Maximizing Network Lifetime Online by Localized Probabilistic Load Balancing

Yongcai Wang[1], Yuexuan Wang[1], Haisheng Tan[2], and Francis C.M. Lau[2]

[1] Institute for Interdisciplinary Information Science, Tsinghua University, China
[2] Department of Computer Science, the University of Hong Kong, Hong Kong, China
{wangyc,wangyuexuan}@mail.tsinghua.edu.cn{hstan,fcmlau}@cs.hku.hk

Abstract. Network lifetime maximization is a critical problem for long-term data collection in wireless sensor networks. For large-scale networks, distributed and self-adaptive solutions are highly desired. In this paper, we investigate how to optimize the network lifetime by a localized method. Specifically, the network lifetime maximization problem is converted to a localized cost-balancing problem with an appropriately designed local cost function. A distributed algorithm, *LocalWiser*, which adopts the idea of adaptive probabilistic routing, is proposed to construct a localized and self-adaptive optimal solution to maximize the network lifetime. We analyze LocalWiser in both static and dynamic networks. In static networks, it is formally proved that 1) LocalWiser can reach a stable status; 2) the stable status is optimal for maximizing the network lifetime. In dynamic networks, our extensive simulations illustrate that LocalWiser can converge to the optimal status rapidly for the network topology and flow dynamics.

1 Introduction

For long-term autonomous data collection in large-scale wireless sensor networks, it is important for the networks to be not only optimal in lifetime, but also self-adaptive to the online network dynamics. However, even using centralized computation when the sensors are of identical initial energy and of the identical data collection rate, it is NP-Complete to solve maximize the network lifetime by constructing the energy optimal routing tree[8]. If probabilistic multi-path routing is allowed, the network lifetime maximization problem can be solved by a centralized Linear Programming model [2]. However, these centralized methods need global information and global coordination which are not scalable, and they can not be self-adaptive to the online network dynamics, such as the variations of the network topologies and the data flows.

Some other previous works studied the network lifetime maximization by distributed approaches based on load-balancing or probabilistic routing methods. However, because of the restriction of local information and local computation, instability and local optimum are their common weakness. To our best knowledge, in the literature there are no distributed solutions that can simultaneously guarantee 1) stability, 2) network lifetime optimization at the stable status, and 3) self-adaptivity to online network dynamics.

H. Frey, X. Li, and S. Ruehrup (Eds.): ADHOC-NOW 2011, LNCS 6811, pp. 332–345, 2011.

1.1 Our Contribution

Although the network lifetime maximization is a global optimization problem, we can map the problem to a localized cost-balancing problem with an appropriately designed local cost function. As a major contribution, we develop a localized cost-balancing algorithm, *LocalWiser*, to construct a scalable, stable and self-adaptive optimal solution for the network lifetime maximization problem. In the algorithm, we make use of adaptive probabilistic routing and propose a *virtual guidance* for the sensors to overcome instability and local optimum during their local computation.

Further, we formally prove that in static networks 1) LocalWiser is guaranteed to reach a stable status, and 2) the stable status is optimal for maximizing the network lifetime. LocalWiser is also adaptive to changes in the network. When the network topology or data flow changes dynamically, our extensive simulations show that LocalWiser can adjust the transmission probabilities and makes the network converge quickly to the optimal status for network lifetime maximization.

1.2 Related Works

The related work includes research results on network lifetime maximization, load-balancing, energy-efficient routing and distributed consensus. We introduce the most important related contributions in the literature in two categories: the centralized approaches and the distributed approaches.

For the centralized approaches, Buragohain, et al. [2] studied the optimal routing tree problem by modeling the data collection network as a sensor database. They proved that it is NP-Complete to find the energy-optimal routing tree for *partially aggregated* and *unaggregated* queries in random networks. Moreover, they claimed probabilistic routing can improve the network lifetime compared with the tree-based approaches, and proposed a centralized linear programming model to solve the network lifetime maximization problem in polynomial time through probabilistic routing . Liang and Liu [8,9] studied lifetime maximization through the construction of the energy optimized spanning tree. They also proved it is NP-complete and devised several heuristics to prolong the network lifetime. Dai, et al. [3] studied the load balancing problem in a grid topology. They made use of the Chebyshev sum inequality as a criteria for load balancing. All the above solutions require centralized computation and global coordination for the data routing.

