Available online at www.sciencedirect.com
ScienceDirect
Ad Hoc Networks 6 (2008) 380–392
Ad Hoc Networks
www.elsevier.com/locate/adhoc

Data aggregated maximum lifetime routing for wireless sensor networks ☆

Cunqing Hua *, Tak-Shing Peter Yum

Department of Information Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong

Received 12 September 2006; received in revised form 9 February 2007; accepted 12 February 2007
Available online 28 February 2007

Abstract

In this paper, we present a data aggregated maximum lifetime routing scheme for wireless sensor networks. We address the problem of jointly optimizing data aggregation and routing so that the network lifetime can be maximized. A recursive smoothing method is adopted to overcome the non-differentiability of the objective function. We derive the necessary and sufficient conditions for achieving the optimality of the optimization problem and design a distributed gradient algorithm accordingly. Extensive simulations are carried out to show that the proposed algorithm can significantly reduce the data traffic and improve the network lifetime. The convergence property of the algorithm is studied under various network configurations.

Keywords: Sensor networks; Maximum lifetime routing; Data aggregation; Recursive smoothing

1. Introduction

Energy-efficient routing [1–3] has long been studied in the context of wireless ad hoc networks and sensor networks. The basic idea is to deliver packets through the minimum energy paths so as to reduce the end-to-end energy consumption. But this class of approaches tends to overwhelm the minimum energy paths, causing nodes on the paths to run out of battery energy quickly and break the network connectivity. This is undesirable for sensor networks since sensor nodes have to collaborate for common mission, the failure of nodes may break the network functionality.

To cope with this problem, maximum lifetime routing has been proposed recently [4–7]. The key idea is to maximize the network lifetime by balancing the traffic load across the network. These solutions are applicable for ad hoc networks where traffic is conserved between source and destination nodes. However, data collected by sensor nodes may contain redundant information due to the spatial–temporal correlation. Therefore, it is desirable to aggregate the data at the intermediate nodes to remove the redundant information. A few schemes have been proposed to exploit this feature to improve

☆ This work was supported in part by the Hong Kong Research Grants Council under Grant CUHK 4220/03E. Part of this work appears in IFIP Networking 2006.
* Corresponding author. Tel.: +852 26098493; fax: +852 26035032.
E-mail addresses: chua0@ie.cuhk.edu.hk (C. Hua), yum@ie.cuhk.edu.hk (T.-S. Peter Yum).

1570-8705/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.adhoc.2007.02.006

the network performance [8–13]. For example, in [14], the authors propose a Minimum Energy Gathering Algorithm (MEGA). This algorithm requires to maintain two trees – the coding tree for raw data aggregation and the *shorted path tree* (SPT) for delivering the compressed data to the sink node. These work demonstrate that data aggregation can improve the performance of various communication protocols (channel coding, routing, MAC, etc.).

However, existing work does not consider data aggregation with maximum lifetime routing scheme. By jointly optimizing routing and data aggregation, the network lifetime can be extended from two aspects. First, data aggregation can help to reduce the traffic to the sink node, which in turn can reduce the power consumption of intermediate nodes. Second, maximum lifetime routing can balance the traffic across the network so as to avoid overwhelming the bottleneck nodes. In this work, we present a model that can optimize routing and data aggregation simultaneously. The basic idea is to adopt the geometric routing [15] whereby the routing is determined solely by the nodal position. We associate each link with two variables, one for data aggregation, the other for routing. This allows the *foreign-coding* [14] model to be incorporated without intervening the underlying routing scheme. The problem is therefore focused on computing the optimal set of variables so that the network lifetime can be maximized subject to energy constraints. Since the problem cannot be solved directly using the simple distributed methods, we adopt a recursive smoothing function to approximate the original problem. We derive the necessary and sufficient conditions required to achieve the optimality of the smoothed problem. A distributed gradient algorithm is designed accordingly with which nodes can compute their variables using the information from neighbors. It is shown by simulations that the proposed scheme can significantly reduce the traffic and improve the network lifetime. The distributed algorithm can converge to the optimal points efficiently under various network configurations.

In the following, we first introduce the system models and define the data aggregated maximum lifetime routing problem in Section 2. We then introduce the recursive smoothing method in Section 3 and derive the optimality conditions in Section 4. The implementation issues of the distributed algorithm are discussed in Section 5 and performance evaluation is presented in Section 6. Finally we conclude this paper in Section 7.

2. System models and problem formulation

In this section, we first introduce the routing model, the data correlation and aggregation model, and the power consumption model. Based on these models, we define the network lifetime and formulate the optimization problem.

2.1. Routing model

We consider a wireless sensor network with a set of sensor nodes N that generate data constantly, and a single sink node s that is responsible for collecting data from sensor nodes. Each node has multiple routing paths to the sink node. The routing algorithm suitable for use belongs to the class of *geometric routing* algorithms [15]. Every sensor node is assumed to know its own position as well as that of its neighbors, which can be obtained with some positioning schemes [16,17]. Each node can forward packets to its neighbor nodes within its transmission range that are closer to the sink node than itself. Since nodes can make routing decisions based on the position information of its neighbors and the sink node, this routing algorithm is localized and particularly suitable for large-scale sensor networks.

Let N_i denote the set of neighbors of node i and $N_i = \{j|d_{ij} \leqslant R,\ j \in N\}$, where d_{ij} is the Euclidean distance of node i and j, and R is the radius of the transmission range. According to the *geometric routing*, only those neighbors that are closer to the sink node s can serve as the downstream nodes. Let us denote this set of downstream neighbors as $S_i = \{k|d_{ks} < d_{is},\quad k \in N_i\}$. Similarly, the set of upstream neighbors is denoted as $A_i = \{k|d_{ks} \geqslant d_{is}, k \in N_i\}$. Note that in case a node has no neighbors that are closer to the sink node than itself, we encounter a problem known as "local maximum" where the node fails to find routing path to the sink node according to *geometric routing*. A few solutions have been proposed for this problem [18–20]. However, the consideration of these solutions is beyond the scope of this paper. In the following, we assume that the downstream neighbor set S_i is non-empty for all $i \in N$.

2.2. Data correlation and aggregation model

In sensor networks, data collected by neighboring nodes is normally correlated due to the spatio-temporal characteristics of the physical medium being sensed, such as the temperature and humidity sensors in a similar geographic region, or magnetometric sensors tracking a moving vehicle. As a result, the data collected by sensor nodes often carries redundant information. Data aggregation (combining the data at the intermediate nodes) is an effective way to remove the redundant information and reduce the traffic. To incorporate data aggregation into the *geometric routing* model, we adopt the *foreign-coding* model [14] scheme. Specifically, we assume a node i is able to compress the data originating at its upstream neighbor j using its local data. The compression ratio depends on the data correlation between node i and j, which is denoted by the correlation coefficient $q_{ji} = 1 - H(X_j|X_i)/H(X_j)$, where $H(X_j)$ is the entropy coded data rate of the information X_j at node j, and $H(X_j|X_i)$ is the conditional entropy coded data rate of the same information X_j at node i given the side information X_i. Examples of correlation models include the *Gaussian random field* model [10] which assumes that the correlation coefficient q_{ji} decreases exponentially with the distance between nodes, or $q_{ji} = \exp(-\alpha d_{ji}^2)$, and the inverse model [14] which assumes the data correlation is inversely proportional to the Euclidean distance between nodes, or $q_{ij} = 1/(1 + d_{ji})$.

