Optimal Routing and Data Aggregation for Maximizing Lifetime of Wireless Sensor Networks

Cunqing Hua and Tak-Shing Peter Yum, *Senior Member, IEEE*

Abstract—An optimal routing and data aggregation scheme for wireless sensor networks is proposed in this paper. The objective is to maximize the network lifetime by jointly optimizing data aggregation and routing. We adopt a model to integrate data aggregation with the underlying routing scheme and present a smoothing approximation function for the optimization problem. The necessary and sufficient conditions for achieving the optimality are derived and a distributed gradient algorithm is designed accordingly. We show that the proposed scheme can significantly reduce the data traffic and improve the network lifetime. The distributed algorithm can converge to the optimal value efficiently under all network configurations.

Index Terms—Data aggregation, maximum lifetime routing, network lifetime, smoothing methods, wireless sensor networks.

I. INTRODUCTION

THE main operation of a wireless sensor network (WSN) is to monitor the physical environment, process the sensed information, and deliver the results to some specific sink nodes. Sensor nodes are normally powered by batteries with limited energy resource. Therefore, the primary challenge for this energy-constrained system is to design energy-efficient protocols to maximize the lifetime of the network. Since radio transmission is the primary source of power consumption [1], the design of communication protocols for topology management, transmission power control, and energy-efficient routing has been the focus of many studies [2]–[8].

Among these schemes, energy-efficient routing [6]–[8] is one of the well-studied approaches for both wireless ad hoc networks and sensor networks. The basic idea is to route the packet through the minimum energy paths so as to minimize the overall energy consumption for delivering the packet from the source to the destination. The drawback of this approach is that it tends to overwhelm the nodes on the minimum energy path, which is undesirable for sensor networks since all sensor nodes are collaborating for a common mission and the duties of failed nodes may not be taken by other nodes.

A few schemes have been proposed to address this problem by studying the maximum lifetime routing problem [9]–[12]. The problem focuses on computing the flow and transmission power to maximize the lifetime of the network, which is the time at which the first node in the network runs out of energy. Some distributed solutions based on subgradient algorithms [11] and utility-based algorithm [13] have been proposed. The common assumption of these works is that the data flows are conserved during the transmission from the sensor nodes to the sink node, which however is not true for sensor networks because data collected by neighboring nodes are often spatially correlated. Therefore, redundant information can be removed through data aggregation at the intermediate nodes.

Some research efforts have been made to exploit the data correlation feature to improve the performance of the communication protocols. In [14], Kalpakis *et al.* study the maximum lifetime data aggregation (MLDA) problem. The objective is to find a set of data gathering schedules to maximize the system lifetime—a schedule is defined as a collection of directed spanning trees rooted at the sink node. In [15], the impact of the data correlation on the routing schemes is studied and a static clustering scheme is proposed that achieves a near-optimal performance for various spatial correlations. Two complementary data aggregation approaches are proposed in [16]. One is to perform blind data compression at the source nodes using Slepian–Wolf Coding, the other is to aggregate data using the explicit side information from other nodes. In [17], the authors propose a Minimum Energy Gathering Algorithm (MEGA). The algorithm requires to maintain two trees—the coding tree for raw data aggregation and the shortest path tree (SPT) for delivering the compressed data to the sink node. These works demonstrate that data aggregation can greatly improve the performance of various communication protocols.

However, none of the existing works have considered the integration of data aggregation and maximum lifetime routing. By jointly optimizing routing and data aggregation, the network lifetime can be extended from two dimensions. One is to reduce the traffic across the network by data aggregation, which can reduce the power consumption of the nodes close to the sink node. The other is to balance the traffic to avoid overwhelming the bottleneck nodes. In this paper, we present a model to integrate routing and data aggregation. We adopt the geometric routing [18] whereby the routing is determined solely according to the nodal position. This allows different data correlation models such as that in [17] to be incorporated without intervening the underlying routing scheme. The problem there-

The authors are with Information Engineering Department, Chinese University of Hong Kong, Shatin, N.T., Hong Kong (e-mail: chua0@ie.cuhk.edu.hk; yum@ie.cuhk.edu.hk).

fore is focused on computing the optimal routing variables that maximize the network lifetime. Since the maximum lifetime problem cannot be solved directly using the simple distributed methods, we propose a smoothing function to approximate the original max function by exploiting the special structure of the network. We derive the necessary and sufficient conditions for achieving the optimality of the smoothing function and design a distributed gradient algorithm accordingly. We conduct extensive simulations to show that the proposed scheme can significantly reduce the data traffic and improve the network lifetime. The distributed algorithm can converge to the optimal values efficiently under all network configurations.

