix

A. Linearization Technique

Consider a non-negative real variable x and a binary variable a. The product of these variables can be replaced by a new real variable z. The value of z corresponds to the value of x and a as follows:

$$z = \begin{cases} 0, & \text{if } a = 0, \\ x, & \text{if } a = 1. \end{cases} \quad (22)$$

Assume that x_{max} is an upper bound for the real variable x. The desired correspondence between x, a, and the new variable z is obtained by requiring that [34]:

$$\begin{aligned} &0 \le z \le x, \\ &x - x_{max}(1-a) \le z \le x_{max} a. \end{aligned} \quad (23)$$

B. Regularization Technique

Consider the following convex optimization problem with variable $\mathbf{x}$:

$$\begin{aligned} \text{minimize} \quad & f_0(\mathbf{x}) \\ \text{subject to} \quad & f_i(\mathbf{x}) \ge 0, \qquad i = 1, \ldots, m \\ & \mathbf{x} \succeq \mathbf{0}. \end{aligned} \quad (24)$$

where f_0, f_i's can be linear or nonlinear functions. The *optimal value* (i.e., minimal value) is denoted by $p^\star$ and is achieved at an *optimal solution* $\mathbf{x}^\star$. That is, $p^\star = f_0(\mathbf{x}^\star)$. Now consider the following optimization problem:

$$\begin{aligned} \text{minimize} \quad & f_0(\mathbf{x}) + \delta\phi(\mathbf{x}) \\ \text{subject to} \quad & f_i(\mathbf{x}) \ge 0, \qquad i = 1, \ldots, m \\ & \mathbf{x} \succeq \mathbf{0}. \end{aligned} \quad (25)$$

Mangasarian *et al.* [26] proved that if the problem is a linear programming problem, for all values of δ below some positive threshold, the optimal solutions of the regularized problem (25) are also the optimal solution in problems (24). Recently in [27], Friedlander *et al.* extended this work and proved that when the problem is nonlinear, the result still holds. They showed that this threshold is the inverse of the Lagrange multiplier of the second inequality constraint in the following problem:

$$\begin{aligned} \text{minimize} \quad & \phi(\mathbf{x}) \\ \text{subject to} \quad & f_i(\mathbf{x}) \ge 0, \qquad i = 1, \ldots, m \\ & f_0(\mathbf{x}) \le p^\star \\ & \mathbf{x} \succeq \mathbf{0}. \end{aligned} \quad (26)$$

They proved that this problem always has a Karush-Kuhn-Tucker (*KKT*) point. Thus, the threshold always exists.

C. Proof of Lemma 1

We use mathematical induction to prove this lemma. If there is at least one path between each source and its associated sink, the feasible set of problem MIP-1 is nonempty. Consequently, the optimal set is also nonempty. If the feasible region of problem MIP-n is nonempty then its optimal set is nonempty. Since the feasible set of problem MIP-$(n+1)$ is the optimal set in problem MIP-n, therefore it is nonempty. ∎

D. Proof of Lemma 2

We use mathematical induction to prove the following parts:

1) The inverse of the minimum lifetime is equal to the maximum normalized power consumption.
2) If the inverse of the ith minimum lifetime is equal to the ith maximum normalized power consumption, then

the inverse of the $(i+1)$th minimum lifetime is equal to the $(i+1)$th maximum normalized power consumption.

Both parts can be proved by contradiction. We only prove part one; the proof for the second part can be derived in similar manner. Let T_1 denote the minimum lifetime (i.e., the maximum achievable value for the minimum lifetime) and g_1 denote the maximum normalized power consumption (i.e., the minimum achievable value for the maximum normalized power consumption).

Now assume that $g_1 \neq \frac{1}{T_1}$. If $g_1 > \frac{1}{T_1}$, it means that it is possible to reduce maximum normalized power consumption to $\frac{1}{T_1}$ while we have assumed that g_1 is the minimum achievable value for the maximum normalized power consumption, which is a contradiction. If $g_1 < \frac{1}{T_1}$, it means that it is possible to reduce maximum normalized power consumption to $\frac{1}{T_1}$ while we have assumed that g_1 is the minimum achievable value for the maximum normalized power consumption which is a contradiction. If $g_1 > \frac{1}{T_1}$, it means that it is possible to increase minimum lifetime to $\frac{1}{g_1}$ while we have assumed that T_1 is the maximum achievable value for the minimum lifetime. It is also a contradiction. ■