Using the *foreign-coding* model, the traffic of a node is classified into two categories: transit data from upstream neighbors and local data generated by itself. To separate the routing of these types of traffic, each node maintains two routing variables ϕ_{ik} and ψ_{ik} for the link to its downstream neighbor k, where ϕ_{ik} denotes the fraction of transit data to be routed from node i to node k, and ψ_{ik} denotes the fraction of local data to be routed from node i to node k. Clearly, it is required that $\sum_{k\in S_i}\phi_{ik} = 1$ and $\sum_{k\in S_i}\psi_{ik} = 1$.

The data aggregation and routing work as follows. For the local data generated by node i itself, it is directly forwarded to the downstream neighbors according to routing variables ψ_{ik}s. For the data received from upstream neighbor j (which contains both raw data generated by node j and transit data passing through node j), node i performs two different operations. For the raw data of node j, it is encoded with the local information, while for the transit data passed from node j (which has been encoded by node j or its upstream nodes), no further encoding is performed. All these transit traffic is forwarded to the downstream neighbors according to the other set of variables ϕ_{ik}s. Mathematically, let r_j denote the data generating rate of node j, λ_i and λ_j denote the aggregated transit traffic rate at node i and j, respectively. The aggregated transit traffic of node i is a superposition of two parts: the transit traffic passed from the upstream nodes, and the raw data originated from the upstream nodes that is to be encoded using the local information. That is,

$$\lambda_i = \sum_{j\in A_i}[\lambda_j\phi_{ji} + r_j\psi_{ji}(1 - q_{ji})]. \tag{1}$$

2.3. Power consumption model

The power consumption of a sensor node consists of four parts: sensing and generating data, idling, receiving, and transmitting. The power e_g for generating one bit of data is assumed to be the same for all nodes. The idle power consumed by a node, again assumed to be the same for all nodes and independent of traffic, is denoted by e_s. For power consumption in receiving and transmitting, we adopt the *first order radio model* in [1]. Specifically, a node needs $\epsilon_{\text{elec}} = 50$ nJ to run the circuitry and $\epsilon_{\text{amp}} = 100$ pJ/bit/m^2 for the transmitting amplifier. Therefore, the power consumption for receiving one bit of data is given by $e_r = \epsilon_{\text{elec}}$. The power consumption for transmitting one bit of data to a neighbor node j is given by $e_{ij} = \epsilon_{\text{elec}} + \epsilon_{\text{amp}} \cdot d_{ij}^n$, where n is the path loss exponent, which typically ranges between 2 and 4 for free-space and short-to-medium-range radio communication. Let E_i denote the initial battery energy of node i, and w_i denote the fraction of power consumption in each time unit. We have

$$w_i = \left(e_s + e_g r_i + e_r\sum_{j\in A_i}(\lambda_j\phi_{ji} + r_j\psi_{ji}) + \sum_{k\in S_i}e_{ik}(\lambda_i\phi_{ik} + r_i\psi_{ik})\right)\Bigg/E_i, \tag{2}$$

where the first term is the idling power consumption, the second term is the power for sensing, the third term is the power consumption for receiving and the last term is the power consumption for transmitting.

2.4. Data-aggregated maximum lifetime routing problem

The lifetime T_i of node i is the time for the node to run out of the battery energy. Since w_i is the fraction of power consumed in each time unit, it is obvious that $T_i = 1/w_i$. Similar to [5,6], we define the network lifetime T_{net} as the time at which the first node in the network runs out of energy, that is

$$T_{\text{net}} = \min_{i\in N} T_i. \tag{3}$$

The power consumption w_i is a function of $\boldsymbol{r}$, $\boldsymbol{\lambda}$ and $\boldsymbol{\phi}$. However, the set of aggregated transit traffic $\boldsymbol{\lambda}$ can be obtained from $\boldsymbol{r}$, $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ using (1). Therefore, T_{net} depends only on $\boldsymbol{r}, \boldsymbol{\phi}, \boldsymbol{\psi}$ and the initial battery energy $\boldsymbol{E}$. If $\boldsymbol{r}$ and $\boldsymbol{E}$ are given, the network lifetime is solely determined by the set of variables $\{\boldsymbol{\phi}, \boldsymbol{\psi}\}$. We therefore define the Data-Aggregated Maximum Lifetime Routing (DA-MLR) problem as follows:

DA-MLR: Given the traffic generating rate $\boldsymbol{r} = \{r_i\}$, the initial battery energy $\boldsymbol{E} = \{E_i\}$ and the data correlation coefficient $\boldsymbol{q} = \{q_{ij}\}$, find two set of routing variables $\boldsymbol{\phi} = \{\phi_{ij}\}$ and $\boldsymbol{\psi} = \{\psi_{ij}\}$ for a sensor network such that the network lifetime T_{net} is maximized.

Since maximizing the network lifetime T_{net} is equivalent to minimizing the maximum power consumption w_i for all $i \in N$, we can rewrite the DA-MLR problem formally as

$$\begin{aligned} \text{minimize} \quad & \max_{i \in N} w_i \\ \text{subject to} \quad & \phi_{ik} \geqslant 0, \ \sum_{k \in S_i} \phi_{ik} = 1, \quad \forall i, \\ & \psi_{ik} \geqslant 0, \ \sum_{k \in S_i} \psi_{ik} = 1, \quad \forall i. \end{aligned} \tag{4}$$

3. Recursive smoothing of DA-MLR problem

The max function in the DA-MLR problem (4) is non-linear and non-differentiable, so some distributed solutions based on the gradient methods are not directly applicable. There are many different approaches to overcome this difficulty. One is to transform the min–max problem to an equivalent optimization problem by introducing an extra upper bound parameter (e.g., [21]). This approach is exploited in a recent work [7] where subgradient algorithms are developed to solve the dual optimization problem. But the algorithms are shown to converge slowly. There is also a family of regularization approaches to obtain the smooth approximation for the max function in literature, such as the entropy type approximation [22,23], the two dimensional approximation [24] and the recursive approximation [25]. All these approaches are special cases of so-called *smoothing method*. An overview of these approaches can be found in [26]. A penalty-function based approximation is proposed in [27], which however lacks theoretical convergence property. In this section, we briefly introduce the recursive smoothing method [25] and adopt this method to construct a smoothing function for DA-MLR problem.

3.1. Recursive smoothing methods

A high dimensional max function with n variables can be expressed recursively as [25]

$$\max\{x_1, \ldots, x_n\} = \max\{\max\{x_1, \ldots, x_m\}, \max\{x_{m+1}, \ldots, x_n\}\}. \tag{5}$$

On the other hand, it is well-known that a two dimensional $\max\{x_1, x_2\}$ function can be approximated by the following function [24]

$$f(x;t) = tf(t^{-1}x) = \frac{\sqrt{(x_1 - x_2)^2 + t^2} + x_1 + x_2}{2}, \tag{6}$$

where t is an approximation parameter, and when t approaches 0, $f(x;t)$ approaches $\max\{x_1, x_2\}$. Based on this observation, a recursive procedure has been proposed to construct a high dimensional approximation for max function with more than two variables [25]. Specifically, a recursive function $f_{i,j}(x_i, \ldots, x_j)$, $1 \leqslant i < j \leqslant n$, and $k = j - i + 1$ is defined as

$$f_{i,j}(x_i, \ldots, x_j) = \begin{cases} f(x_i, x_j), & \text{if } k = 2, \\ f(f_{i,u_k}(x_i, \ldots, x_{u_k}), f_{l_k,j}(x_{l_k}, \ldots, x_j)), & \text{if } k > 2, \end{cases} \tag{7}$$

where

$$u_k = \mathrm{i} + \lceil k/2 \rceil - 1 \quad \text{and} \quad l_k = \begin{cases} u_k, & \text{if } k \text{ is odd}, \\ u_k + 1, & \text{if } k \text{ is even}. \end{cases}$$