In the next section, we first present the system models and define the maximum lifetime routing problem. In Section III, we propose a smoothing function to approximate the maximum lifetime routing problem and derive the optimality conditions. The implementation issues of the distributed algorithm are discussed in Section IV. Performance evaluation is presented in Section V and finally we conclude this paper in Section VI.

II. SYSTEM MODELS

We model the topology of a wireless sensor network as a undirected graph $G(N, A)$, where N is the set of nodes, and A is the set of undirected links. A sink node $t \in N$ is responsible for collecting data from all other nodes. To capture the characteristics of this sensor network, we present the routing model, the data aggregation model, and the power consumption model in the following subsections.

A. Routing Model

The routing algorithm suitable for use belongs to the class of *geometric routing* algorithms [18]. Every sensor node is assumed to know its own position as well as that of its neighbors, which can be obtained with some localization schemes [19], [20]. Each node can forward packets to its neighbors within its transmission range that are closer to the sink node than itself.

Let N_i denote the set of neighbors of node i and $N_i = \{j|d_{ij} \leq R, j \in N\}$, where d_{ij} is the Euclidean distance of node i and node j, and R is the radius of the transmission range. Let us define the set of upstream neighbors as $A_i = \{k|d_{kt} \geq d_{it}, k \in N_i\}$, and similarly define the set of downstream neighbors as $S_i = \{k|d_{kt} < d_{it}, k \in N_i\}$. According to the *geometric routing* rule, the outgoing traffic from node i can only be forwarded towards the sink node through the set of downstream neighbors. For each downstream neighbor k, a routing variable ϕ_{ik} is associated with the link between node i and k that denotes the fraction of traffic to be routed from node i to node k. Clearly, the *flow conservation law* requires $\sum_{k \in B_i} \phi_{ik} = 1$.

The following theorem specifies the conditions under which the resulting network has directed acyclic property after applying the *geometric routing* rules. Due to the limited space, we omit the proof and refer the reader to the technical report [21].

Theorem 1: If the original graph $G(N, A)$ satisfies the conditions: 1) the graph is connected, i.e., every pair of nodes are connected by at least a path, and 2) there exists at least one neighbor $k \in N_i$ for each node i satisfying $d_{kt} < d_{it}$ and $d_{ik} < d_{it}$,

then the resulting graph $G'(N, A')$ is directed and acyclic graph (DAG) strongly connected to sink node t.

B. Data Aggregation Model

A salient feature of sensor networks is that data collected by the neighboring sensor nodes may carry redundant information due to the spatio-temporal correlation characteristics of the physical medium being sensed [22], [23], such as the temperature and humidity sensors in a similar geographic region or magnetometric sensors tracking a moving vehicle.

To remove the redundant information and reduce the traffic, it is necessary to aggregate the data at the intermediate nodes. To incorporate data aggregation with the *geometric routing*, we adopt the *foreign-coding* model [17]. In this model, a node i is assumed to be able to compress the data originating from its upstream neighbor j using its local data. The compression ratio between nodes i and j is characterized by the correlation coefficient $q_{ji} = 1 - H(X_j|X_i)/H(X_j)$, where $r_j = H(X_j)$ is the entropy coded data rate of the information X_j at node j, and $H(X_j|X_i) \leq r_j$ is the conditional entropy coded data rate of the same information X_j at node i given the side information X_i. Some correlation models have been proposed, such as the *Gaussian random field* model [16] which assumes the correlation coefficient q_{ji} decreases exponentially with the distance between nodes or $q_{ji} = \exp(-\alpha d_{ji}^2)$, and the inverse model [17] which assumes the data correlation is inverse proportional to the Euclidean distance between nodes or $q_{ij} = 1/(1 + d_{ji})$.

Using this data aggregation model, a node i performs two different operations for the data received from its upstream neighbors. For the raw data generated by the upstream neighbors, it encodes the data using the local information. For the transit data (already compressed by the upstream nodes), it directly forwards the data to the next-hop neighbors. Let r_i denote the traffic generating rate at node j, R_i and R_j denote the aggregated transit traffic at node i and j, respectively. The aggregated transit traffic R_i consists of two parts: the transit traffic passed from the upstream nodes and the raw data originated from the upstream nodes that is compressed using the local information, that is,