E. Proof of Theorem 1

The set of constraints in problem (20) constructs a closed convex set. Let g_i^{max} be a loose upper bound for the normalized power consumption of node $i \in \mathcal{V}$. Let $\mathbf{g}$ denote the vector of normalized power consumption and $\mathbf{g}^\gamma$ represent the optimal value of problem (20) when $f(x) = (h(x))^\gamma$. The closed feasible set and the loose upper bound constraint construct a compact feasible set for $\mathbf{g}^\gamma$. Since $\mathbf{g}^\gamma$ is a sequence in a compact set, there exists a subsequence of γ, $\{\gamma_m, m \geq 1\}$, such that $\mathbf{g}^{\gamma_m}$ converges to some $\mathbf{g}^\star$ as $m \to \infty$, where $\mathbf{g}^\star$ is the limit point of $\mathbf{g}^{\gamma_m}$. We prove that $\mathbf{g}^\star$ is the lexicographically smallest normalized power consumption vector and is unique.

Proof by contradiction: Assume that $\mathbf{g}^\star$ is not the lexicographically optimal solution and $\mathbf{g}'$ is the lexicographically optimal vector. $\mathbf{g}'$ is lexicographically less than $\mathbf{g}^\star$. It means that, there exists i such that $g_j^\star = g_j'$ for all $j < i$ and $g_i^\star > g_i'$. Let $\xi = g_i^\star - g_i'$. From the convergence of $\mathbf{g}^{\gamma_m}$ to $\mathbf{g}^\star$, there exists m_0 such that for all $m \geq m_0$, $g_n^{\gamma_m}$ is in the ϵ-neighborhood of $g_n^\star$ for all $n \in \mathcal{V}$:

$$g_n^\star - \epsilon \leq g_n^{\gamma_m} \leq g_n^\star + \epsilon. \tag{27}$$

Consider the expression A_m defined by:

$$A_m = \sum_{n=1}^{|\mathcal{V}|} \left(f_m\left(g_n'\right) - f_m\left(g_n^{\gamma_m}\right) \right), \tag{28}$$

where $f_m(x) = (h(x))^{\gamma_m}$. From the optimality of $\mathbf{g}^{\gamma_m}$, we have $A_m \geq 0$. Since for the values of $n < i$, the members of two vectors are equal, we have:

$$A_m = f_m\left(g_i'\right) - f_m\left(g_i^{\gamma_m}\right) + \sum_{n=i+1}^{|\mathcal{V}|} \left(f_m\left(g_n'\right) - f_m\left(g_n^{\gamma_m}\right) \right). \tag{29}$$

Since the function $f_m(x) > 0$, we have:

$$\begin{aligned} A_m &\leq f_m\left(g_i'\right) - f_m\left(g_i^{\gamma_m}\right) + \sum_{n=i+1}^{|\mathcal{V}|} f_m\left(g_n'\right) \\ &= -f_m\left(g_i^{\gamma_m}\right) + \sum_{n=i}^{|\mathcal{V}|} f_m\left(g_n'\right). \end{aligned} \tag{30}$$

The vector $\mathbf{g}'$ is lexicographically ordered. Thus, for all $n > i,\ g_n' < g_i'$, we have:

$$A_m \leq -f_m\left(g_i^{\gamma_m}\right) + M f_m(g_i'), \tag{31}$$

where $M = |\mathcal{V}| - i + 1$. We know that $g_i' = g_i^\star - \xi$. We also use equation (27) to obtain:

$$\begin{aligned} A_m &\leq -f_m(g_i^\star - \epsilon) + M f_m(g_i^\star - \xi) \\ &= -f_m(g_i^\star - \epsilon) \left(1 - M \frac{f_m(g_i^\star - \xi)}{f_m(g_i^\star - \epsilon)} \right). \end{aligned} \tag{32}$$

Since ϵ can be any number, we choose it such that $\epsilon < \xi$. The last term is equal to:

$$\frac{f_m(g_i^\star - \xi)}{f_m(g_i^\star - \epsilon)} = \left(\frac{h(g_i^\star - \xi)}{h(g_i^\star - \epsilon)} \right)^{\gamma_m}, \tag{33}$$

where $h(x)$ is an increasing function. When $\gamma_m \to \infty$ (i.e., $m \to \infty$), this term tends to zero. Therefore, $A_m \leq -1$, which is a contradiction. ■

Note that our proof is related to that of Lemma 3 in [35]. In [35], Mo *et al.* considered utility maximization problems and proved a similar theorem for max-min fairness; however, here we consider the normalized power consumption minimization problem and the lexicographically optimal is the notion of fairness.

References

[1] Y. Xue, Y. Cui, and K. Nahrstedt, "Maximizing lifetime for data aggregation in wireless sensor networks," *ACM/Kluwer Mobile Networks and Applications (MONET)*, vol. 10, pp. 853–864, Dec. 2005.

