On Maximizing Reliability of Lifetime Constrained Data Aggregation Tree in Wireless Sensor Networks

Mengfan Shan*, Guihai Chen*†, Fan Wu†, Xiaobing Wu*, Xiaofeng Gao†, Pan Wu*, Haipeng Dai*
*State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
{nju.mfshan,panwunju}@gmail.com, {wuxb,haipengdai}@nju.edu.cn
†Shanghai Key Laboratory of Scalable Computing and Systems, Shanghai Jiao Tong University, China
{gchen,fwu,gao-xf}@cs.sjtu.edu.cn

***Abstract*—Tree-based routing structures are widely used to gather data in wireless sensor networks. Along with tree structures, in-network aggregation is adopted to reduce transmissions, to save energy and to prolong the network lifetime. Most existing works that focus on the lifetime optimization for data aggregation do not take the link quality into consideration. In this paper, we study the problem of Maximizing Reliability of Lifetime Constrained data aggregation trees (MRLC) in WSNs. Considering the NP-completeness of the MRLC problem, we propose an algorithm, namely Iterative Relaxation Algorithm (IRA), to iteratively relax the optimization program and to find the aggregation tree subject to the lifetime bound with a sub-optimal cost. To adapt to the distributed nature of the WSNs in practice, we further propose a Prufer code based distributed updating protocol. Through extensive simulations, we demonstrate that IRA outperforms the best known related work in term of reliability.**

***Keywords*—data aggregation; iterative relaxing algorithm; maximizing reliability**

I. Introduction

Data collection is one of the most basic operations in many typical applications, such as habitat monitoring and structure maintenance, in Wireless Sensor Networks (WSNs). It requires sensor nodes to monitor environmental conditions continuously. Each sensor periodically reports the sensing data to the sink for processing. To collect sensing data, tree-based routing structures are widely adopted in WSNs.

Since sensors are usually battery-powered and have limited energy, how to perform data collection in an energy efficient way is a critical problem in WSNs. However, in a data collection tree, sensor nodes closer to the sink have to forward more packets and might deplete their power more quickly. This is so-called the energy hole phenomenon [2]. To alleviate this problem, some divisible functions (e.g., SUM, MAX, top-k, *etc.*) are used to fuse data packets and to eliminate redundant transmissions. Meanwhile, Wu *et al.* [1] proved the NP-Completeness of finding a maximum lifetime tree routing structure. They proposed an approximation algorithm that finds a sub-optimal tree in terms of network lifetime. Other related works (e.g., [3], [4], [5], [6]) present different schemes for finding maximum lifetime data aggregation trees with different additional constraints.

The major shortcoming of existing works is that they do not take the link quality into consideration. Considering the error-prone nature of wireless links, ignoring the link quality might cause high packet loss ratio and is not practical in WSNs. We define the reliability of a data aggregation tree as the probability that a round of data collection is successfully performed. It is conflicting to achieve both high reliability and optimal network lifetime in the general case. Therefore, we need to carefully balance the trade-off between these two contradicting objectives.

In this paper, we consider the problem of Maximizing Reliability of Lifetime Constrained data aggregation trees (MRLC) in WSNs with unreliable wireless links. We aim at maximizing the reliability of the chosen data aggregation tree while satisfying the lifetime bound required. We first analyze the energy consumption model by preliminary experiments. As the MRLC problem is NP-complete, we relax the formulation of the MRLC problem to a linear program. We present a centralized algorithm called Iterative Relaxation Algorithm (IRA), which iteratively relaxes the linear program to get an integral solution. For better applicability of our result to WSNs, we present a distributed updating protocol based on message exchanging.

Our main contributions are as follows.

- To the best of our knowledge, we are the first to study the problem of maximizing reliability of lifetime constrained data aggregation tree taking unreliable wireless links into consideration. The lifetime constraint guarantees the lifetime performance and maximizes reliability of the data aggregation tree.
- We solve the MRLC problem by proposing an algorithm, namely IRA, to find the aggregation tree subject to the lifetime bound with a sub-optimal cost.
- We further propose a Prüfer code based distributed updating protocol. The updating protocol is called in two cases: a tree-link gets worse or a non-tree link gets better.
- We carry out trace-driven evaluations so as to analyze the performance of our approaches. The results show that our approaches greatly improve the reliability of WSNs.

The rest of this paper is organized as follows. Section II reviews related works. Section III introduces our motivation and the network model. In Section IV, we give the problem definition and formulate the MRLC problem. We present the centralized algorithm IRA in Section V and the distributed protocol in Section VI. Section VII gives the performance evaluations. Finally, we conclude our work in Section VIII.

0190-3918/15 $31.00 © 2015 IEEE
DOI 10.1109/ICPP.2015.17

II. Related Works

There are many related works on the problem of building data collection and aggregation structures in wireless sensor networks. In this section, we briefly review related works.

Park and Sahni [8] proved that optimal routing in terms of the network lifetime is NP-hard in wireless sensor networks. They proposed a non-tree-based scheme for maximizing network lifetime in data collection networks. However, the solution is a heuristic approach and it does not consider the energy consumption of receiving. Wu *et al* [2] investigated the energy hole problem in data collection. They proposed a distributed routing algorithm based on shortest path trees, but the sensors have to be deployed in a predefined distribution.

Some related works [22], [13], [17] study the problem of reducing total energy consumption of data collection and aggregation networks. Crowcroft *et al.* [22] considered the problem of efficient data gathering in sensor networks and the efficiency is measured by total energy consumption, the quality of the transmissions, etc. The main objective is to construct efficient backbones for multi-hop data collection with data aggregation. Wu *et al.* [13] studied the problem of energy-efficient wake-up scheduling in data aggregation networks. They proposed efficient approximation algorithms for constructing data aggregation trees that guarantee both the energy consumption and network throughput are within a constant factor of optimal. Kuo and Tsai [17] studied the problem when a certain aggregation ratio q is given. The problem is NP-complete and they proved that every shortest path tree has an approximation ratio of 2. They also proved that the problem is NP-complete if some relay nodes are available and propose an 7-approximation algorithm. However, the total energy consumption is different from the network lifetime. Reducing the total energy consumption does not guarantee the improvement of the network lifetime.

Building a tree structure to maximize network lifetime is first proposed in data collection networks without aggregation. In data gathering without aggregation, nodes close to the sink would suffer from heavy loads. To extend the network lifetime, energy consumption of nodes should be balanced effectively. Some related works focus on maximizing lifetime through efficient balancing the energy consumption among sensor nodes. Levin *et al.* [20] studied the problem under two network models: homogeneous networks with obstacles and heterogeneous networks without obstacles. They provided an approximation algorithm for each network model. However, they did not consider the energy consumption for receiving, which should not be ignored in most sensor networks. Liang *et al.* [19] proved that the problem is NP-complete. They provided an algorithm called Maximum lifetime Tree construction for data gaThering without aggregation (MITT) and proved that MITT has an approximation ratio of $\Omega\left(\log n/\log\log n\right)$. Imon *et al.* [23] proposed a novel and efficient algorithm called Randomized Switching for Maximizing Lifetime (RaSMaLai). RaSMaLai aims at maximizing the network lifetime through load balancing with a low time-complexity. They also designed a distributed version of RaSMaLai.