For the distributed algorithms, Jarry [5] etc. proved that the flow maximization is equivalent to network lifetime maximization in data collection networks. They investigated the network structures that support the max-flow and proposed two online distributed probabilistic routing algorithms for energy balancing in the optimal network structures. However, in their work, the sensors are allowed to transmit data to the nodes beyond their parent level, which may be hard to implement due to the sensors' limited communication range. Parametric Probabilistic Routing (PPR) [1] addressed the energy efficient routing by assigning higher transmission probabilities to the nodes which are closer to the destination. Dynamic Load Balance Tree(DLBT) algorithm [17] calculated routing probabilities based on the current loads of the parent candidates. The stability of DLBT had been verified by simulations, but its stable status was not guaranteed to achieve the global optimum. Wang et al. [14] presented a level-based load-balancing algorithm to get level-by-level load balancing in the network, which provides a stable but sub-optimal localized solution.

In some other works, Zhu and Girod [19] proposed an algorithm to minimize congestion by decomposing the total target flow rate into a sequence of rate increments, where the classical Bellman-Ford algorithm was adopted to find a corresponding minimum-cost route for each increment. Tsai et al.[12] proposed a routing protocol for load-balancing based on path energy and self-maintenance. Mauro [4] proposed a gossip-based distributed balancing algorithm in which the nodes need extensive information in neighborhood. Most recently, Lee et al.[7] applied DAG's (Direction Acyclic Graph) to different data collection cycles for load balancing so that energy efficiency was improved in both time and spatial domains.

The rest of paper is structured as follows. In Section 2, the network lifetime maximization problem is mapped to a local cost-balancing problem. Section 3 presents the LocalWiser algorithm to solve the local cost-balancing problem. In Section 4, we formally prove the stability and optimality properties of LocalWiser in the static networks. Section 5 illustrates the performance of LocalWiser in static and dynamic networks. There are conclusion remarks and discussion of future works in Section 6.

2 Bridge Network Lifetime to Node's Local Cost

2.1 System Model

We study the online network lifetime maximization problem after a network is deployed over a region for periodical data collection. The network contains N sensors and a single sink, which is organized into a level structure. Each sensor computes its level index by calculating the minimum hops to the sink[18]. The calculation can be carried out distributively after the sink floods several messages. Each sensor can only find one-hop neighbors within its limited communication range. The one-hop neighbors in a node's parent level (one hop closer to the sink) are called *parent candidates* and the node establishes probabilistic links to them for probabilistic minimum hop routing. The one-hop neighbors in the child level (one hop further from the sink) are called *child candidates*, and the other nodes in the same level are called *siblings*.

We consider both static and dynamic networks. In static networks, the sensors are assumed to be stationary, which have identical initial energy and collect identical amount of data in each round. In dynamic networks, the joining, leaving and movement of the sensors are considered as topology dynamics. The changes of the data collection amount in different rounds are considered as flow dynamics. In both types of networks, the sensors communicate with omni-directional antenna and use identical, limited and fixed transmission power. The disc radio model and symmetric channels are assumed.

2.2 Network Lifetime Maximization Problem

We model the above network as a weighted dynamic graph $G = \{\mathcal{V}, \mathcal{E}, \mathcal{P}(t), \mathcal{L}(t)\}$, where $\mathcal{V}$ is the N sensor nodes and $\mathcal{E}$ is the directed links from each node to its parent candidates. The transmission probability of a link $\langle i, j\rangle$ at round t is denoted as $P_{i,j}(t)$. Vector $\mathcal{L}(t) = \{L_i(t)\}$ denotes the *load* of each sensor at round t, where $L_i(t)$ means the amount of data that sensor i needs to transmit in round t. The load of a sensor in round t is the sum of data captured by itself and the loads transmitted from its children:

$$L_i(t) = u_i + \sum_{j \in \mathbb{C}_i} P_{j,i}(t) L_j(t), \tag{1}$$

where $\mathbb{C}_i$ is the child set of node i, and u_i is the locally captured data of node i.

The network lifetime is defined as the network's work duration before its first node dies [2][8][9][14]. Since it is well known that data transmission is dominant in sensor network energy consumption, we can only take into account the transmission energy consumption and assume the other consumptions such as receptions are negligible [8]. Sensors are assumed to consume one unit of energy by transmitting one unit of data. If the network experiences a quick self-optimizing phase to converge to a stable status, where the sensors' loads are stable, node i can estimate its own lifetime by its energy-load ratio, $T_i(s) = e_i/L_i(s)$, where e_i is its initial energy, s is a flag standing for the stable status, and $L_i(s)$ is its stable load. Therefore, we have

Problem 1. The online network lifetime maximization problem is to minimize the maximum load-energy ratio among all the sensors at the stable status.

$$\max\{T(s)\} \Leftrightarrow \min\{\max_i\{L_i(s)/e_i\}\}. \tag{2}$$

Since all data has to be be forwarded by the first level sensors, the sensors with the heaviest load must be in the first level. Therefore, the network lifetime maximization problem can be reduced to the problem of balancing the load among the first level sensors, which meets the lifetime maximization problem formulated in [2][8].