The recursive procedure in (7) can be simplified as

$$f_{i,j}(x, t) = f(f_{i,u_k}(x^{(1)}), f_{l_k,j}(x^{(2)})), \quad 1 \leqslant i < j \leqslant n. \tag{8}$$

By defining

$$x^{(1)} = (x_i, \ldots, x_{u_k}) \quad \text{and} \quad x^{(2)} = (x_{l_k}, \ldots, x_j).$$

The partial derivative of $f_{i,j}(x, t)$ with respect to a variable x_l can be obtained recursively using the chain rule as

$$\begin{aligned} \frac{\partial f_{i,j}(x,t)}{\partial x_l} = & \frac{\partial f(f_{i,u_k}(x^{(1)},t),t)}{\partial f_{i,u_k}(x^{(1)},t)} \frac{\partial f_{i,u_k}(x^{(1)},t)}{\partial x_l} \\ & + \frac{\partial f(f_{l_k,j}(x^{(2)},t),t)}{\partial f_{l_k,j}(x^{(2)},t)} \frac{\partial f_{l_k,j}(x^{(2)},t)}{\partial x_l}. \end{aligned} \tag{9}$$

It is shown in [25] that the computational complexities of (8) and (9) are both $O(k)$.

3.2. Smoothing function for DA-MLR

The N dimensional max function of the DA-MLR problem in (4) can be approximated by the recursive function in (8) with $w_1, \ldots, w_N$ as inputs. That is, we can construct the following smoothing function for the max function of the DA-MLR problem

$$U(w,t) = f_{1,N}(w;t). \tag{10}$$

It is easy to see that $U(w,t)$ is an approximation of the N dimensional max function and converges to the max function as parameter t goes to 0, that is

$$\lim_{t\to 0} U(w,t) = \max_{i\in N} w_i. \tag{11}$$

Thus, instead of solving problem (4), we can solve the following approximate problem

$$\begin{aligned} \text{minimize} \quad & U(w,t) \\ \text{subject to} \quad & \phi_{ik} \geqslant 0, \ \sum_{k\in S_i} \phi_{ik} = 1, \quad \forall i, \\ & \psi_{ik} \geqslant 0, \ \sum_{k\in S_i} \psi_{ik} = 1, \quad \forall i. \end{aligned} \tag{12}$$

4. Optimality conditions

To solve problem (12) in a distributed manner, using $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ as the control variables, we extend the techniques in [28] to derive the necessary and sufficient conditions for achieving the optimality of the smoothing function $U(w,t)$.

By differentiating $U(w,t)$ with respect to ϕ_{ik} and ψ_{ik} for $i \in N$ and $k \in S_i$, we obtain

$$\frac{\partial U(w,t)}{\partial \phi_{ik}} = \sum_{l\in N} \frac{\partial f_{1,N}(w,t)}{\partial w_l} \frac{\partial w_l}{\partial \phi_{ik}}, \tag{13}$$

$$\frac{\partial U(w,t)}{\partial \psi_{ik}} = \sum_{l\in N} \frac{\partial f_{1,N}(w,t)}{\partial w_l} \frac{\partial w_l}{\partial \psi_{ik}}. \tag{14}$$

In (13) and (14), $\partial f_{1,N}(w,t)/\partial w_l$ can be computed recursively with (9) using $w_1, \ldots, w_N$ as inputs. The partial derivatives of w_l with respect to ϕ_{ik} and ψ_{ik} involve three nodes i, k and l. Note that if node l is not on the downstream paths of node i, both $\partial w_l/\partial \phi_{ik}$ and $\partial w_l/\partial \psi_{ik}$ are zeros because the traffic of node i does not pass through node l. Thus, we can narrow down the discussion to node $l \in N$ that is on the downstream path of node i.

In order to derive $\partial w_l/\partial \phi_{ik}$ and $\partial w_l/\partial \psi_{ik}$, we introduce a dummy variable r', which can be interpreted as the dummy traffic injected into node i that follows the set of routing of transit traffic λ_i, but without considering data aggregation. We consider three scenarios of source node i and node l as shown in Fig. 1.

(a) *Node i and l are non-adjacent*: If the source node i is not adjacent to node l as shown in Fig. 1a, let us consider a small increment ϵ to the input rate r'_i, this will cause an increment $\epsilon\phi_{ik}$ to the transit data of its nexthop neighbor k. This extra traffic is equivalent to an increment of $\epsilon\phi_{ik}$ to the input rate r'_k. Therefore, the contribution of the increment of r'_i to the power consumption of node l can be expressed via r'_k as $\epsilon\phi_{ik}\partial w_l/\partial r'_k$. This reasoning is applicable for all nexthop neighbors. Summing up over all $k \in S_i$ gives

$$\frac{\partial w_l}{\partial r'_i} = \sum_{k\in S_i} \phi_{ik} \frac{\partial w_l}{\partial r'_k}. \tag{15}$$

Suppose that the transit traffic λ_i of node i is fixed. An increment ϵ to the routing variable ϕ_{ik} will cause an increment $\epsilon\lambda_i$ to node k, which is equivalent to an increment of $\epsilon\lambda_i$ to the input rate r'_k. Therefore, we have

$$\frac{\partial w_l}{\partial \phi_{ik}} = \lambda_i \frac{\partial w_l}{\partial r'_k}. \tag{16}$$

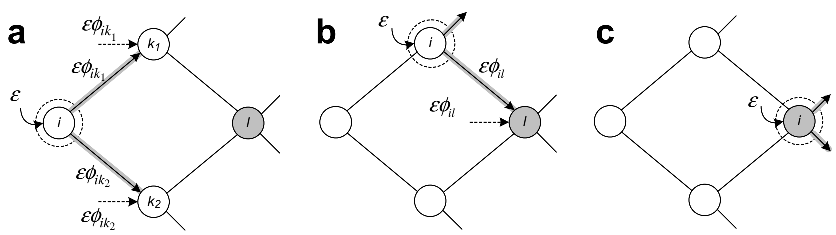

Fig. 1. Three scenarios of node i and l: (a) node i and node l are non-adjacent; (b) node i and node l are adjacent; (c) node i and node l are co-located.

Similarly, by fixing r_i and introducing an increment ϵ to the routing variable ψ_{ik} will cause an increment ϵr_i to node k, which is equivalent to an increment of $\epsilon r_i(1-q_{ik})$ to the input rate r'_k after the aggregation. Therefore,

$$\frac{\partial w_l}{\partial \psi_{ik}} = r_i(1-q_{ik})\frac{\partial w_l}{\partial r'_k}. \tag{17}$$

(b) *Node i and node l are adjacent*: If the source node i is adjacent to node l as shown in Fig. 1b, then the increment of power consumption of node l due to the increment of the input rate r'_i is composed of two parts. One is for receiving the increased traffic $\epsilon\phi_{il}$, which is given by $\epsilon\phi_{il}(e_r/E_l)$. The other is for transmitting the traffic $\epsilon\phi_{il}$, which is given by $\epsilon\partial w_l/\partial r'_l$ following the analysis of first scenario. Taking into account the indirect increment from other non-adjacent neighbor $k \neq l$ as derived above, we obtain

$$\begin{aligned}\frac{\partial w_l}{\partial r'_i} &= \sum_{k\in S_i, k\neq l}\phi_{ik}\frac{\partial w_l}{\partial r'_k} + \phi_{il}\left(\frac{e_r}{E_l}+\frac{\partial w_l}{\partial r'_l}\right)\\ &= \sum_{k\in S_i}\phi_{ik}\frac{\partial w_l}{\partial r'_k} + \frac{\phi_{il}e_r}{E_l}.\end{aligned} \tag{18}$$