There are some related works focusing on maximizing lifetime in data aggregation networks. Cheng *et al.* [15] studied the degree bounded data aggregation network problem, but only proposed heuristic approaches. Li [16] presented the first distributed method to construct a bounded degree planar connected structure LRNG (Local Relative Neighborhood Graph), whose total link length is within a constant factor of the minimum spanning tree using total $O(n)$ messages under the broadcast communication model. However, this scheme is only for nodes distributed in a two dimensional plane and the transmission range is uniform and normalized to one unit. An LRNG has more links than a minimum spanning tree and it consumes more energy than that by MST. Tan and Körpeoğlu [3] proposed two algorithms for maximizing lifetime. The algorithms are based on sensor locations and minimum spanning tree construction. A linear program based approach presented by Xue *et al.* [4] gives an approximation algorithm for finding a maximum lifetime aggregation tree. Wu *et al.* [1] first proved that the problem of maximizing lifetime of data aggregation networks is NP-complete. They also proposed an approximation algorithm. In all these works, they have a strong assumption that wireless links are reliable.

Only a few works take both lifetime and unreliable links into consideration. Kwon *et al.* [21] tried to maximize lifetime under reliability and stability constraints in wireless sensor networks. They found that short-distance hops are not always optimal for energy saving and using multi-hop paths may consume higher energy. They investigated the problem of maximizing the network lifetime under reliability constraint on the end-to-end success probability. In a tree-based data collection model, it is important to balance loads among sensor nodes and the reliability model is different from the end-to-end paths. Shen *et al.* [5] presented the problem of constructing data gathering tree to maximize the lifetime of unreliable wireless sensor network under delay constraint. Their solution is based on the minimum spanning tree of WSNs where the weights of links are the Expected Transmission Count and the network reliability is guaranteed by retransmissions. In this paper, we use reliability to define the goodness of wireless links and maximize the reliability of data aggregation trees. The retransmissions are not necessary for our solution and the lifetime preformance is still outstanding.

III. Preliminaries

In this section, we first introduce the motivation of our work. We then present the network model in this paper.

A. Motivation

Our work is motivated by the intersection between maximizing network lifetime and unreliable wireless links in wireless data aggregation sensor networks.

First, wireless links are highly unreliable in wireless sensor networks. Link-quality based forwarding strategies are designed to improve the performance of end-to-end throughput and transmission delay in tradition ad hoc networks and sensor networks and not suitable for data aggregation networks. For example, ETX [10] is a link-quality based forwarding strategy based on the expected transmission count. A node sends a packet again and again until the packet is received correctly by a forwarder. It is energy consuming and is not suitable for sensor networks where lifetime is a key issue. Retransmission is also not necessary in some time critical applications, such as

tracking and fire monitoring, etc. Figure 1 shows the average number of packets for one round of data aggregation with unreliable links for different sizes of networks. As shown in Figure 1, when the network has 16 nodes, the total number of packets increases from 15 to 150 while the average link quality decrease from 100% to 10%. The total number of packets increases even more when there are more nodes in the networks. It means that nodes spend 90% of energy in retransmission.

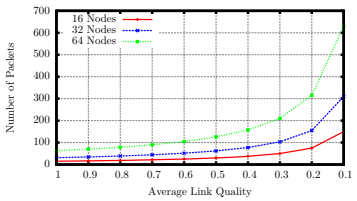


Fig. 1. Average Number of Packets vs. Average Link Quality

Another problem is that most previous works focusing on maximizing network lifetime perfer networks with good connectivity, because better connectivity leads to more forwarding candidates. Nodes might take long-distance links to get better connectivity. Figure 2 shows the average packet reception ratio of different distances. We also vary the transmitting power of the TelosB nodes. As shown in Figure 2, the link quality decreases while the distance increases when $Tx = 19$ and the average link quality goes from almost 100% to less than 10% while the distance increases from $4ft$ to $16ft$ when the transmission power is 11 and 15.

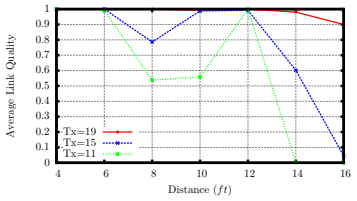


Fig. 2. Distance vs. Average Link Quality

B. Network Model

Suppose that there is a sink node and $n-1$ sensor nodes $V = \{v_0, v_1, \cdots, v_{n-1}\}$, where v_0 is the sink node and $\{0, 1, 2, \cdots, n-1\}$ are the labels of nodes. The initial energy of v_i is denoted by $I(v_i)$. We use an undirected graph $G = (V, E)$ to represent the network, where E is the set of wireless links. G is connected so that every node has at least one path to the sink. Nodes can do in-network aggregation, i.e., each node aggregates the received messages from its children, combines them with its own sensing information into a single outgoing message, and sends the combined message to the next hop node.

To estimate the lifetime, we use PowerMonitor [9] to measure the energy consumption of sensor nodes under 3 working states: sending, receiving and idle. We use three identical TelosB sensor nodes and show their energy consumption in Figure 3. The first sensor keeps sending packets. Each packet is 34 bytes' long and the energy consumption is shown in Figure 3(a), which has an average of $80mW$. The second sensor receives all packets the first sensor sent. Its energy consumption is about $60mW$ as shown in Figure 3(b). The last sensor does all the Led blinking and computing as the first two with its radio module turned off. It costs only $80\mu W$, as shown in Figure 3(c).

According to the observations above, most energy is used in the sending and receiving states. In this paper, we estimate network lifetime by only considering the energy consumption for sending and receiving packets. We denote Tx as the energy cost of sending a packet and Rx as the energy consumption of receiving a packet. For an arbitrary data aggregation tree T, the lifetime of any node v is related to the number of packet sending and receiving in each data aggregation round. The lifetime of a node v is defined as:

$$L(v) = \frac{I(v)}{Tx + Rx * Ch_T(v)}, \tag{1}$$

where $Ch_T(v)$ is the number of children in the aggregation tree T. The network lifetime L is defined as the time till the first sensor node depletes its energy, which is presented as:

$$L = \min_{\forall v \in V} L(v).$$

In this paper, we use the Packet Reception Ratio as the reliability of wireless links, which is denoted by q_e. The Packet Reception Ratio (PRR, which is also referred as the Packet Success Ratio) is a basic metric for link quality estimation and has been widely used in many sensor networks. It can be estimated by the ratio of the number of correctly received packets to the number of total transmitted packets:

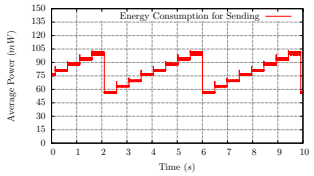
$$q_e = \frac{N_r}{N_s}, \tag{2}$$

where N_r is the number of successfully received packets and N_s is the number of total transmitted packets.