2.3 Local Cost of Network Lifetime

The difficulty to address network lifetime maximization problem locally is lack of the global information at the distributed sensors. Thus, we define a Local Cost function of Network Lifetime (LCNL) at each sensor as a criterion during its local computing to bridge its local decision to the global optimization of the network lifetime.

Definition 1. For a sensor i at round t, its Local Cost function of Network Lifetime (LCNL) is defined as:

$$C_i(t) = \begin{cases} L_i(t)/e_i, & \textit{if } \text{i} \textit{ is in the first level} \\ \sum_{j \in \mathbb{P}_i} P_{i,j}(t) * C_j(t), & \textit{otherwise} \end{cases} \tag{3}$$

If sensor i is in the first level, its LCNL is defined as its load-energy ratio. Otherwise, the LCNL is defined as the sum of the products of its transmission probabilities with its parent candidates' LCNLs. $\mathbb{P}_i$ is the set of all the parent candidates of node i. $P_{i,j}(t)$ is the transmission probability from i to a parent candidate j at round t. Here, we can see 1) the LCNL can be locally computed with the LCNLs of its parent candidates and its local transmission probabilities; 2) the LCNL of each node is a recursive function, which can be expanded to a function of the LCNLs of the first level sensors and the transmission probabilities of the multi-hop links. Moreover, we have

Theorem 1. When all the sensors in the network have equal LCNLs, the network's lifetime is maximized.

Proof. When all the sensors have the equal LCNL, the sensors in the first level must have the same LCNL, i.e., the same load-energy ratio. According to the equivalence in Eqn (2), the lifetime of the network is maximized.

Therefore, the network lifetime maximization problem is mapped to a local cost balancing problem at each sensor. The work left for a sensor is to self-optimize its transmission probabilities locally so that the network converges to a stable status when all the nodes have the equal LCNL.

3 Local Algorithm to Maximize the Network's Lifetime

3.1 Challenges

The local cost balancing problem is difficult due to the issues of instability and local optimum. *Instability* is a general challenge to the concurrent distributed decisions, which prevents the system to reach a consensus [11]. In our problem, at each round one node is not aware of the concurrent changes to the probability assignments of its siblings. For example, when some adjacent siblings observe the load of a parent candidate is too high, they will react to this local observation by decreasing their own transmission probabilities to the parent candidate. The changes may lead to an overreaction which make the load of the parent from over-high to over-low. Such kinds of overreaction result in "instability", which prevents the network from converging to a stable status. *Local optimum* is also a general difficulty in distributed algorithms [14], and is a challenge of the local cost balancing problem. Since each node only has local information, it cannot know the global status of the network. Even if a stable status can be reached, local algorithms may stop at a local optimum. i.e., it converges to a sub-optimal status and cannot proceed towards the global optimum any more.

3.2 "Virtual Guidance" in LocalWiser

To overcome instability and local optimum, we propose a 'virtual guidance'. We suppose that all the sensors can reach an ideal status where all of their LCNLs are equal to an expected average cost $E(C)$. If an online sensor can know the value of $E(C)$, it will have a guidance for its local computation to avoid overreaction and local optimum. But unfortunately, the value of $E(C)$ can not be computed by any sensor locally. Therefore we can only suppose the sensors in the network have the value of $E(C)$ virtually. Actually, it will be shown that the exact value of $E(C)$ will not be needed in the our final algorithm.

For a node i, $j \in \mathbb{P}_i$ is one of its parent candidates. After round $t-1$, the transmission probability from i to j is $P_{i,j}(t-1)$ and the LCNL of node j is $C_j(t-1)$. To update the transmission probability at round t, node i is virtually guided to balance $C_j(t-1)$ towards $E(C)$. Therefore, it sets the transmission probability $P_{i,j}(t)$ as:

$$P_{i,j}(t) = P_{i,j}(t-1)\frac{E(C)}{C_j(t-1)M_i(t)}, \tag{4}$$

where $M_i(t) = \sum\limits_{j \in \mathbb{P}_i} P_{i,j}(t-1)\frac{E(C)}{C_j(t-1)}$ is a normalizer that keeps the sum of the transmission probabilities of node i equal to 1. By substituting $M_i(t)$ in Eqn (4), we have

$$P_{i,j}(t) = \frac{P_{i,j}(t-1)/C_j(t-1)}{\sum_{j \in \mathbb{P}_i} P_{i,j}(t-1)/C_j(t-1)}. \tag{5}$$

The most interesting point in Eqn(5) is that $E(C)$ is counteracted. The reason is that every sensor only needs a qualitative guidance, which is the LCNL of the parent level nodes should be equal, but not the exact value of $E(C)$. In static networks, when LCNLs of all sensors are balanced, the value of $E(C)$ is reached automatically.