[2] Y. Hou, Y. Shi, and H. D. Sherali, "On node lifetime problem for energy-constrained wireless sensor networks," *ACM/Kluwer Mobile Networks and Applications (MONET)*, vol. 10, pp. 875–878, Dec. 2005.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Elsevier Computer Networks Journal*, vol. 38, pp. 393–422, March 2002.

[4] A. Ephremides, "Energy concerns in wireless networks," *IEEE Wireless Commun.*, vol. 9, pp. 48–59, Aug. 2002.

[5] J. Chang and L. Tassiulas, "Routing for maximum system lifetime in wireless ad-hoc networks," in *Proc. of Allerton Conference Communication, Control, and Computing*, Urbana, Illinois, Sept. 1999.

[6] T. Brown, H. Gabow, and Q. Zhang, "Maximum flow-life curve for a wireless ad hoc network," in *Proc. of ACM MobiHoc*, Long Beach, CA, Oct. 2001.

[7] A. Giridhar and P. R. Kumar, "Maximizing the functional lifetime of sensor networks," in *Proc. of the Fourth International Conference on Information Processing in Sensor Networks (IPSN)*, Los Angeles, CA, April 2005.

[8] H. Zhang and J. C. Hou, "On deriving the upper bound of α-lifetime for large sensor networks," in *Proc. of ACM MobiHoc*, Roppongi Hills, Japan, May 2004.

[9] M. Bhardwaj, T. Garnett, and A. Chandrakasan, "Upper bounds on the lifetime of sensor networks," in *Proc. of IEEE ICC*, Helsinki, Finland, July 2001, pp. 785–790.

[10] J. Chang and L. Tassiulas, "Energy conserving routing in wireless ad-hoc networks," in *Proc. of IEEE INFOCOM*, Tel-Aviv, Israel, March 2000.

[11] ——, “Maximum lifetime routing in wireless sensor networks,” *IEEE/ACM Trans. on Networking*, vol. 12, pp. 609–619, Aug. 2004.

[12] R. Madan and S. Lall, “Distributed algorithms for maximum lifetime routing in wireless sensor networks,” *IEEE Trans. on Wireless Communications*, vol. 5, pp. 2185–2193, Aug. 2006.

[13] R. Madan, S. Cui, S. Lall, and A. Goldsmith, “Cross-layer design for lifetime maximization in interference-limited wireless sensor networks,” *IEEE Trans. on Wireless Communications*, vol. 5, pp. 3142–3152, Nov. 2006.

[14] H. Nama, M. Chiang, and N. Mandayam, “Optimal utility-lifetime trade-off in self-regulating wireless sensor networks a distributed approach,” in *Proc. of 40th Annual Conference on Information Sciences and Systems (CISS)*, Princeton, NJ, March 2006.

[15] J. Zhu, S. Chen, B. Bensaou, and K. Hung, “Tradeoff between lifetime and rate allocation in wireless sensor networks: A cross layer approach,” in *Proc. of IEEE INFOCOM*, Anchorage, Alaska, May 2007.

[16] J. C. Dagher, M. W. Marcellin, and M. A. Neifeld, “A theory for maximizing the lifetime of sensor networks,” *IEEE Trans. on Communications*, vol. 55, pp. 323–332, Feb. 2007.

[17] E. I. Oyman and C. Ersoy, “Multiple sink network design problem in large scale wireless sensor networks,” in *Proc. of IEEE ICC*, Paris, France, June 2004.

[18] P. Thulasiraman, S. Ramasubramanian, and M. Krunz, “Disjoint multi-path routing to two distinct drains in a multi-drain sensor network,” in *Proc. of IEEE INFOCOM*, Anchorage, Alaska, May 2007.

[19] V. Shah-Mansouri and V. W. S. Wong, “Bounds for lifetime maximization with multiple sinks in wireless sensor networks,” in *Proc. of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, Victoria, Canada, Aug. 2007.

[20] H. Kim, Y. Seok, N. Choi, Y. Choi, and T. Kwon, “Optimal multi-sink positioning and energy efficient routing in wireless sensor networks,” *Information Networking, Lecture Notes in Computer Science*, vol. 3391, Springer, pp. 264–274, Dec. 2005.

[21] F. Lin and Y. Wen, “Multi-sink data aggregation routing and scheduling with dynamic radii in WSNs,” *IEEE Communications Letters*, vol. 10, pp. 692– 694, Oct. 2006.

[22] K. Yuen, B. Li, and B. Liang, “Distributed data gathering in multi-sink sensor networks with correlated sources,” in *Proc. of IFIP Networking*, Coimbra, Portugal, May 2006.