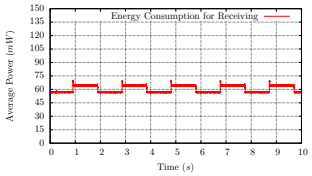
In some applications of WSNs, such as tracking and fire monitoring, retransmission and acknowledgement are normally disabled for real-time data gathering. As there is no retransmission or ACK, in each data aggregation round, the successful ratio of all packets received is the product of all edges' packet reception ratios. The reliability of a data aggregation tree T is defined as the product of all edges' packet reception ratios. Let $Q(T)$ be the reliability of tree $T(V, E_T)$ and it is defined as follows:

$$Q(T) = \prod_{e \in E_T} q_e.$$

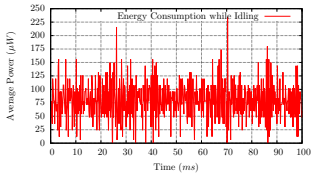
A toy example of the network reliability is shown in Figure 4. There are 6 sensor nodes in the network and Figure 4(a) is a data aggregation tree with the PPR of each links. In an data aggregation round, Node 4 receives packets from Node 2 and Node 3, aggregates with its own sensing information, and then sends a packet to the sink node. For time critical applications where retransmissions are not necessary, if Node 4 did not receive the packet from Node 2, due to the unreliable link, the sink will not get the sensing information of Node 2. As we can see, the sink can only get sensing information from all nodes if every packet was correctly received. The network reliability of the aggregation tree in Figure 4(a) is $0.36 = 0.8 \times 0.5 \times 0.9 \times 1.0 \times 1.0$, which is the product of all links?PPRs. Another possible aggregation



(a) The power consumption during a TelosB node sending packets



(b) The power consumption during a TelosB node receiving packets



(c) The power consumption during a TelosB node idling

Fig. 3. The figures show the energy consumption of TelosB nodes during sending, receiving and idling, respectively. We get the data by PowerMonitor [9]. The average power consumption during a TelosB node sending packets is about $80mW$. When a node is listening and receiving, the energy consumption is at the average of $60mW$. When a TelosB node is in the idle state, the power consumption is only about $80\mu W$.

tree is shown in Figure 4(b). The aggregation reliability is $0.648 = 0.8 \times 0.9 \times 0.9 \times 1.0 \times 1.0$. We can see that the aggregation tree in Figure 4(b) has a better performance in term of reliability.

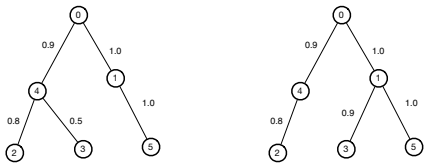


Fig. 4. A toy example of the network reliability.

IV. Problem Formulation

In this paper, we fuse a linear program to formulate the MRLC problem. The linear program is based on Subtour Linear Program, which will be introduced first in this section. We then give the formal definition of the Maximizing Reliability of Lifetime Constrained (MRLC) data aggregation tree problem. At last we formulate the MRLC problem by a linear program.

A. Subtour LP

We first formulate MST with Subtour LP. Subtour LP is a linear program which is usually used in the study of MST and TSP. Later we will show the MRLC problem can be formulated with a similar linear program.

$$\begin{aligned} \min \quad & \sum_{e \in E} c_e x_e && \text{(Subtour LP)} \\ \text{s.t.} \quad & x(E(S)) \leq |S| - 1, \forall S \subseteq V && (3) \\ & x(E(V)) = |V| - 1, && (4) \\ & x_e \geq 0, \qquad \forall e \in E. && (5) \end{aligned}$$

x_e is the variable that indicates whether e is in the tree or not. S is any subset of V and $E(S)$ denotes the set of edges with both endpoints in S and $x(F)$ is used to denote $\sum_{e \in F} x_e$ where F is a set of links. Constraint 3 guarantees that there is no loop among any subset of V, and Constraint 4 ensures that a spanning tree has $|V| - 1$ edges. Though the number of constraints is exponential, we can prove that an integral solution of the Subtour LP can be found in polynomial time. We first introduce the theorem of Grötschel, Lovász and Schrijver in [11].

Theorem 1. *Given a full-dimensional polytope P and a polynomial-time separation oracle for P, one can find an optimal extreme point solution to a linear objective function over P (assuming it is bounded) via the Ellipsoid algorithm that uses a polynomial number of operations and calls to the separation oracle.*

According to Theorem 1, we only need to find a way to check if x satisfies the Subtour LP. Related works [12] have proven that there is a Min-Cut based polynomial time separation oracle which finds an S with minimum $|S| - 1 + x(E(V)) - x(E(S))$. If it is less than $|V| - 1$, x is not in the polytope, and vice verse.

Then we have the following lemma.

Lemma 1. *An extreme point solution to the Subtour LP is integral.*

To prove Lemma 1, we first prove that

$$x_e \leq 1, \quad \forall e \in E. \tag{6}$$

Assume that u and v are two endpoints of e. Set S to be $\{u, v\}$. According to Equation 3, we can get that $x(E(S)) \leq |S| - 1$ and $x_e \leq 1$.

Next we introduce the *laminar* family and intersecting. Let F be the family of tight constraints in Equation 3 and 4:

$$F = \{S \subseteq V \mid x(E(S)) = |S| - 1\}. \tag{7}$$

Call two sets $X, Y \in F$ intersecting if $X \cap Y$, $X \setminus Y$ and $Y \setminus X$ are nonempty. A family of sets is *laminar* if no two sets are intersecting. Let $\mathcal{L} \subseteq F$ be the maximal laminar family. As there is no intersecting, sets in $\mathcal{L}$ are linear independent. According to the Rank Lemma [12], the number of linear independent tight constraints equals to non-zero variables. In this case, we have that:

$$|E^*| = |\mathcal{L}|, \tag{8}$$

where $E^* = \{e \mid x_e > 0\}$.

Lemma 2. *A laminar family $\mathcal{L}$ over the ground set V without singletons (subsets with only one element make no contribution in Equation 3) has at most $|V| - 1$ distinct member.*

Proof: We prove this lemma by induction on the size of the $|V|$. If $|V| = 2$, it is quite clear that $|\mathcal{L}| \leq 1$. Let the claim be true for all laminar families over ground sets of size strictly smaller than n. Let $|V| = n$ and S be a maximal set

in $\mathcal{L}$ which is not equal to V. According to the definition of *laminar* family, each set in $\mathcal{L}$, except V, is either contained by S, or is not intersected with S. The number of sets contained by S (including S) is at most $|S|-1$, by induction hypothesis. The sets in $\mathcal{L}$, which are not intersected with S form a *laminar* family on the ground set $V \setminus S$, which has a size at most $|V| - |S| - 1$ by induction hypothesis. Along with V, the *laminar* family $\mathcal{L}$ has at most $|S|-1+|V|-|S|-1+1 = |V|-1$ sets. ■

According to Equation 6 and Equation 4, we have that $|E^*| \geq |V| - 1$. We can also prove that $|E^*| \leq |V| - 1$ by Equation 8 and Lemma 2. Then we have $|E^*| = |V| - 1$ and Lemma 1 is proved.