Similarly, an increment ϵ to ϕ_{ik} leads to an increment of $\epsilon\lambda_i$ to node k, therefore

$$\frac{\partial w_l}{\partial \phi_{ik}} = \lambda_i\left(\frac{e_r}{E_l}+\frac{\partial w_l}{\partial r'_l}\right). \tag{19}$$

Also, an increment ϵ to ψ_{ik} leads to an increment of $\epsilon r_i(1-q_{ik})$ to node k, so

$$\frac{\partial w_l}{\partial \psi_{ik}} = r_i\left(\frac{e_r}{E_l}+(1-q_{il})\frac{\partial w_l}{\partial r'_l}\right). \tag{20}$$

(c) *Node i and node l are co-located*: If $i = l$, we can obtain from (1) and (2)

$$\begin{aligned}\frac{\partial w_i}{\partial r'_i} &= \sum_{k\in S_i}\frac{e_{ik}\phi_{ik}}{E_i},\\ \frac{\partial w_i}{\partial \phi_{ik}} &= \frac{\lambda_i e_{ik}}{E_i},\\ \frac{\partial w_i}{\partial \psi_{ik}} &= \frac{r_i e_{ik}}{E_i}.\end{aligned} \tag{21}$$

Combining the above results with (13) and (14), we obtain

$$\frac{\partial U(w,t)}{\partial \phi_{ik}} = \lambda_i\left(\sum_{l\in N}\frac{\partial f_{1,N}(w,t)}{\partial w_l}\frac{\partial w_l}{\partial r'_k} + \frac{\partial f_{1,N}(w,t)}{\partial w_i}\frac{e_{ik}}{E_i} + \frac{\partial f_{1,N}(w,t)}{\partial w_k}\frac{e_r}{E_k}\right) \tag{22}$$

and

$$\frac{\partial U(w,t)}{\partial \psi_{ik}} = r_i\left((1-q_{ik})\sum_{l\in N}\frac{\partial f_{1,N}(w,t)}{\partial w_l}\frac{\partial w_l}{\partial r'_k} + \frac{\partial f_{1,N}(w,t)}{\partial w_i}\frac{e_{ik}}{E_i} + \frac{\partial f_{1,N}(w,t)}{\partial w_k}\frac{e_r}{E_k}\right). \tag{23}$$

Applying the Lagrange multipliers v_i and μ_i to the constraints $\sum_{k\in S_i}\phi_{ik} = 1$ and $\sum_{k\in S_i}\psi_{ik} = 1$, respectively, and taking into account the constraints $\phi \succeq 0$ and $\psi \succeq 0$, the necessary condition for $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ to be the minimizer of $U(w,t)$ is given by the following theorem:

Theorem 1 (Necessary condition). *Let* $\partial U(w,t)/\partial\phi_{ik}$ *and* $\partial U(w,t)/\partial\psi_{ik}$ *be given by* (22) *and* (23), *respectively, the necessary conditions for the existence of the minimum* $U(w,t)$ *with respect to* ϕ^* *and* ψ^* *for all* $i \in N$ *are*

$$\frac{\partial U(w,t)}{\partial \phi^*_{ik}} = \begin{cases} = v_i, & \phi^*_{ik} > 0,\\ \geqslant v_i, & \phi^*_{ik} = 0\end{cases} \tag{24}$$

and

$$\frac{\partial U(w,t)}{\partial \psi^*_{ik}} = \begin{cases} = \mu_i, & \psi^*_{ik} > 0,\\ \geqslant \mu_i, & \psi^*_{ik} = 0.\end{cases} \tag{25}$$

This states that all links (i,k) for which $\phi_{ik} > 0$ must have the same value of $\partial U(w,t)/\partial\phi_{ik}$, and this value must be less than or equal to the value of $\partial U(w,t)/\partial\phi_{ik}$ for the links on which $\phi_{ik} = 0$. The same argument is held for ψ_{ik}.

However, the conditions given by (24) and (25) are not sufficient to minimize $U(w,t)$ because these conditions are automatically satisfied if r_i and λ_i are zeros for some node i. Since λ_i and r_i are decoupled, we show next that (24) and (25) would be sufficient to minimize $U(w,t)$ if the factors λ_i and r_i were removed from the conditions.

Let us define

$$\frac{\partial U(w,t)}{\partial r'_k} = \sum_{l\in N}\frac{\partial f_{1,N}(w,t)}{\partial w_l}\frac{\partial w_l}{\partial r'_k} \tag{26}$$

and

$$Z_{ik} = \frac{\partial f_{1,N}(w,t)}{\partial w_i}\frac{e_{ik}}{E_i} + \frac{\partial f_{1,N}(w,t)}{\partial w_k}\frac{e_r}{E_k}. \tag{27}$$

Theorem 2 (Sufficient condition). *Let $\partial U(w,t)/\partial\phi_{ik}$ and $\partial U(w,t)/\partial\psi_{ik}$ be given by (22) and (23), respectively. The sufficient conditions for ϕ_{ik} and ψ_{ik} to be the minimizer of $U(w,t)$ are*

$$\frac{\partial U(w,t)}{\partial r'_k} + Z_{ik} \geqslant \frac{\partial U(w,t)}{\partial r'_i} \tag{28}$$

and

$$(1-q_{ik})\frac{\partial U(w,t)}{\partial r'_k} + Z_{ik} \geqslant \frac{\partial U(w,t)}{\partial r'_i}, \tag{29}$$

respectively, where the equality is achieved for k whose routing variables ϕ_{ik} and ψ_{ik} are greater than 0.

In other words, when the optimality is achieved, the transit traffic is distributed only over those links with the smallest and identical values of $\partial U(w,t)/\partial r'_k + Z_{ik}$, while the raw data is distributed only over those links with the smallest values of $(1-q_{ik})\partial U(w,t)/\partial r'_k + Z_{ik}$. The proofs of the necessary and sufficient conditions are provided in Appendices A and B, respectively.

5. Distributed algorithm and protocol

In this section, we first design a gradient algorithm for nodes to update their routing variables according to the sufficient conditions. We then discuss the protocol that nodes can exchange information and execute the gradient algorithm to maximize the network lifetime.

5.1. Gradient algorithm

The gradient algorithm is based on the sufficient conditions given by (28) and (29). Every node executes the algorithm to update its routing variables ϕ and ψ iteratively until these conditions are satisfied. The algorithm is operated in the following steps.