3.3 LocalWiser Algorithm

Based on the above idea, the LocalWiser algorithm is derived based on time synchronization and level-based scheduling. In each round, the sensors are scheduled according to their level indices. As shown in Fig.1, in each round, the sensors in the child level collect data, run LocalWiser and transmit data before the parent level nodes. This meets the general transmission scheduling schemes for energy efficient multi-hop data aggregation[6]. The issues of time synchronization, collision avoidance and scheduling are studied in [10][13]. The LocalWiser algorithm for a sensor i in round t is given in Algorithm 1.

- In line 11, the node updates its local load information. Since the children of node i execute LocalWiser before it, node i can use the children's loads at round t to update its load information.
- In line 13, the node updates its LCNL information. When node i executes LocalWiser, their parents have not executed LocalWiser. Therefore, node i uses $C_j(t-1)$ which is the LCNL information in the $t-1$ round to update $C_i(t)$.

In Algorithm 1, we can see during the local computation, no global information is used in LocalWiser. Moreover, in the following sections, we will analyze our algorithm in both static and dynamic networks to illustrate its properties of stability, optimality and good self-adaptivity to network dynamics.

Algorithm 1. LocalWiser Algorithm for a sensor i, in round t

while $(t < \infty)$ **do**
  if $(t == 0)$ **then**
    for $(n{=}1,\ j = \mathbb{P}_i(n);\ n <= \#\mathbb{P}_i;\ n++)$ **do**
      $P_{i,j}(t) = 1/(\#\mathbb{P}_i)$ //initiate $P_{i,j}$
    end for
  else
    for $(n{=}1,\ j = \mathbb{P}_i(n);\ n <= \#\mathbb{P}_i;\ n++)$ **do**
      $P_{i,j}(t) = \frac{P_{i,j}(t-1)/C_j(t-1)}{\sum_{j\in\mathbb{P}_i} P_{i,j}(t-1)/C_j(t-1)}$
      $L_i(t) = u_i + \sum_{m\in\mathbb{C}_i} P_{m,i}(t)L_m(t)$
      $C_i(t) = \begin{cases} L_i(t)/e_i, & i \in \text{level } 1 \\ \sum\limits_{j\in\mathbb{P}_i} P_{i,j}(t) * C_j(t-1), & i \notin \text{level } 1 \end{cases}$
    end for
  end if
end while

4 Stability and Optimality of LocalWiser

4.1 Stability

Stability is proved in two steps. 1). we prove the maximum LCNL of all the nodes (denoted by $C_{max}(t)$) will converge to a stable value. 2). When $C_{max}(t)$ converges, we can finally prove the stability of LocalWiser.

Lemma 1 (Convergence of $C_{max}(t)$). *The maximum LCNL of all the nodes, $C_{max}(t)$ must converge to a stable value.*

Proof. $\forall i \notin$ level 1

$$\begin{aligned} C_i(t) &= \sum_{j\in\mathbb{P}_i} P_{i,j}(t)\cdot C_j(t-1), \\ &= \sum_{j\in\mathbb{P}_i}\left(\frac{P_{i,j}(t-1)/C_j(t-1)}{\sum_{j\in\mathbb{P}_i} P_{i,j}(t-1)/C_j(t-1)}\cdot C_j(t-1)\right) \\ &\le \sum_{j\in\mathbb{P}_i}\left(\frac{P_{i,j}(t-1)}{\sum_{j\in\mathbb{P}_i} P_{i,j}(t-1)/C_{max}(t-1)}\right) \\ &\le \sum_{j\in\mathbb{P}_i}\left(\frac{P_{i,j}(t-1)}{1/C_{max}(t-1)}\right) \le C_{max}(t-1) \end{aligned} \tag{6}$$

For $\forall i \in$ level 1, it can be proved similarly by expanding Eqn (1) to get $C_i(t) \le C_{max}(t-1)$. Because $C_{\max}(t) = \max\{C_1(t), ..., C_N(t)\}$, we get

$$C_{\max}(t) \le C_{\max}(t-1). \tag{7}$$

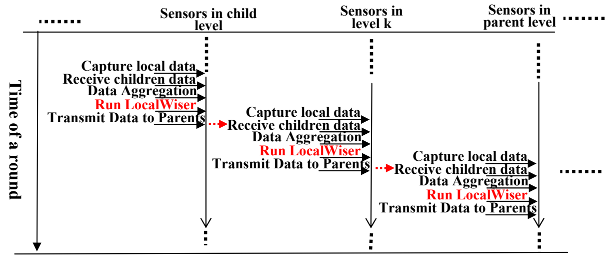

Fig. 1. The work-flow of sensors in level-based scheduling

Since $C_{max}(t)$ is lowerbounded, it cannot be smaller than the average load of the first level nodes. As time passing, $C_{max}(t)$ will be infinitely close to $C_{max}(t-1)$ or equal to $C_{max}(t-1)$. Therefore, Lemma 1 is proved. ■

When $C_{max}(t)$ is stable, there is at least one node having LCNL equal to $C_{max}(t)$. We call such nodes as the maximum cost nodes (*MCNs*).