B. Problem Definition

Based on the aforementioned model, we give the definition of the **M**aximizing **R**eliability of **L**ifetime **C**onstrained data aggregation tree (MRLC) problem.

Problem 1 (MRLC Problem)**.** *Given a network topology $G = (V, E)$, the packet reception ratio of all possible links: $E \to \mathbb{R}$, initial energy of all sensor nodes: $V \to \mathbb{R}$ and a threshold of the lifetime LC, for any data aggregation tree T and its lifetime $L(T)$, we want to maximize $Q(T)$ while $L(T) \geq LC$.*

However, it is hard to directly express $Q(T)$ in a graph. We first introduce a notion, edge cost, in order to maximize the reliability of sensor networks. For any wireless link e, the cost is defined as the logarithm of ETX. ETX is a high throughput metric for wireless networks proposed by Couto *et al.* in [10]. Let $ETX(e)$ be the ETX of link e, which is the expected total number of packet transmissions (without ACK).

$$c_e = \log ETX(e) = -\log q_e. \tag{9}$$

The cost of a data aggregation tree is the sum of all edges' cost, which is:

$$C(T) = \sum_{e \in T} c_e. \tag{10}$$

Lemma 3. *The aggregation tree with maximum reliability has the minimum cost defined in Equation 10.*

Proof: By Equation 9 and 10, we have that:

$$\begin{aligned} C(T) &= \sum_{e \in T} \log ETX(e), \\ &= -\log \prod_{e \in T} q_e, \\ &= -\log Q(T). \end{aligned}$$

According to the above equation, we can prove the lemma. ■

Problem 2. *Given a network topology, the cost of each link, the initial energy of all sensor nodes, and a lifetime threshold of the lifetime LC, for any data aggregation tree T, we want to minimize $C(T)$ while $L(T) \geq LC$.*

According to Lemma 3, Problem 2 is essentially equivalent to Problem 1.

Algorithm 1 Iterative Relaxation Algorithm

Input:
The network topology $G(V, E)$;
The packet reception ratio of each edge $q_e, e \in E$;
Initial energy of every node $I(v), v \in V$;
The threshold of lifetime B.
Output:
The spanning tree T.

```
W ← V.
I_min = min_{v_i ∈ V} I(v_i), ∀v_i ∈ W.
L' ← I_min LC / (I_min − 2RxLC).
while W ≠ ∅ do
    Find an extreme point solution x of LP(G, L', W).
    Remove every edge e with x_e = 0 from G.
    Let E* be the support of x.
    If these is a vertex v_i ∈ W with E*(L(v_i)) ≥ LC,
    update W ← W \ {v_i}.
end while
return T = (V, E*).
```

C. Problem Formulation

With lifetime constraint added to the Subtour LP, the MRLC problem is formulated as follows:

$$\min \quad \sum_{e \in E} c_e x_e \tag{11}$$

$$\text{s.t.} \quad x_e \geq 0, \qquad \forall e \in E \tag{12}$$

$$x(E(S)) \leq |S| - 1, \forall S \subseteq V \tag{13}$$

$$x(E(V)) = |V| - 1, \tag{14}$$

$$x(L(v_i)) \geq L', \qquad \forall v_i \in W. \tag{15}$$

The parameter G is the network topology, L' is a given constraint for lifetime and W is the set of nodes whose lifetime is constrained by L'. The variable x_e is used to indicate that whether e is included in the data aggregation tree and c_e is the cost of edge e. Equation 12, 13 and 14 guarantee that the output is a spanning tree. Equation 15 promises that every node's lifetime is no less than L'. The objective function is to minimize the total cost, which equals to maximize reliability, according to Lemma 3.

As we can see, an integral solution of $LP(G, L', W)$ is the optimal solution to our problem if $L' = LC$ and $W = V$. However, integer programming is NP-hard. In the following sections, we propose a centralized algorithm by iteratively relaxing the linear program and get a sub-optimal solution.

V. Centralized Algorithm

In this section, we present our Iterative Relaxation Algorithm (IRA), followed by the correctness and performance analysis.

A. Iterative Algorithm

The proposed algorithm is shown in Algorithm 1. Let I_{min} be the minimum initial energy among all nodes. L' is the lifetime constraint used in the $LP(G, L', W)$, which is a little larger than LC. If there is a data aggregation tree with lifetime bounded by L', IRA returns a data aggregation tree

whose lifetime is bounded by LC and the total cost is at most $OPT(L')$, where $OPT(L')$ is the minimum spanning tree with lifetime constrained by L'. In [14], Fürer and Raghavachari proved that it is possible to find a spanning tree with maximum degree at most $k+1$ if there is a spanning tree with maximum degree at most k. In this case, we can say that either IRA:

- shows that there is no data aggregation tree with lifetime bounded by LC,
- or finds a data aggregation tree with lifetime bounded by LC and total cost at most $OPT(L')$.

B. Correctness and Performance Analysis

In this subsection, we prove the correctness of IRA, which means the algorithm can always find a node in W whose lifetime constraint can be removed in Line 8 of Algorithm 1. Once all lifetime constraints are removed, the problem is transformed to a minimum spanning tree problem. With Lemma 1, we can prove that an extreme point solution to $LP(G, L', W)$ is integral.

Assume that x is an extreme point solution to $LP(G, L', W)$ and E^* is the support of x, which is:

$$E^* = \{e \mid x_e > 0\}. \tag{16}$$

Let F be the family of tight constraints in Equation 13 and 14:

$$F = \{S \subseteq V \mid x(E(S)) = |S| - 1\}. \tag{17}$$

According to the Rank Lemma [12], we have the following characterization:

Lemma 4. *Let x be an extreme point solution to $LP(G, L', W)$ and E^* is the support of x. There exists a set $T \in W$ and a laminar family $\mathcal{L}$ such that*

- *$x(L(v)) = L'$ for each $v \in T$ and $x(E(S)) = |S| - 1$ for each $S \in \mathcal{L}$.*
- *$\{E(S) \mid S \in \mathcal{L}\} \cup \{\delta(v) \mid v \in T\}$ are linear independent, where $\delta(v)$ is the set of edges connected to v.*
- *$|\mathcal{L}| + |T| = |E|$.*

Theorem 2. *There exists a node $v \in W$ with $x(L(v)) \geq L'$ if $W \neq \emptyset$.*

Proof: First of all, we observe that if $T = \emptyset$, $LP(G, L', W)$ can be seen as the Subtour LP. Hence, the problem is a simple minimum spanning tree problem and an extreme point of the linear program must be integral.