1. Calculate $\partial U(w,t)/\partial r'_k$ and Z_{ik} for every neighbor $k \in S_i$. Find the neighbors k_1 and k_2 such that

$$k_1 \leftarrow \arg\min_{k\in S_i}\left\{\frac{\partial U(w,t)}{\partial r'_k} + Z_{ik}\right\} \tag{30}$$

   and

$$k_2 \leftarrow \arg\min_{k\in S_i}\left\{(1-q_{ik})\frac{\partial U(w,t)}{\partial r'_k} + Z_{ik}\right\}. \tag{31}$$

2. Calculate the amounts of reduction $\Delta\phi_{ik}$ and $\Delta\psi_{ik}$. Define

$$a_{ik} = \frac{\partial U(w,t)}{\partial r'_k} + Z_{ik} - \left\{\frac{\partial U(w,t)}{\partial r'_{k_1}} + Z_{ik_1}\right\} \tag{32}$$

   and

$$b_{ik} = (1-q_{ik})\frac{\partial U(w,t)}{\partial r'_k} + Z_{ik} - \left\{(1-q_{ik_2})\frac{\partial U(w,t)}{\partial r'_{k_2}} + Z_{ik_2}\right\}. \tag{33}$$

   The amounts of reduction to ϕ_{ik} and ψ_{ik} are given respectively by

$$\Delta\phi_{ik} = \min\{\phi_{ik}, \gamma a_{ik}/\lambda_i\}, \quad k \in S_i \tag{34}$$

   and

$$\Delta\psi_{ik} = \min\{\psi_{ik}, \eta b_{ik}/r_i\}, \quad k \in S_i, \tag{35}$$

   where γ and η are positive scale parameters.
3. Update routing variables as follows

$$\phi_{ik} = \begin{cases} \phi_{ik} - \Delta\phi_{ik}, & k \neq k_1, \\ \phi_{ik} + \sum_{k\in S_i, k\neq k_1}\Delta\phi_{ik}, & k = k_1 \end{cases} \tag{36}$$

   and

$$\psi_{ik} = \begin{cases} \psi_{ik} - \Delta\psi_{ik}, & k \neq k_2, \\ \psi_{ik} + \sum_{k\in S_i, k\neq k_2}\Delta\psi_{ik}, & k = k_2. \end{cases} \tag{37}$$

Using this algorithm, each node i gradually decreases the routing variables for which the values $\partial U(w,c)/\partial r'_k + Z_{ik}$ and $(1-q_{ik})\partial U(w,c)/\partial r'_k + Z_{ik}$ are larger, and increases the routing variables for which the values are the smallest until the sufficient conditions (28) and (29) are satisfied.

5.2. Protocol

Let M_i denote the set of downstream nodes of node i. A table is maintained by node i for all nodes in M_i, where each entry of the table consists of the node identity l, the power consumption w_l and the power consumption rate $\partial w_l/\partial r'_i$.

In each iteration, the DA-MLR protocol is operated as follows by each node i:

1. Wait until receiving the table from all of its downstream neighbors and merge M_k of neighbor k with the local set M_i.
2. Calculate the new routing variables using the gradient algorithm.

3. Calculate the power consumption w_i and power consumption rate $\partial w_i / r'_i$, and add a new entry into the table with the local information.
4. Update the power consumption rate $\partial w_l / \partial r'_i$ for all $l \in M_i$ in the table using the recursive equation (15).
5. Pass the table to upstream neighbors.

Each iteration of the DA-MLR algorithm only involves the communications between neighboring nodes, and the communication cost is bounded by the number of downstream nodes. Thus, every node can update the routing variables in a distributive manner. This procedure is repeated until the sufficient conditions are satisfied and the global optimality is achieved. The convergence property of the algorithm is shown in Section 6.2 through simulations.

6. Performance evaluation

6.1. Simulation setup

We compare the performance of DA-MLR algorithm with the Minimum Energy Gathering Algorithm (MEGA) [14] and the Minimum Energy Routing (MER) algorithms.

1. MEGA – This algorithm tries to optimize the aggregation costs for raw data and the transmission costs for compressed data. It maintains two trees – the coding tree and the *shorted path tree* (SPT). The coding tree is constructed with *directed minimum spanning tree* (DMST) algorithm for data aggregation, and the SPT is for delivering the compressed data.
2. MER – This algorithm tries to minimize the overall energy consumption of delivery of a packet using the shortest path from the source node to the sink node in term of energy cost. For fair comparison, we revise the MER algorithm to take into account the data aggregation. That is, raw data is firstly compressed at the next-hop node along the shortest path. After that, the compressed data is delivered through the shortest path.

The network size in the simulation varies between 20 and 80. For each network size, 20 random network topologies are generated and average results are obtained for these algorithms. Sensor nodes are randomly distributed on a $100\,\text{m} \times 100\,\text{m}$ region. The transmission radius of all nodes is $R = 20$ m. For radio power consumption setting, we adopt the *first order radio model* and set $\epsilon_{\text{elec}} = 50$ nJ/bit, $\epsilon_{\text{amp}} = 100$ pJ/bit/m^2 and path loss exponent $n = 2$. For data correlation setting, we adopt the *Gaussian random field* model [10] such that the correlation coefficient q_{ik} decreases exponentially with the increase of the distance between nodes, or $q_{ik} = \exp(-\alpha d_{ik}^2)$. Here α is the correlation parameter ranging between $\alpha = 0.001/\text{m}^2$ (high correlation) and $\alpha = 0.01/\text{m}^2$ (low correlation) in the simulations. All nodes have the same battery energy $E_i = 1$ kJ and constant data rate $r_i = 1$ kbps. The decreasing sequences of step size γ, η and the approximation parameter t are used for the DA-MLR algorithm in the experiments.

6.2. Simulation results

In Fig. 2 we show the network lifetime obtained by these algorithms under two data correlation settings ($\alpha = 0.001$ and $\alpha = 0.01$). DA-MLR algorithm can almost double the network lifetime compare to MEGA and MER algorithms. The network lifetimes obtained by DA-MLR algorithm increase gradually as the network size grows, while the lifetimes obtained by MEGA and MER algorithms drop continuously. This can be explained as follows. The overall raw data rate is proportional to the number of nodes in the network. Thus, it is expected that the network lifetime should decrease as the network size grows and more data traffic is generated. On the other hand, the increase of nodes in the network also drives the network topology from sparse to dense, which affects the network in two ways. First, the distance between neighboring nodes

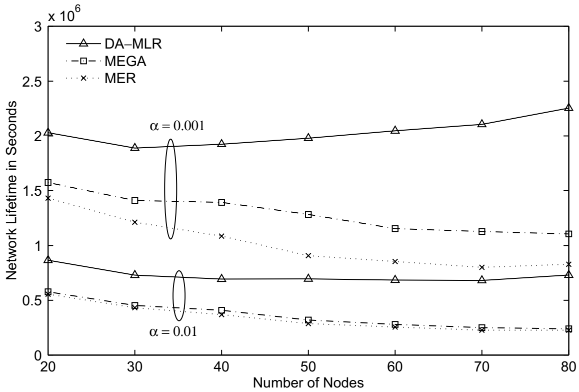

Fig. 2. Network lifetime.

becomes smaller, so a node needs less power to send data to its neighbors. Second, the data correlation between neighboring nodes becomes higher, so more redundant information can be removed through data aggregation. Both effects help to reduce the energy consumption per node. From the simulation, we can see that MEGA and MER algorithms do not exploit this feature and the network lifetime drops continuously as the network size grows. In particular, under lower correlation condition ($\alpha = 0.01$), the network lifetimes returned by both algorithms are very close. However, MEGA outperforms MER algorithm under higher correlation condition ($\alpha = 0.001$) because it can optimize the data aggregation, but MER algorithm cannot. DA-MLR algorithm, on the other hand, can optimize both routing and data aggregation, therefore it performs much better than MEGA and MER algorithms. For example, for the network size with 80 nodes, the network lifetime obtained by DA-MLR algorithm is around two times of MEGA algorithm and three times of MER algorithm for $\alpha = 0.001$. For $\alpha = 0.01$, the network lifetime of DA-MLR algorithm is around three times of both MEGA and MER algorithms.

The aggregated data rate at the sink node is shown in Fig. 3. We can see that DA-MLR algorithm has better aggregation results than MER algorithm. For MEGA algorithm, its aggregated rate is comparable to DA-MLR algorithm under higher correlation condition ($\alpha = 0.001$), but is still worse than DA-MLR algorithm under lower correlation condition. Comparing to the results in Fig. 2, we can see that MEGA algorithm helps to optimize data aggregation, but fails to balance the traffic across the network since it uses the shortest path to deliver compressed data. Therefore, under lower correlation condition where no much data can be compressed, the network lifetimes of MEGA and MER algorithms are quite close.