[23] V. Shah-Mansouri, A. H. Mohsenian-Rad, and V. W. S. Wong, “Multi-commodity lifetime routing for wireless sensor networks with multiple sinks,” in *Proc. of IEEE ICC*, Beijing, China, May 2008.

[24] S. Sarkar and L. Tassiulas, “Fair bandwidth allocation of multicasting in networks with discrete feasible set,” *IEEE Trans. on Computers*, vol. 53, pp. 785–797, July 2004.

[25] V. Shah-Mansouri and V. W. S. Wong, “Distributed maximum lifetime routing in wireless sensor networks based on regularization,” in *Proc. of IEEE Globecom*, Washington, DC, Nov. 2007.

[26] O. Mangasarian and R. Meyer, “Nonlinear perturbation of linear programs,” *SIAM Journal on Control and Optimization*, vol. 17, pp. 745–752, 1977.

[27] M. Friedlander and P. Tseng, “Exact regularization of convex programs,” *SIAM Journal on Optimization*, vol. 18, pp. 1326–1350, Nov. 2007.

[28] “ILOG CPLEX.” http://www.ilog.com/products/cplex/, 2007.

[29] “MOSEK.” http://www.mosek.com, 2007.

[30] A. H. Land and A. G. Doig, “An automatic method for solving discrete programming problems,” *Econometrica*, vol. 28, pp. 497–520, 1960.

[31] H. Taha, *Operations Research: An introduction*, 7th ed. Prentice Hall, 2003.

[32] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

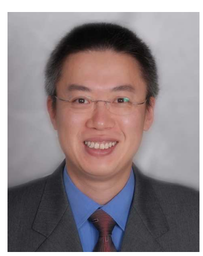
[33] L. Chen, S. H. Low, M. Chiang, and J. C. Doyle, “Optimal cross-layer congestion control, routing and scheduling design in ad hoc wireless networks,” in *Proc. of IEEE Infocom*, Barcelona, Spain, April 2006.

[34] F. Glover, “Improved linear integer programming formulations of non-linear integer problems,” *Management Science*, vol. 22, pp. 455–460, Dec. 1975.

[35] J. Mo and J. Walrand, “Fair end-to-end window-based congestion control,” *IEEE/ACM Trans. on Networking*, vol. 8, pp. 556–567, Oct. 2000.

Vahid Shah-Mansouri (S’02) is a Ph.D. candidate in the Department of Electrical and Computer Engineering at the University of British Columbia, Vancouver, BC, Canada. He received the B.Sc. and M.Sc. degrees in electrical engineering from University of Tehran, Tehran, Iran in 2003 and Sharif University of Technology, Tehran, Iran in 2005, respectively. From 2005 to 2006, he was with Farineh-Fanavar Co., Tehran, Iran. His current research interests are in power aware cross-layer design for wireless sensor networks.

Amir-Hamed Mohsenian-Rad (S’04) received his B.Sc. degree from Amir-Kabir University of Technology (Tehran, Iran) in 2002, M.Sc. degree from Sharif University of Technology (Tehran, Iran) in 2004, and Ph.D. degree from the University of British Columbia (Vancouver, Canada) in 2008, all in electrical engineering. From March to July 2007, he was also a visiting scholar at Princeton University (Princeton, NJ). As a graduate student, Dr. Mohsenian-Rad received the UBC Graduate Fellowship as well as the Pacific Century Graduate Scholarship from the British Columbia Provincial Government. He currently serves as TPC member for the IEEE International Conference on Communications (ICC’09) and the IEEE Consumer Communications and Networking Conference (CCNC’09). His research interests are in the area of optimization theory and its applications in computer communications and wireless networking.



Vincent W.S. Wong (SM’07) received the B.Sc. degree from the University of Manitoba, Winnipeg, MB, Canada, in 1994, the M.A.Sc. degree from the University of Waterloo, Waterloo, ON, Canada, in 1996, and the Ph.D. degree from the University of British Columbia (UBC), Vancouver, BC, Canada, in 2000. From 2000 to 2001, he worked as a systems engineer at PMC-Sierra Inc. He joined the Department of Electrical and Computer Engineering at UBC in 2002 and is currently an Associate Professor. His research interests are in resource and mobility management for wireless mesh networks, wireless sensor networks, and heterogeneous wireless networks. Dr. Wong is an associate editor of the IEEE Transactions on Vehicular Technology and an editor of KICS/IEEE Journal of Communications and Networks. He serves as TPC member in various conferences, including IEEE Globecom, ICC, and Infocom’09. He is a senior member of the IEEE and a member of the ACM.