On the other hand, if $T \neq \emptyset$, suppose for the sake of contradiction that

$$E^*(L(v)) > LC, \quad \forall v \in W. \tag{18}$$

We give one token for each edge in E^* and the sum of all tokens equals to $|\mathcal{L}| + |W|$ by Lemma 4. We redistribute the tokens by the following two rules. Each edge $e \in E^*$ gives:

- x_e fractional token to the smallest set in $\mathcal{L}$ containing both endpoints of e,
- and $(1 - x_e)/2$ fractional token to each of its endpoints for the lifetime constraints.

According to the definition of the laminar family, every tight set constraint in $\mathcal{L}$ suffices to obtain one token in the first rule of redistribution. Thus the sum of tokens assigned to sets in the laminar is $|\mathcal{L}|$. Let $v \in T$ be a node with a lifetime constraint. v receives $(1 - x_e)/2$ token for each edge incident at v, for a total token of value:

$$\sum_{e \in \delta(v)} \frac{1 - x_e}{2} > \frac{E^*(\delta(v)) - x(\delta(v))}{2}. \tag{19}$$

As $E^*(L(v)) > LC$ and $x(L(v)) \geq L'$, we have that:

$$\frac{E^*(\delta(v)) - x(\delta(v))}{2} = \frac{I(v)}{I_{min}} > 1, \tag{20}$$

where Equation 19 holds since $x(L(v)) = L'$ and Equation 20 holds by $E^*(L(v)) > LC$. Every node in T gets more than 1 tokens and the sum of the second redistribution is more than $|T|$.

As we can see above, tokens are more than $|\mathcal{L}| + |T|$, which gives us the contradiction. ■

Next we analyse the performance of our scheme. First of all, for any iterative step, let C_1 be the optimal solution to $LP(G, L', W)$, let C_2 be the optimal solution after removing edges with $x_e = 0$. As we only remove edges with $x_e = 0$, the optimal solution should be the same. So we have $C_1 = C_2$. Let C_3 be the optimal solution after removing some lifetime constraints. The solution should be the same or better because there are less constraints in the linear program. We have that

$$C_3 \leq C_2 = C_1. \tag{21}$$

According to the above inequation, the total cost of the aggregation tree would never get worse in each iteration step. Let $C(IRA)$ be the final solution of IRA and we have that:

$$C(IRA) \leq OPT(L'). \tag{22}$$

VI. Distributed Protocol

In a distributed system, the link quality might change as time goes and the environment changes. It is inefficient to recall the centralized algorithm every time. In this section, we propose the distributed updating schemes for maintaining a sub-optimal aggregation tree. The updating schemes are activated under two circumstances. One is that a high quality wireless link in the data aggregation tree suddenly gets unreliable. The other is an unreliable link which is not in the data aggregation tree gets better. For each case, we propose a distributed updating scheme by message passing.

A. Prüfer Code

Before introducing the distributed protocol, we introduce the Prüfer code first, as it is very efficient for any node changing its parent when the spanning tree is presented by the Prüfer code. The Prüfer code, which is also called Prüfer sequence or Prüfer number, is a unique code associated with a labeled spanning tree. The Prüfer code needs only needs $n-2$ integers to present a labeled spanning tree with n nodes and both the encoding and decoding algorithms have low time complexity. Traditional Prüfer code is used to unrooted tree. We extend the Prüfer code to adapt the data aggregation tree, in which the root is the sink node and every non-sink node is aware of its parent.

Algorithm 2 Encoding

Input:
The labeled tree T where the label of a node is its ID.
Output:
Prüfer code P for T
1: $P = ()$.
2: $T' = T$.
3: **for** $i = 1$ to $n - 2$ **do**
4: Let u be the leaf with the largest label in T'.
5: There must exist a node v that $(u, v) \in T'$.
6: Remove u and (u, v) from T'
7: Append v to P ($p_i = v$).
8: **end for**
9: **return** $P = (p_1, p_2, \cdots, p_{n-2})$

Algorithm 3 Decoding

Input:
Prüfer code $P = (p_1, p_2, ..., p_{n-2})$
Output:
Decoded code $D = (d_1, d_2, \cdots, d_n)$.
1: $D = ()$.
2: $\bar{P} = P$.
3: **for** $i = 1$ to $n - 2$ **do**
4: Let u be the node with the largest label which is not in $D \bigcup \bar{P}$.
5: Append u to D ($d_i = u$).
6: Remove p_i from $\bar{P}$.
7: **end for**
8: Append p_{n-2} to D ($d_{n-1} = p_{n-2}$).
9: Append 0 to D ($d_n = 0$).
10: **return** $D = (d_1, d_2, \cdots, d_n)$.

1) Encoding: The encoding algorithm is shown in Algorithm 2. In each round, we find the leaf with the largest ID, add its parent into the Prüfer sequence and remove the leaf node from the tree. The algorithm goes until only 2 nodes remain. Figure 5(a) is an example of encoding. The tree has 5 leaf nodes, labeled as $\{1, 3, 5, 6, 7\}$ and 7 is the largest one. We remove 7 from the tree, and append 0 (7's parent) to P. The second removed node is 6 and 2 is appended to P. After removing 6, 2 is a leaf node and added to the leaf node set. We repeat these steps for $n - 2$ times until there are only 2 nodes and 1 edge left in the tree. We get the Prüfer code $P = (0, 2, 8, 4, 4, 0, 8)$ by removing $(6, 3, 2, 4, 7, 5, 1)$, respectively. With proper data structure, Algorithm 2 has a time complexity of $O(n \lg n)$.

2) Decoding: For any node in the data aggregation tree, by the time it received the Prüfer code, it can decode P by Algorithm 3. In the decoding algorithm, we find the children of nodes in the Prüfer code one by one. For all candidates, we pick the one with the largest ID. We use the encoded Prüfer code $P = (0, 2, 8, 4, 4, 0, 8)$ above as an example. First we set $\bar{P} = P$ and $D = ()$. Let u be the node with the largest label and not in $D \bigcup \bar{P}$, which is 7. Append 7 to D and remove 0 from $\bar{P}$. Repeat these steps until there is no node in $\bar{P}$ and D is now $D = (7, 6, 5, 3, 2, 4, 1)$. We than append 8, which is the last remove node from $\bar{P}$ and 0, which is the sink node with the smallest label, to D and we get $D = (7, 6, 5, 3, 2, 4, 1, 8, 0)$. As we can see in Figure 5(a), the edge set of the spanning tree is $\{(d_1, p_1), (d_2, p_2), \cdots, (d_{n-2}, p_{n-2})\} \bigcup \{(d_{n-1}, d_n)\}$ precisely. Algorithm 3 has also a time complexity of $O(n \lg n)$.