In Fig. 4 we show the average network lifetimes given by DA-MLR, MEGA and MER algorithms as the correlation parameter α increases from 0.001 to 0.01. We can see MEGA and MER algorithms achieve better network lifetime for the smaller network size (40 nodes) than the larger network size (80 nodes) under all correlation situations. For the same network size, the performance of MEGA and MER algorithms degenerates as the correlation becomes smaller. This coincides with the observation in Fig. 2. On the other hand, the network lifetime of DA-MLR algorithm is higher in larger

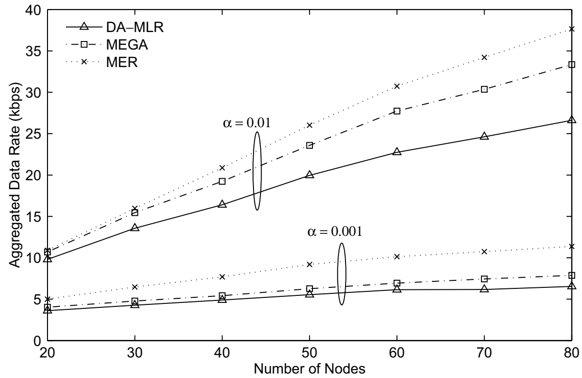

Fig. 3. Aggregated data rate.

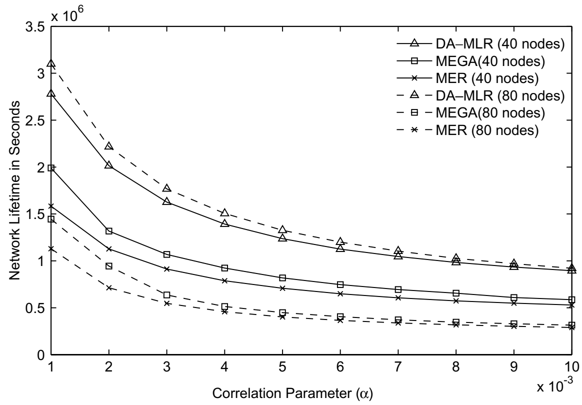

Fig. 4. Network lifetime.

network. But the difference diminishes as the correlation is decreasing. Under the same settings, we show in Fig. 5 the aggregated data rate at the sink node with these algorithms. We see that DA-MLR algorithm effectively reduces the network traffic compare to MEGA and MER algorithms.

We study the impact of the number of source nodes on the performance by fixing the network size to 80 nodes and varying the number of source nodes. We assume that data aggregation is performed only when two source nodes are neighbors. In Fig. 6 we show the network lifetime for various number of source nodes. We can see that under the low correlation case ($\alpha = 0.01$, 0.005), the network lifetime drops with the increase of source nodes. However, it is interesting to see that under high correlation case ($\alpha = 0.01$, 0.005), the network lifetime stops decreasing after the source nodes reaches certain number. This is because when the number of source nodes increases, the chance of a source node to find a neighboring source node

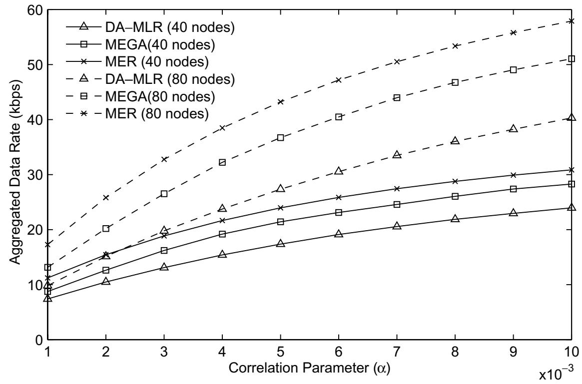

Fig. 5. Aggregated data rate.

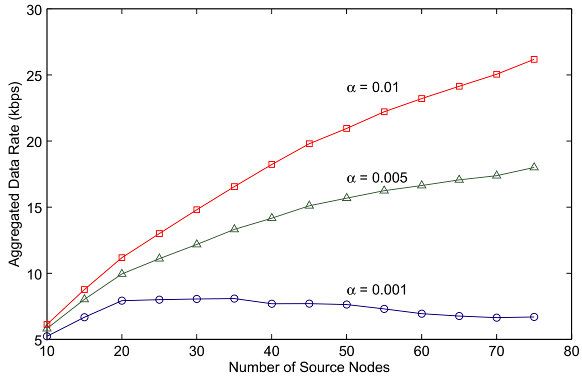

Fig. 7. Aggregated data rate.

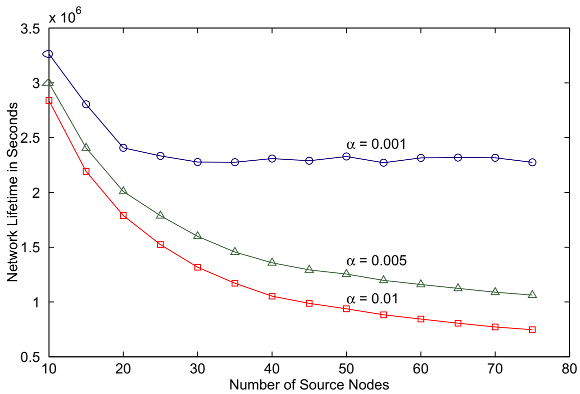

Fig. 6. Network lifetime.

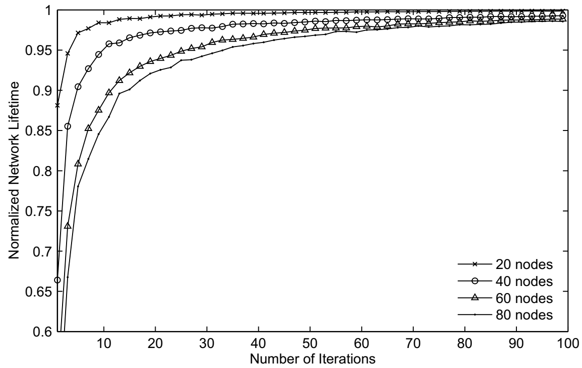

Fig. 8. Normalized network lifetime.

increases accordingly. Therefore, they can take advantage of data aggregation to reduce data traffic, which can partially cancel out the effect of traffic increase due to the increase of source nodes. Similar effects can be observed in Fig. 7, which shows the aggregated data rate at the sink node. It is interesting to see that for the lower correlation case, the aggregated data rate is simply increased as the source nodes grows. However, for the high correlation case, the data rate stops increasing after it reaches the peak value where the source nodes reach certain number. This coincides with the observation in Fig. 6.

The convergence property is another important performance metric for DA-MLR algorithm. Fig. 8 shows the normalized network lifetime obtained by DA-MLR algorithm for various network sizes (20, 40, 60 and 80 nodes). The network lifetime is computed at each iteration and normalized with respect to the optimal value. We see that the algorithm can converge efficiently. The number of iterations required for the network lifetime to converge to over 95% of the optimal values is 5, 10, 25 and 30 iterations, respectively for network size ranging from 20 to 80 nodes. The effectiveness of the distributed DA-MLR algorithm can also be observed from Fig. 9 which shows the normalized

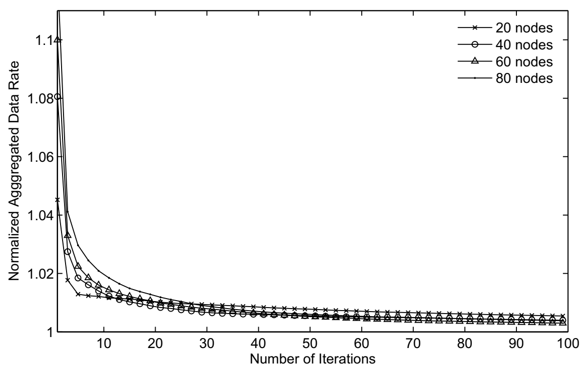

Fig. 9. Normalized aggregated data rate.

aggregated data rate at the sink node for various network sizes. The aggregated data rate is normalized with respect to the optimal value. Again, we see that the distributed DA-MLR algorithm successfully reduces the data rate.