Lemma 2. When $C_{max}(t)$ is stable, if a node is a MCN, all its ancestors that it can reach by positive pathes (links on the path have positive transmission probabilities) are MCNs.

Proof. Without loss of generality, we suppose node i is a MCN. From Eqn.(6), the condition of $C_i(t) = C_{max}(t) = C_{max}(t-1)$ is:

$$\sum_{j \in \mathbb{P}_i} P_{i,j}(t-1)/C_j(t-1) = 1/C_{max}(t-1) \tag{8}$$

The condition that Eqn. (8) holds is :$\forall j \in \mathbb{P}_i$, if $P_{i,j}(t-1) > 0$, then $C_j(t-1) = C_{max}(t-1) = C_{max}(t)$. Therefore, if the transmission probability from a MCN to a parent candidate is positive, this parent candidate must also be a MCN. The same condition will also holds by its parent candidates. Therefore, all its ancestors that it can reach by positive pathes are MCNs. ■

Lemma 3. When $C_{max}(t)$ is stable, if a node is not a MCN but it has some MCNs as its parent candidates, the transmission probabilities from it to these MCNs will converge to 0.

Proof. Suppose a node i is not a MCN with local cost $C_i(t) < C_{max}(t)$. Suppose it has a parent j who is a MCN. From Eqn.(5), we have

$$\frac{P_{i,j}(t)}{P_{i,j}(t-1)} = \frac{1/C_{max}(t-1)}{\sum_{m \in \mathbb{P}_i} P_{i,m}(t-1)/C_m(t-1)} < 1 \tag{9}$$

As time passes, the transmission probability from i to j will be decreased until:

$$P_{i,j}(t) = \frac{P_{i,j}(t-1)/C_{max}(t-1)}{\sum_{m \in \mathbb{P}_i} P_{i,m}(t-1)/C_m(t-1)} = P_{i,j}(t-1) \tag{10}$$

The only condition that Eqn.(10) holds is $P_{i,j}(t) = P_{i,j}(t-1) = 0$. Therefore, the transmission probability from i to j will be decreased monotonously to 0. ■

Theorem 2 (Stability). The LocalWiser algorithm can guarantee the network converge to a stable status, where both the LCNLs and the transmission probabilities of all sensors are stable.

Proof. We proof Theorem2 by an isolation and reconstruction method. Lemma2 and Lemma3 have proved that after $C_{\max}(t)$ is converges, if a node is a MCN, all its parent candidates with positive transmission probabilities are MCNs; if a node is not a MCN, its transmission probabilities to the MCNs must converge to zero. Therefore, the MCNs can be isolated from the network, i.e., the network can be divided into a MCN set and a non-MCN set. The transmission probabilities between the MCN set and the non-MCN set are zero. If we subtract the nodes in MCN set and the links among MCNs from the network, the costs and the transmission probabilities of the nodes left will not be affected. After the substraction, Lemma1, Lemma2 and Lemma 3 will still hold for the remaining network, and a new set of MCNs can be determined in the remaining network. We can repeat the substraction process until all the remaining nodes have the equal LCNLs, which reaches a stable status. After we add the removed MCNs and the links back into the network in the reversed order, the stable scenario of the network is obtained. ■

Corollary 1 (Properties of the stable status of the network). *When the network reaches the stable status, we have*

1. *The network is composed of K isolated sets, where K is a positive integer.*
2. *Nodes in each isolated set have equal LCNL. The transmission probabilities among nodes in different sets are zero.*
3. *Each isolated set contains at least one first level nodes, and the first level nodes are fully covered by the K isolated sets.*

Proof. Property 1) and property 2) are directly proved by Theorem 2. For property 3), Lemma 3 gives that two nodes in different isolated sets cannot have positive pathes to the same node, so the first level nodes are covered by the K isolated sets without overlapping. ■

Fig.2 illustrates the stable status of a network. Three isolated sets are formed in the network. The first set contains MCNs of the whole network; the second contains MCNs after substraction of the first set; the third contains MCNs after substraction of the first and the second sets. In each set all nodes have equal LCNL. The transmission probabilities between any two different sets have converged to zero.