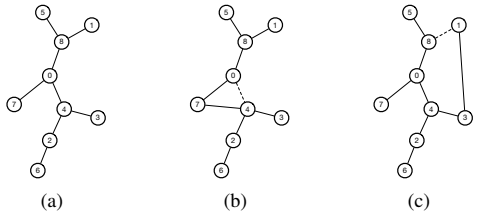


Fig. 5. An example of Prüfer code. The Prüfer code of Figure 5(a) is $P = (0, 2, 8, 4, 4, 0, 8)$ and the decode sequence is $D = (7, 6, 5, 3, 2, 4, 1, 8, 0)$. The Prüfer code P has a length of $n - 2$ and the decode sequence D has a length of n, where n is the number of nodes. Both the coding and decoding algorithms have the time complexity of $O(n \lg n)$. Figure 5(b) is an example of parent changing by the Prüfer code. 4 wants to change its parent from 0 to 7. After decoding the Prüfer code, 4 first finds its connected component without $(4, 0)$ and it is $(6, 3, 2, 4)$.

According to Equation 1, the lifetime of a node is only related to the number of its children in the tree. In Prüfer code $P = (0, 2, 8, 4, 4, 0, 8)$, 0, 4 and 8 all appear twice and 2 only appears once. In Figure 5(a), both 4 and 8 has two children in the tree, 2 has one children. The number of children is the same as the number appeared in the Prüfer code. The only exception is sink node 0, which has 3 children but 2. Besides the sink node, we have the following observation:

$$Ch_T(v) = N_P(v), \forall v \neq v_0, \tag{23}$$

where $N_P(v)$ is the number of v appearing in Prüfer code P. To the sink node, as we can see in Algorithm 2, one edge remains and this edge is always adjacent to the sink node because the sink node has the smallest label. This makes the sink has one more children than it appearing in the Prüfer code.

B. Distributed Protocol

There are two cases an update might happen. One is that a selected link in the tree gets unreliable and the child wants to change its parent. The other is that an unreliable link not in the tree gets better and can be added into the aggregation tree. Once an aggregation tree is constructed, the sink calculates the Prüfer code and broadcasts to all sensors.

1) Link Getting Worse: When a link in the tree gets worse, the child will select a new parent. It decodes the Prüfer code first, removes the link from the tree, find its new parent which connects two separated components with the highest link quality. Take Figure 5(b) as an example. 4 finds that the link $(4, 0)$ gets very unstable and has a high packet loss rate. Assume that 4 has the Prüfer code $P = (0, 2, 8, 4, 4, 0, 8)$ and it gets the sequence $D = (7, 6, 5, 3, 2, 4, 1, 8, 0)$ by decoding. If $(4, 0)$ was removed, the graph would be two connect components. By P and D, we can find that 2 and 3 are children of 4 and 6 is the child of 2. So 4 is in the component of $(6, 3, 2, 4)$ and the new parent of 4 should be in $V \setminus \{6, 3, 2, 4\}$. Assume that $(4, 7)$ has better link quality and 7's lifetime in under constraint after 4 connected. Let P' and D' be the updated sequences. We first move $(6, 3, 2, 4)$ to the top of D', which gives $D' = (6, 3, 2, 4, 7, 5, 1, 8, 0)$. We then move parents of $(6, 3, 2)$ to the top of P', followed 7 and other nodes. The updated P' is $P' = (2, 4, 4, 7, 0, 8, 8)$. As every node has the

Algorithm 4 Iterative Local Updating Algorithm

Input:
A Link (u, v);
Prüfer code $P = (p_1, p_2, ..., p_{n-2})$.
Output:
An Updated Aggregation Tree.

1: Decode P and get D.
2: Let p_u and p_v are parents of u and v, respectively.
3: Without loss of generality, we assume that the cost of (v, p_v) is smaller than (u, p_u).
4: **if** u can have more children under the lifetime constraint & the cost (v, p_v) is larger than (v, u) **then**
5: v changes its parent from p_v to u.
6: Recall this algorithm by giving (v, p_v) and updated Prüfer code.
7: **else if** v can have more children under the lifetime constraint & the cost (u, p_u) is larger than (v, u) **then**
8: u changes its parent from p_u to v.
9: Recall this algorithm by giving (u, p_u) and updated Prüfer code.
10: **end if**
11: **return** The updated tree.

same information, 4 only needs to broadcast a Parent-Changing information to other nodes and every node could get the same P' and D'. For every sensor node, it only needs to maintain P' and D'. The time complexity of parent changing is only $O(n)$ for each sensor.

2) Link Getting Better: Adding an edge in a tree is much more involved than the first case. There exists a loop in the graph after the edge is added. Usually a depth-first-search algorithm is used to find the loop in a graph. The time complexity of finding the loop in a graph is $O(n)$ and it can hardly be modified to a distributed system. Given the Prüfer code, it might be easier to find the loop. However, cutting a loop might change the child-parent relationship. For this problem, we present the ILU (Iterative Local Updating) algorithm. The algorithm is shown in Algorithm 4. Assume that the link between u and v is getting better and it is more reliable than the link between u and its parent. We first remove the link between u and its parent. We then let v be u's new parent by adding the link (u, v). At last, we recall the algorithm by inputting the link between u and its last parent.

An example of this algorithm is shown in Figure 5(c) while a link between 1 and 3 is getting better. Assume that $(3, 4)$ has lower cost than $(1, 8)$ and 3 can have more children under the lifetime constraint. The ILU algorithm solves the problem in two steps. First 1 changes its parent from 8 to 3 by using the Link Getting Worse scheme. Then we recall the ILU algorithm by regarding $(8, 1)$ as a link getting better. The ILU algorithm only needs information among two neighbors and it can search the loop in a distributed way iteratively.

VII. Performance Evaluation

We evaluate the performance of our approaches with trace data from a device free localization system. The system comprises a wireless sensor network consisting 16 adjustable tripods with a height of $0.9m$, along the perimeter of a $3.6m \times 3.6m$ square. The distance between two adjacent

Fig. 6. The Device Free Localization System

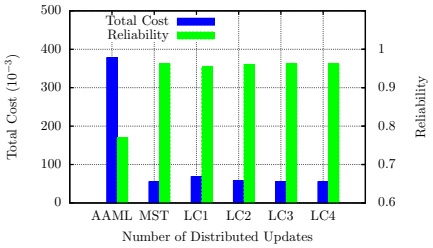


Fig. 7. Performance in the DFL system

sensors is $0.9m$. The sensors are labeled from 0 to 15 and the node 0 is the sink node. All sensors are powered by two AA batteries with initial energy of $3000J$. At the beginning, every sensor node broadcasts a thousand rounds of beacons to estimate the link quality. The energy consumption for sending and receiving a packet are $1.6^{-4}J$ and $1.2^{-4}J$, respectively. The network lifetime is defined as the total number of data aggregation rounds until the first node depletes all its energy. The system is shown in Figure 6.