7. Conclusions

In this paper we present the data-aggregated maximum lifetime routing for wireless sensor networks. Network lifetime is maximized by jointly optimizing routing and data aggregation variables. A recursive smoothing function is adopted to approximate the original optimization problem. We derive the necessary and sufficient conditions for the smoothing problem and design a distributed algorithm as the solution. Simulation results demonstrate that the proposed scheme can significantly reduce the traffic and improve the network lifetime.

Appendix A. Proof of necessary conditions

Proof. We now prove that (24) and (25) are the necessary conditions to minimize $U(w,t)$ by defining the following Lagrange function

$$U(w,t,\mu,v,\kappa,\chi) = U(w,t) + \sum_{i\in N}\mu_i\left(1-\sum_{k\in S_i}\phi_{ik}\right) - \sum_{i\in N,k\in S_i}\kappa_{ik}\phi_{ik} + \sum_{i\in N}v_i\left(1-\sum_{k\in S_i}\psi_{ik}\right) - \sum_{i\in N,k\in S_i}\chi_{ik}\psi_{ik}, \tag{A.1}$$

where $\mu=(\mu_1,\ldots,\mu_N)$ and $v=(v_1,\ldots,v_N)$ are the Lagrange multipliers.

According to Kuhn–Tucker theorem, the necessary condition for a ϕ^* to be a minimizer of $U(w,c,\mu,v,\kappa,\chi)$ is that there exist Lagrange multipliers μ_i^* and $\kappa_{ik}^*, i\in N, k\in S_i$ such that

$$\begin{aligned}&\frac{\partial U}{\partial\phi_{ik}^*} - \mu_i^* - \kappa_{ik}^* = 0,\\ &\kappa_{ik}^* = 0,\ \text{if}\,\phi_{ik}^* > 0,\quad \forall i,k,\\ &\kappa_{ik}^* > 0,\ \text{if}\,\phi_{ik}^* = 0,\quad \forall i,k.\end{aligned} \tag{A.2}$$

Rearranging the first equation as $\partial U/\partial\phi_{ik}^* = v_i^* + \kappa_{ik}^*$, and taking into account the second and the third conditions will complete the proof of (24). Similarly, we can prove that (25) is the necessary condition for ψ^* to be a minimizer of $U(w,c,\mu,v,\kappa,\chi)$. □

Appendix B. Proof of sufficient conditions

Proof. To prove that (28) and (29) are sufficient conditions to minimize $U(w,t)$, let us assume that there are routing variables ϕ^* and ψ^* satisfying (28) and (29), respectively. Let the corresponding node flow be λ^*+r^* and link flow be x^*, where $x_{ik} = \lambda_i\phi_{ik} + r_i\psi_{ik}$. Let ϕ and ψ be any other set of routing variables with the corresponding node flow $\lambda + r$ and link flow x. Define $x(\theta)$ as the convex combination of x^* and x with respect to a variable θ, that is,

$$x_{ik}(\theta) = (1-\theta)x_{ik}^* + \theta x_{ik}. \tag{B.1}$$

Therefore, each w_l can be represented by the link flow $x(\theta)$. Since each $w_l(\theta)$ is a convex function of the link flow x, therefore $U(\theta)$ is also a convex function of θ, so it is obvious

$$\left.\frac{\mathrm{d}U(\theta)}{\mathrm{d}\theta}\right|_{\theta=0} \leqslant U(\phi,\psi) - U(\phi^*,\psi^*). \tag{B.2}$$

Since ϕ and ψ are arbitrary routing variables, it will complete the proof by proving that $\mathrm{d}U(\theta)/\mathrm{d}\theta \geqslant 0$ at $\theta = 0$.

From (2) and (B.1), it is straightforward to express w_l as a function of the link flow $x(\theta)$ as

$$w_l(\theta) = \frac{1}{E_l}\left(e_s + e_g r_l + \sum_{i\in A_l} x_{il}(\theta)e_r + \sum_{k\in S_l} x_{lk}(\theta)e_{lk}\right). \tag{B.3}$$

Differentiating w_l with respect to θ, and from (B.1) and (B.3) we obtain

$$\frac{\partial w_l}{\partial\theta} = \sum_{i\in A_l}\frac{e_r}{E_l}(x_{il} - x_{il}^*) + \sum_{k\in S_l}\frac{e_{lk}}{E_l}(x_{lk} - x_{lk}^*). \tag{B.4}$$

We can obtain $\mathrm{d}U(\theta)/\mathrm{d}\theta$ as

$$\begin{aligned}\left.\frac{\mathrm{d}U(\theta)}{\mathrm{d}\theta}\right|_{\theta=0} &= \sum_{l\in N}\frac{\partial f_{1,N}(w,t)}{\partial w_l}\frac{\partial w_l}{\partial\theta}\\ &= \sum_{l\in N}\frac{\partial f_{1,N}(w,t)}{\partial w_l}\\ &\quad\cdot\left\{\sum_{i\in A_l}\frac{e_r}{E_l}(x_{il} - x_{il}^*) + \sum_{k\in S_l}\frac{e_{lk}}{E_l}(x_{lk} - x_{lk}^*)\right\}.\end{aligned} \tag{B.5}$$

We then first prove that

$$\sum_{l\in N}\frac{\partial f_{1,N}(w,t)}{\partial w_l}\cdot\left(\sum_{i\in A_l}\frac{e_r x_{il}}{E_l}+\sum_{k\in S_l}\frac{e_{lk}x_{lk}}{E_l}\right) \geqslant \sum_{i\in N} r_i\frac{\partial U(w,t)}{\partial r_i'}. \tag{B.6}$$

Multiplying both sides of (28) with λ_i and ϕ_{ik}, and multiplying both sides of (29) with r_i and ψ_{ik}, summing over all $i\in N$ and $k\in S_i$. Using the fact that $\lambda_i=\sum_{j\in A_i}[\lambda_j\phi_{ji}+r_j(1-q_{ji})\psi_{ji}]$, we can obtain the result for the left-hand side as

$$\mathrm{LHS}=\sum_{i\in N}\sum_{k\in S_i}[\lambda_i\phi_{ik}+r_i\psi_{ik}(1-q_{ik})] \times\frac{\partial U(w,t)}{\partial r_k'}+\sum_{i\in N}\sum_{k\in S_i}(\lambda_i\phi_{ik}+r_i\psi_{ik})Z_{ik} \tag{B.7}$$

and the right-hand side as

$$\begin{aligned}\mathrm{RHS}&=\sum_{i\in N}(\lambda_i+r_i)\frac{\partial U(w,t)}{\partial r_i'}\\&=\sum_{i\in N}\sum_{j\in A_i}[\lambda_j\phi_{ji}+r_j(1-q_{ji})\psi_{ji}]\\&\quad\times\frac{\partial U(w,t)}{\partial r_i'}+\sum_{i\in N}r_i\frac{\partial U(w,t)}{\partial r_i'}.\end{aligned} \tag{B.8}$$

Notice that the first term of LHS and RHS are equivalent and can be canceled out. Substituting Z_{ik} from (27) into LHS and recalling the inequality between (B.7) and (B.8), we obtain

$$\sum_{l\in N}\frac{\partial f_{1,N}(w,t)}{\partial w_l}\left(\sum_{i\in A_l}\frac{e_r}{E_l}(\lambda_i\phi_{il}+r_i\psi_{il})+\sum_{k\in S_l}\frac{e_{lk}}{E_l}(\lambda_l\phi_{lk}+r_l\psi_{lk})\right) \geqslant \sum_{i\in N} r_i\frac{\partial U(w,t)}{\partial r_i'}. \tag{B.9}$$

Recalling that $x_{il}=\lambda_i\phi_{il}+r_i\psi_{il}$, substituting this into (B.9) we can obtain (B.6).