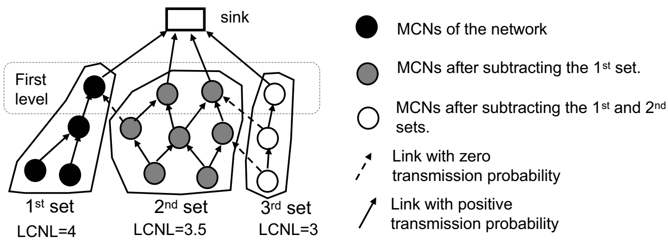

Fig. 2. The stable scenarios of a network in which nodes form isolated sets

4.2 Optimality

We prove the stable status of LocalWiser is optimal to maximize the network lifetime. A Chebyshev's Sum Inequality based metric is used to measure the load-balancing performance of the first level sensors [3].

$$\theta = \frac{\left(\sum_{i=1}^{q} L_i(s)\right)^2}{q\sum_{i=1}^{q}(L_i(s))^2} = \frac{\left(\sum_{k=1}^{K} n_k \ell_k\right)^2}{q\sum_{k=1}^{K} n_k(\ell_k)^2}, \tag{11}$$

where $L_i(s)$ is the stable load of a first level node i. s is a stable flag. The right part of Eqn. (11) is a set representation of the metric. We suppose the network contains q first level nodes, which are fully covered by the K isolated sets. The first level nodes in the kth set are denoted by a set $\mathbb{S}_k$, and the number of first level nodes in $\mathbb{S}_k$ is denoted by n_k; The load of the first level sensors in $\mathbb{S}_k$ is denoted by ℓ_k.

It is easy to verify that θ is not larger than 1. Only if all the first level nodes have equal loads, $\theta = 1$; Otherwise, $\theta < 1$. The smaller θ is, the worse the load balancing performance is.

Theorem 3 (Optimality). When the network has reached stable status, any modification to the probability assignments at any node can only make the load balancing performance of the first level nodes worse.

Proof. We consider different cases of probability assignment modifications.

1). If the transmission probability modifications are within one isolated set (say the kth set), only the loads of the first level nodes in this set will be affected. Suppose after modification the loads of the first level sensors in this set are $\{\ell_k + x_j\}$, where $\sum_{j\in\mathbb{S}_k}(x_j) = 0$ as the total loads in the set will not change. The load-balancing metric after modification is:

$$\theta' = \frac{\left(\sum_{i=1}^{K} n_i \ell_i\right)^2}{q\left(\sum_{i=1, i\neq k}^{K} n_i(\ell_i)^2 + \sum_{j\in\mathbb{S}_k}(\ell_k + x_j)^2\right)} \tag{12}$$

As $\sum_{j\in\mathbb{S}_k}(\ell_k + x_j)^2 > n_k\ell_k^2$, we have $\theta' < \theta$, which means the load balancing performance of the first level nodes becomes worse after the modification.

2). Consider the case when the transmission probabilities among different sets are modified from zero to some positive values (the dashed links shown in Fig.2). Without loss of generality, we suppose the first level sensors of loads ℓ_k and ℓ_j are affected by such modifications. If in stable status $\ell_k > \ell_j$, the modifications can only further increase the loads of ℓ_k and further decrease the loads of ℓ_j. After the network converging to a new stable status, if no other sets are affected, the new stable loads of the two sets will be $\ell_k + \frac{x}{n_k}$ and $\ell_j - \frac{x}{n_j}$ respectively, where x is a positive value indicating the data transmission from set j to set k. It is easy to verify by Rearrangement Inequality [16] that $\theta' < \theta$, and the load-balancing performance of the first level becomes worse after the modification.

If the inter-set modifications affect more than two sets, since the modification can only increase the loads of the larger-load nodes and decrease the loads of the smaller-load nodes in the first level, we can prove the load-balancing metric will also be worse similarly according to the Rearrangement Inequality.

Based on the cases in 1) and 2), the stable status of LocalWiser is optimal to maximize the network lifetime. ■

5 Numerical Results in Static and Dynamic Networks

5.1 Simulation Setups

For a static network, the sensors are randomly deployed to a 2D space in a uniform random distribution. A sink is deployed at the center of the network. Sensors are organized into levels according to their minimum hops to the sink. Each sensor establishes probabilistic links to its parent candidates. The communication radius of the sensors is set identical and normalized to $r = 1$. All sensors' initial energy is set to 1.

Four other algorithms are used during the performance comparison, which are 1)Centralized algorithm using global information and Linear Programming (LP)[2], which is the global optimal result; 2) Greedy algorithm for local load-balancing (Greedy); 3)Dynamic Load-Balanced routing Tree(DLBT)[17]; 4) Level-based load balancing algorithm (Level-based Balancing [14]).