We compare our algorithm IRA with two related works. One approach is Approximation Algorithm for Maximizing Lifetime (AAML) proposed in [1]. AAML starts from an arbitrary tree and iteratively reduce the load on bottleneck nodes. The bottleneck nodes are likely to deplete their energy due to high number of children or low remaining energy. Wu *et al.* prove that AAML terminates in polynomial time and is near optimal. The other solution (MST) is the Prim's Algorithm for the Minimum Spanning Tree problem [18]. It initializes a tree with the root node. Then it grows the tree by one edge: of the edges that connect the tree to vertices not yet in the tree, find the min-cost edge and transfer it to the tree. Repeat this step until all nodes are added into the tree. Since the MRLC problem is NP-complete and there is no efficient algorithm returning the optimal solution. The optimal solution of MRLC should be at least the cost of MST. We use MST as the lower bound of optimal solutions to our problem. In some cases, we only compare the cost of different solutions as cost is the metric used in both IRA and MST. According to Lemma 3, low cost equals to high reliability.

A. Performance in the DFL System

We run our algorithm with trace data from the device free localization system, as well as AAML and MST. We compare the total cost and the reliability of data aggregation trees given by different approaches. As AAML does not take link quality under consideration, we ignore unreliable links with the packet reception ratio lower than 0.95. The result is shown in Figure 7. The blue bars are the total cost and the

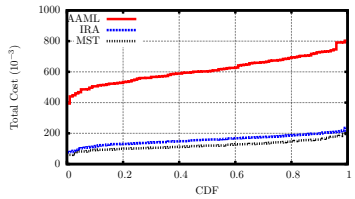


Fig. 8. Same Initial Energy

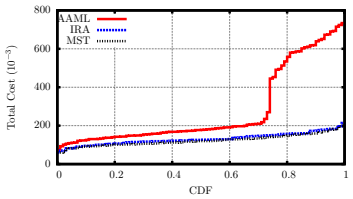


Fig. 9. Different Initial Energy

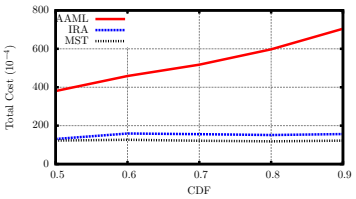


Fig. 10. Different Link Probabilities

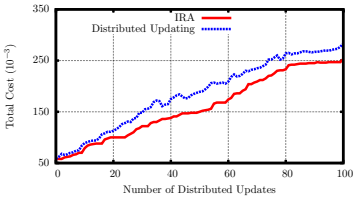


Fig. 11. Cost of Distributed Protocol

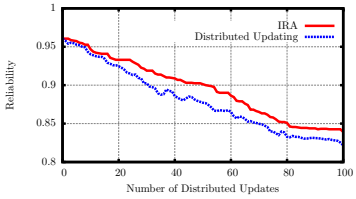


Fig. 12. Reliability of Distributed Protocol

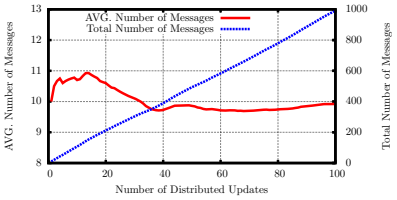


Fig. 13. Number of Messages

green bars are the reliability of different data aggregation trees. AAML has a total cost of 378 and the reliability is about 0.77. The aggregation tree generated by MST has the minimum cost among all spanning trees in the network, which is 55 and the reliability is about 0.963. Moreover, we use different lifetime constraints in our approaches, which are L_{AAML}, $1.5L_{AAML}$, $2L_{AAML}$ and $2.5L_{AAML}$, and L_{AAML} is the near optimal lifetime given by AAML. When $LC1 = L_{AAML}$, the total cost of IRA is 68 and the reliability is 0.954. Without loss of lifetime performance, the total cost is only 18% of which in AAML and the reliability has an improvement of 24%. The performances of IRA under some other lifetime constraints are also provided. When $LC2 = 1.5L_{AAML}$, the total cost is 58. When $LC3 = 2L_{AAML}$, the cost of IRA equals to that of MST, which is the minimum cost of all possible aggregation trees in the network. We can conclude that our solution has much better cost and reliability performance than AAML even under the same lifetime constraint and achieve the optimal reliability by a little violation of lifetime.

B. Performance in Random Graphs

We run our approaches on random graphs with different network parameters. Each random graph has 16 nodes and every possible edge occurs independently with probability 70%. The link quality of each edge is randomly selected in $(0.95, 1)$. We use the lifetime of aggregation trees by AAML as the lifetime constraint in our scheme. Here results of MST are also provided as a low bound.

1) Same Initial Energy: In this set of simulations, every sensor node has the same initial energy $3000J$. The results are shown in Figure 8, where three curves present the cost of AAML, IRA and MST, respectively. The solid curve shows the cost of AAML in [1]. The curve starts at about 400 and the maximum cost is over 800. The reliability of AAML is between 57% and 75%. The blue curve is the cost of IRA, which starts at about 75 and ends at a little above 225. As we can see, with the same lifetime performance, the total cost of IRA is only about 30% of which given by AAML. The cost of IRA is between 75 and 250, which indicates that the reliability of IRA trees is between 85% and 95%. The black curve presents the cost of MST and it is quite close to the blue curve. As we can see, in most cases, the cost of IRA is only about 20 more than MST, which has no lifetime constraint and is the lower bound of optimal solution to the MRLC problem. We can conclude that our solution is more reliable than AAML and the cost is near optimal.

2) Different Initial Energy: We analyze the performance of IRA under different initial energy in this part. As the energy consumption is different among sensor nodes, it is quite normal that sensor nodes have different energy after the system is running for a while. In this set of simulations, a sensor node has the initial energy randomly selected in $[1500J, 5000J]$. The results are shown in Figure 10. The red curve is for AAML, the blue curve is for IRA and the black one is for MST. As we can see, the IRA and MST curves are more closer than in Figure 9 and AAML also have a better performance. It is mainly because that the network is highly connected (about 70% pairs of nodes are connected). When nodes have different initial energy, the lifetime is mainly based on nodes with less initial energy and they usually are used as leaves in the aggregation trees. Nodes with more initial energy have more choices and thus have a better performance. However, unlike IRA and MST, AAML's performance is quite unstable and the cost is very high in about 30% of cases. In most situations, the cost of AAML is at least 50% higher than that of IRA. We conclude that IRA has a better performance than AAML for different initial energy situations and the cost of IRA is close to the optimal solution.

3) Different Link Probability: In the above simulations, the network is highly connected with a link probability of 70%. In this set of simulations, we give more results with different link probabilities. For each link connection probability, we construct 100 random graphs and the results are shown in Figure 10. The red curve presents the average cost of AAML, the blue curve presents the average of IRA and the black curve is MST. As we can see, the red curve increases as the link probability increases while the IRA and MST almost stays the same under different link connection probabilities. That is mainly because the AAML has more choices as more links are in the network while IRA and MST only cares more about low cost links, other than the number of links.