Following the same derivation procedure, if λ^*, r^*, ϕ^* and ψ^* are substituted for λ, r, ϕ and ψ, this becomes an equality from the equations for $\partial U/\partial r_i'$ in (28) and (29). That is,

$$\sum_{l\in N}\frac{\partial f_{1,N}(w,t)}{\partial w_l}\left(\sum_{i\in A_l}\frac{e_r x^*_{il}}{E_l}+\sum_{k\in S_l}\frac{e_{lk}x^*_{lk}}{E_l}\right) = \sum_{i\in N} r_i\frac{\partial U(w,t)}{\partial r_i'}. \tag{B.10}$$

Substituting (B.6) and (B.10) into (B.5), we see that $\mathrm{d}U(\theta)/\mathrm{d}\theta\geqslant 0$ at $\theta=0$, which complete the proof. □

References

[1] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy-efficient communication protocol for wireless microsensor networks, in: Proc. International Conference on System Sciences, 2000.

[2] S. Singh, M. Woo, C.S. Raghavendra, Power-aware routing in mobile ad hoc networks, in: MobiCom'98, 1998, pp. 181–190.

[3] V. Rodoplu, T.H. Meng, Minimum energy mobile wireless networks, IEEE Journal on Selected Areas in Communications 17 (8) (1999) 1333–1344.

[4] J.-H. Chang, L. Tassiulas, Routing for maximum system lifetime in wireless ad-hoc networks, in: Proc. of Allerton Conference on Communication, Control and Computing, 1999.

[5] J.-H. Chang, L. Tassiulas, Energy conserving routing in wireless ad-hoc networks, in: Infocom'00, 2000, pp. 22–31.

[6] A. Sankar, Z. Liu, Maximum lifetime routing in wireless ad-hoc networks, in: Infocom'04, 2004.

[7] R. Madan, S. Lall, Distributed algorithms for maximum lifetime routing in wireless sensor networks, in: Global Telecommunications Conference (GLOBECOM'04), IEEE, vol. 2, 2004.

[8] K. Kalpakis, K. Dasgupta, P. Namjoshi, Maximum lifetime data gathering and aggregation in wireless sensor networks, in: Proc of ICN'02, 2002.

[9] S. Pattem, B. Krishnamachari, R. Govindan, The impact of spatial correlation on routing with compression in wireless sensor networks, in: IPSN'04, ACM Press, Berkeley, CA, USA, 2004, pp. 28–35.

[10] R. Cristescu, B. Beferull-Lozano, M. Vetterli, On network correlated data gathering, in: Infocom'04, Hong Kong, 2004.

[11] M. Gastpar, M. Vetterli, Source-channel communication in sensor networks, in: IPSN'03, 2003.

[12] A. Scaglione, S.D. Servetto, On the interdependence of routing and data compression in multi-hop sensor networks, in: MobiCom'02, ACM Press, Atlanta, Georgia, USA, 2002, pp. 140–147.

[13] M.C. Vuran, O.B. Akan, I.F. Akyildiz, Spatio-temporal correlation: theory and applications for wireless sensor networks, Computer Networks 45 (3) (2004) 245.

[14] P. von Rickenbach, R. Wattenhofer, Gathering correlated data in sensor networks, in: DIALM-POMC'04: Proceedings of the 2004 Joint Workshop on Foundations of Mobile Computing, ACM Press, New York, NY, USA, 2004, pp. 60–66.

[15] M. Mauve, J. Widmer, H. Hartenstein, A survey on position-based routing in mobile ad hoc networks, IEEE Network Magazine 15 (6) (2001) 30–39.

[16] L. Doherty, L.E. Ghaoui, K.S.J. Pister, Convex position estimation in wireless sensor networks, in: Infocom, 2001.

[17] Y. Shang, W. Ruml, Y. Zhang, M.P.J. Fromherz, Localization from mere connectivity, in: MobiCom'03, 2003.

[18] F. Kuhn, R. Wattenhofer, A. Zollinger, Asymptotically optimal geometric mobile ad-hoc routing, in: Proc. of DIALM'99, 2002, pp. 24–33.

[19] P. Bose, P. Morin, I. Stojmenovi, J. Urrutia, Routing with guaranteed delivery in ad hoc wireless networks, in: Proc. of DIALM'99, 1999, pp. 48–55.

[20] B. Karp, H. Kung, GPSR: greedy perimeter stateless routing for wireless networks, in: MobiCom'00, 2000, pp. 243–254.

[21] R.K. Ahuja, Algorithms for the minimax transportation problem, Naval Research Logistics 33 (1986) 725–740.
[22] A. Ben-Tal, M. Teboulle, A smoothing technique for nondifferentiable optimization problems, in: Proceedings of the International Seminar on Optimization, Springer-Verlag New York Inc., New York, NY, USA, 1988, pp. 1–11.
[23] X. Li, An entropy-based aggregate method for minimax optimization, Engineering Optimization 18 (1992) 277–285.
[24] C. Chen, O.L. Mangasarian, Smoothing methods for convex inequalities and linear complementarity problems, Mathematical Programming 71 (1) (1995) 51–69.
[25] S.I. Birbil, S.-C. Fang, J. Frenk, S. Zhang, Recursive approximation of the high dimensional max function, Operations Research Letters 33 (2005) 450–458.
[26] L. Qi, D. Sun, Smoothing functions and a smoothing Newton method for complementarity and variational inequality problems, Journal of Optimization Theory and Applications 113 (2002) 121–147.
[27] C. Hua, T.-S.P. Yum, Maximum lifetime routing and data aggregation for wireless sensor networks, in: IFIP Networking 2006, Coimbra, Portugal, 2006.
[28] R.G. Gallager, A minimum delay routing algorithm using distributed computation, IEEE Transaction on Communications 25 (1) (1977) 73–85.

Cunqing Hua (S'01) received the B.S. degree in Electronics Engineering from University of Science and Technology of China in 2000, M.Phil. and Ph.D. degree in Information Engineering from The Chinese University of Hong Kong in 2002 and 2006 respectively. His general research interests include protocol and algorithm design, analysis and optimization, and performance evaluation for computer networks, wireless mobile and sensor networks.

Tak-Shing Peter Yum (S'76-A'78-SM'86) was born in Shanghai. He received the B.S., M.S., and Ph.D. degree from Columbia University, New York, NY in 1974, 1975 and 1978, respectively.

He joined Bell Telephone Laboratories in April 1978 working on switching and signaling systems. In 1980, he accepted a teaching appointment at the National Chiao Tung University, Taiwan. Then, in 1982, he joined the Chinese University of Hong Kong, where he is now Dean of Engineering and Professor of Information Engineering. He has published widely in Internet research with contributions to routing, buffer management, deadlock handling, message resequencing and multiaccess protocols. He then branched out to work on cellular network, lightwave networks, video distribution networks and 3G networks. His recently interest is in RFID and sensor network technologies and applications.