5.2 Performance in Static Networks

In static networks, the sensors collect constant amount of data in each round and the network topology doesn't change. In our simulation, a high density network of 300 nodes is randomly deployed in a 7*7 area, and a low density network of 100 nodes is randomly deployed in a 10*10 area. Each node collects one unit data in each round. To evaluate network lifetime, the minimum energy-load ratio of all sensors, i.e., $\frac{1}{\max_{i=1}^{N}(L_i(t))}$ is plotted as the evaluation metric for different algorithms. Fig.(3) shows the performances of the different algorithms in the high density network, and Fig.(4) shows the performances in the low density network. For the networks of different densities, LocalWiser can always converge to a stable status where it reaches the global optimal performance as the same as

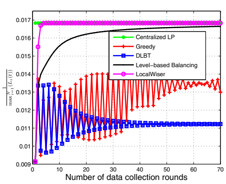

Fig. 3. Performances in a high-density static network

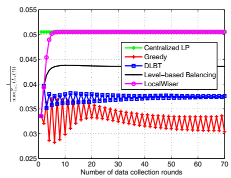

Fig. 4. Performances in a low-density static network

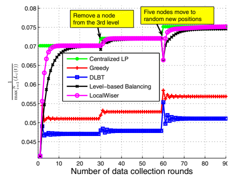

Fig. 5. Self-adapting to topology dynamics

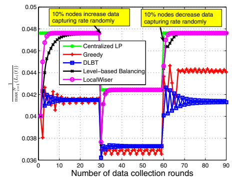

Fig. 6. Self-adapting to flow dynamics

the centralized Linear Programming. We can find the instability of the greedy algorithm from the zigzag performance curves. The DLBT algorithm and the Level-based Balancing can only reach the local optimum. Evaluations in many other networks show the similar results[15]. Therefore, in our simulations LocalWiser guarantees stability and the global optimal network lifetime by localized methods.

According to the simulations, the convergence speed of LocalWiser in static networks can be summarized as: 1) quick convergence to good-enough, and 2) slow improvement from good-enough to the optimum. LocalWiser improves $\frac{1}{\max_{i=1}^{N}(L_i(t))}$ quickly within several rounds to reach a good-enough performance which is very close to the optimum. The reason of "slow convergence from good-enough to optimum" is in Eqn.(5). When $P_{i,j}(t-1)$ is very close to zero, $P_{i,j}(t)$ gets very small gain in every step. Since the good-enough performance is already very close to the global optimum, LocalWiser algorithm provides a fast network lifetime maximizing methods for practical applications.

5.3 Performances in Dynamic Networks

We evaluate LocalWiser in two kinds of dynamic networks: 1) the network's topology is dynamic; 2) the amounts of data collected by sensors are dynamic in different rounds. If the dynamic events are highly frequent, it will be hard to provide an adaptive and online optimal solution, specifically when the variation speed exceeds the adaptation speed. Therefore, we focus on analyzing the adaptation speed of LocalWiser with the infrequent network dynamics.

1) Mechanisms to handle network topology dynamics

Topology dynamics include the joining, leaving and movements of the nodes. When a new node joins the network, it will discover its parent candidates and child candidates by local communications according to the level indices of its neighbors. Then, it assigns transmission probabilities evenly to the parent candidates and notifies the children to reset their transmission probabilities. Node leaving is handled similarly. The children of the leaving node will reset its transmission probabilities to exclude the leaving node. The parent candidates of the leaving node will also notify their children to reset their transmission probabilities. Node movements are treated as a merge event of the node leaving and joining.

An example is shown in Fig.5, where the network has 100 nodes and eight levels. LocalWiser converges to a stable status after about five rounds. In the 30th round when we remove a node from the third level, LocalWiser leads all the sensors to converge to a new stable and optimal status quickly. In the 60th round when five nodes are moved to some random new positions, the network self-adapts to the movement quickly and also converges to a new stable and optimal status after several rounds.

2) Adaptivity to flow dynamics

Nodes may capture different amounts of data in different rounds, which is called flow dynamics. LocalWiser can self-adapt to such flow dynamics and maintains the network to online stable and optimal status.

An example is shown in Fig.6, which is a network of 100 nodes. It converges to a stable status after about five rounds. After that, in the 30th round, 10% nodes increase data capturing rates randomly. LocalWiser uses about five rounds to adapt to such flow dynamics and converges to a new stable and optimal status. In the 60th round, when 10% nodes decrease the data capturing rates randomly, LocalWiser uses about five rounds to converge to another new stable and optimal status again.

We have also evaluated the performance of self-adaptivity of LocalWiser in many other networks and find that the time for LocalWiser to adapt to the network dynamics is only several rounds in general. More simulation results can be found in our online demo[15]. Therefore, we can say when the network dynamics are not very frequent, LocalWiser is a distributed and adaptive online network lifetime maximization method.