C. Performance of Distributed Protocol

In this section, we evaluate the performance of the distributed protocol. We use the DFL system as the initial state of the simulation. An data aggregation tree has been constructed

and every node is aware of the Prüfer code of the aggregation. We simulate the distributed protocol by 100 rounds of update. As ILU is based on iteratively parent-changing, we randomly select a tree edge make it unreliable (cost of select edge increases 10^{-3}) in each round. We compare the distributed protocol with the centralized solution IRA.

Figure 11 shows the total cost by two curves both starting from 58. The red curve shows the cost of aggregation trees given by IRA and the blue curve shows the cost of distributed updates. We can see from the figure that IRA has a better performance. However IRA is very complicated and cannot be implemented in a distributed system. The distributed protocol only needs local information and has quite competitive performance. The cost difference of the centralized algorithm and the distributed protocol is only about 25. Figure 12 shows the comparison of reliability between centralized and distributed approaches. Both approaches have a good reliability performance in the beginning and the reliability goes down as links get unreliable. Similar to the cost performance, reliability of the distributed protocol is quite close to IRA and the largest distinction is only about 0.02. We can conclude that both cost and reliability performances of the distributed protocol are comparable to the centralized approach.

We also analyze the message complexity of the distributed protocol. In the distributed updating protocol, the update information should be broadcasted to all sensors through non-leaf nodes. The results of message complexity analysis are shown in Figure 13. The blue curve shows the number of total messages, which increases when more and more updates happen. The red curve shows the average number of messages for an update. The number of messages transmitted varies a lot at the beginning and becomes stable after 40 updates. For each update, it only needs less than 10 messages to broadcast the update information. We can see that the message complexity of the distributed protocol is quite low.

VIII. Conclusion

In this paper, we have studied the Maximizing Reliability of Lifetime Constrained data aggregation tree problem in wireless sensor networks. We have formulated the MRLC problem with a linear program and presented a centralized algorithm IRA. Moreover, we have presented a distributed updating protocol. We have carried out extensive simulations and some of which is based on trace data from a device free localization system. The results have shown that our IRA outperforms AAML by 24% in term of reliability.

Acknowledgment

This work was supported in part by the State Key Development Program for Basic Research of China (973 project 2012CB316201), in part by China NSF grant 61422208, 61472252, 61272443 and 61133006, in part by CCF-Intel Young Faculty Researcher Program and CCF-Tencent Open Fund, and in part by Jiangsu Future Network Research Project No. BY2013095-1-10. The opinions, findings, conclusions, and recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies or the government.

References

[1] Y. Wu, S. Fahmy, and N.B. Shroff, "On the Construction of a Maximum-Lifetime Data Gathering Tree in Sensor Networks: NP-Completeness and Approximation Algorithm", in Proc. INFOCOM, 2008, pp.356-360.

[2] X. Wu, G. Chen, and S. K. Das, "Avoiding Energy Holes in Wireless Sensor Networks with Nonuniform Node Distribution", IEEE Trans. Parallel Distrib. Syst., 2008, pp.710-720.

[3] H.Ö. Tan and I. Körpeoğlu, "Power efficient data gathering and aggregation in wireless sensor networks", in Proc. SIGMOD Record, 2003, pp.66-71.

[4] Y. Xue, Y. Cui, and K. Nahrstedt, "Maximizing Lifetime for Data Aggregation in Wireless Sensor Networks", In Proc MONET, 2005, pp.853-864.

[5] Y. Shen, Y. Li, and Y. Zhu, "Constructing data gathering tree to maximize the lifetime of unreliable Wireless Sensor Network under delay constraint", in Proc. IWCMC, 2012, pp.100-105.

[6] D. Luo, X. Zhu, X. Wu, and G. Chen, "Maximizing lifetime for the shortest path aggregation tree in wireless sensor networks", in Proc. INFOCOM, 2011, pp.1566-1574.

[7] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol", in Proc. SenSys, 2009, pp.1-14.

[8] J. Park and S. Sahni, "An Online Heuristic for Maximum Lifetime Routing in Wireless Sensor Networks", IEEE Trans. Computers, 2006, pp.1048-1056.

[9] http://msoon.github.io/powermonitor/

[10] D.S.J.D. Couto, D. Aguayo, J.C. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing", in Proc. MOBICOM, 2003, pp.134-146.

[11] M. Grötschel, L. Lovász, and A. Schrijver, "The ellipsoid method and its consequences in combinatorial optimization", Combinatorica, 1981, 1(2), 169-197.

[12] L. C. Lau, R. Ravi, and M. Singh, "Iterative methods in combinatorial optimization (Vol. 46)", Cambridge University Press, 2011, Vol. 46.

[13] Y. Wu, X. Li, Y. Liu, and W. Lou, "Energy-Efficient Wake-Up Scheduling for Data Collection and Aggregation", IEEE Trans. Parallel Distrib. Syst., 2010, 21(2), pp.275-287.

[14] M. Fürer and B. Raghavachari, "Approximating the Minimum Degree Spanning Tree to Within One from the Optimal Degree", in Proc. SODA, 1992, pp.317-324.

[15] H. Cheng, Q. Liu, and X. Jia, "Heuristic algorithms for real-time data aggregation in wireless sensor networks", in Proc. IWCMC, 2006, pp.1123-1128.

[16] X. Li, "Localized Construction of Low Weighted Structure and Its Applications in Wireless Ad Hoc Networks", Wireless Networks, 2005, 11(6), pp.697-708.

[17] T. Kuo and M. Tsai, "On the construction of data aggregation tree with minimum energy cost in wireless sensor networks: NP-completeness and approximation algorithms", in Proc. INFOCOM, 2012, pp.2591-2595.

[18] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, "Introduction to Algorithms", MIT Press, 1989.

[19] J. Liang, J. Wang, J. Cao, J. Chen, and M. Lu, "An Efficient Algorithm for Constructing Maximum lifetime Tree for Data Gathering Without Aggregation in Wireless Sensor Networks", in Proc. INFOCOM, 2010, pp.506-510.

[20] L. Levin, M. Segal, and H. Shpungin, "Cooperative data collection in ad hoc networks", Wireless Networks, 2013, 19(2), pp.145-159.

[21] H. Kwon, T. Kim, S. Choi, and B.G. Lee, "Cross-layer lifetime maximization under reliability and stability constraints in wireless sensor networks", in Proc. ICC, 2005, pp.3285-3289.

[22] J. Crowcroft, M. Segal, and L. Levin, "Improved structures for data collection in wireless sensor networks", in Proc. INFOCOM, 2014, pp.1375-1383.

[23] S.K.A. Imon, A. Khan, M.D. Francesco, and S.K. Das, "RaSMaLai: A Randomized Switching algorithm for Maximizing Lifetime in tree-based wireless sensor networks", in Proc. INFOCOM, 2013, pp.2913-2921.