6 Conclusion

In this paper, we study the network lifetime maximization problem by mapping it to the localized cost-balancing problem. An algorithm, LocalWiser, is proposed to maximize the network lifetime, which is of the following characteristics: 1) computing locally and distributedly; 2) stable; 3) optimal to maximize the network lifetime in the stable status; 4) fast self-adapting to the network dynamics; and 5) easily implemented. In future work, the localized cost-balancing algorithm can be improved to jointly optimize the transmission power and the transmission probability. Mobility of the sensors can improve the performance of LocalWiser. The convergence speed of LocalWiser is also our further work. In addition, LocalWiser can be extended to multi-hop heterogenous networks and multi-hop networks with queues.

Acknowledgement

This work was supported in part by the National Basic Research Program of China Grant 2007CB807900, 2007CB807901, the National Natural Science Foundation of China Grant 61073174, 61033001, 61061130540, and the Hi-Tech research and Development Program of China Grant 2006AA10Z216.

References

1. Barrett, C.L., Eidenbenz, S.J., Kroc, L., Marathe, M., Smith, J.P.: Parametric probabilistic sensor network routing. In: WSNA 2003, pp. 122–131. ACM, New York (2003)
2. Buragohain, C., Agrawal, D., Suri, S.: Power aware routing for sensor databases. In: INFOCOM 2005, pp. 1747–1757, 2881 (2005)
3. Dai, H., Han, R.: A node-centric load balancing algorithm for wireless sensor networks. In: Globecom 2003, pp. 548–552, 4209 (2003)
4. Franceschelli, M., Giua, A., Seatzu, C.: Load balancing over heterogeneous networks with gossip-based algorithms. In: ACC 2009, pp. 1987–1993, 5802 (2009)
5. Jarry, A., Leone, P., Nikoletseas, S., Rolim, J.: Optimal data gathering paths and energy balance mechanisms in wireless networks. In: Rajaraman, R., Moscibroda, T., Dunkels, A., Scaglione, A. (eds.) DCOSS 2010. LNCS, vol. 6131, pp. 288–305. Springer, Heidelberg (2010)
6. Krishnamachari, B., Estrin, D., Wicker, S.B.: The impact of data aggregation in wireless sensor networks. In: ICDCSW 2002, pp. 575–578. IEEE Computer Society, Washington, DC, USA (2002)
7. Lee, H., Keshavarzian, A., Aghajan, H.: Near-lifetime-optimal data collection in wireless sensor networks via spatio-temporal load balancing. ACM Trans. Sen. Netw. 6, 26:1–26:32 (2010)
8. Liang, W.F., Liu, Y.Z.: Online data gathering for maximizing network lifetime in sensor networks. Ieee Transactions on Mobile Computing 6(1), 2–11 (2007)
9. Luo, W.L.: Prolonging network lifetime for data gathering in wireless sensor networks. Submitted to IEEE Trans. Computers (2010)
10. Maróti, M., Kusy, B., Simon, G., Lédeczi, A.: The flooding time synchronization protocol. In: SenSys 2004, pp. 39–49 (2004)
11. Olfati-Saber, R., Fax, J.A., Murray, R.M.: Consensus and cooperation in networked multi-agent systems. Proceedings of the Ieee 95(1), 215–233 (2007)
12. Tsai, Y.-P., Liu, R.-S., Luo, J.-T.: Load balance based on path energy and self-maintenance routing protocol in wireless sensor networks. In: Hong, C.S., Tonouchi, T., Ma, Y., Chao, C.-S. (eds.) APNOMS 2009. LNCS, vol. 5787, pp. 431–434. Springer, Heidelberg (2009)
13. Wang, L., Xiao, Y.: A survey of energy-efficient scheduling mechanisms in sensor networks. Mob. Netw. Appl. 11, 723–740 (2006)
14. Wang, Y.C., Wang, Y.X., Lin, H.: Level-based load-balancing by tuning transmission probabilities locally in wireless sensor networks. Submitted to Computer Communications
15. Wang, Y.C., Wang, Y.X., Qi, X.: Guided-evolution:convergence to global optimal by distributed computing using local information. In: Demo in MobiCom 2010 (2010), `http://www.project.itcs.tsinghua.edu.cn/balance`
16. Wu, K., Liu, A.: Rearrangement inequality. Mathematics Competitions 8, 53–60 (1995)
17. Yan, T., Bi, Y., Sun, L., Zhu, H.: Probability based dynamic load-balancing tree algorithm for wireless sensor networks. In: Lu, X., Zhao, W. (eds.) ICCNMC 2005. LNCS, vol. 3619, pp. 682–691. Springer, Heidelberg (2005)
18. Zheng, M., Zhang, D., Luo, J.: Minimum hop routing wireless sensor networks based on ensuring of data link reliability. In: MSN 2009, pp. 212–217 (2009)
19. Zhu, X.Q., Girod, B.: A distributed algorithm for congestion-minimized multi-path routing over ad hoc networks. In: ICME 2005, pp. 1485–1488, 1598 (2005)