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

C. Power Consumption Model

A sensor node consumes power when it is sensing and generating data, receiving, transmitting, or even simply in standby mode. The power e_g for sensing and generating one bit of data is assumed to be the same for all nodes. The standby 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 used for receiving and transmitting, we adopt the *first-order radio model* in [6]. Specifically, a node needs $\epsilon_{elec} = 50$ nJ to run the circuitry and $\epsilon_{amp} = 100$ pJ/bit/m^2 for the transmitting amplifier. Therefore, the power consumption for receiving one bit of data is given by

$$e_r = \epsilon_{\text{elec}}. \tag{2}$$

The power consumption for transmitting one bit of data to a neighbor node j is given by

$$e_{ij} = \epsilon_{\text{elec}} + \epsilon_{\text{amp}} \cdot d_{ij}^{\alpha} \tag{3}$$

where α is the path loss exponent, which is usually between 2 and 4 for free-space and short-to-medium-range radio communication.

Assuming each node i has an initial battery energy E_i, the uniformed mean power consumption of node i, denoted as w_i, is given by

$$w_i = \left[e_s + e_g r_i + e_r \sum_{j \in A_i} (R_j + r_j) \phi_{ji} \right.$$
$$\left. + (R_i + r_i) \sum_{k \in S_i} e_{ik} \phi_{ik} \right] / E_i \tag{4}$$

where the first term is the standby 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.

D. Maximum Lifetime Routing Problem

The lifetime T_i of node i is the expected time for the node to run out of the battery energy, that is, $T_i = 1/w_i$ where w_i is given by (4). We define the network lifetime T_{net} as the time at which the first node in the network runs out of energy as that in [9], [10], that is,

$$T_{\text{net}} = \min_{i \in N} T_i. \tag{5}$$

The power consumption w_i is a function of r, R, and ϕ. However, the set of aggregated transit traffic R can be obtained from r and ϕ with (1). Therefore, T_{net} depends only on r, ϕ, and the initial battery energy E. If r and E are given, the maximum lifetime routing (MLR) is to find a set of routing variable ϕ such that the network lifetime is maximized. More formally,

$$\begin{aligned} \text{maximize} \quad & T_{\text{net}} \\ \text{subject to} \quad & \phi_{ik} \geq 0, \sum_{k \in S_i} \phi_{ik} = 1, \forall i. \end{aligned} \tag{6}$$

The definition of the MLR problem (6) is different from those in [9]–[11], and [13] in that data aggregation is considered and jointly optimized with the routing in our model. It is possible to consider more general network lifetime definitions such as that in [12], but these are beyond the scope of this work.

It is obvious that maximizing the network lifetime T_{net} is equivalent to minimizing the maximum normalized power consumption w_i for all $i \in N$. We therefore can rewrite the MLR problem as

$$\begin{aligned} \text{minimize} \quad & \max_{i \in N} w_i \\ \text{subject to} \quad & \phi_{ik} \geq 0, \sum_{k \in S_i} \phi_{ik} = 1, \forall i. \end{aligned} \tag{7}$$
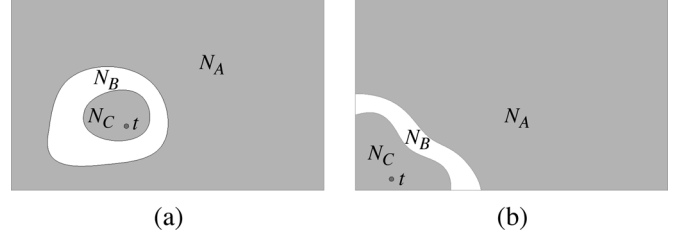

Fig. 1.   Two possible scenarios for node sets N_A, N_B, and N_C. (a) The sink node t is located in the anterior region. (b) The sink node t is located in the boundary region.

III. DISTRIBUTED SOLUTION FOR MLR PROBLEM

The distributed solutions based on the gradient algorithm are not directly applicable for the MLR problem as defined (7) since the max function is not differentiable. One solution for this problem is to transform the $\min - \max$ problem to an equivalent optimization problem by introducing an extra upper bound parameter (e.g., [24]), which is adopted in [11], where subgradient algorithms are developed to solve the dual optimization problem. However, subgradient algorithms are known to converge slowly. The other solution is to approximate the max function using some smoothing function, such as the entropy type approximation [25], [26], the two-dimensional approximation [27], and the recursive approximation [28]. Here, we first propose a smoothing function to approximate the max function in the MLR problem (7) by exploiting the special structure of the network. We then derive the necessary and sufficient conditions required to achieve the optimality of the approximate optimization problem.

A. Smoothing Function

Recalling in Section II-A that we have shown that, by applying the geometric routing, the original undirected network $G(N, A)$ is transformed to a directed acyclic graph (DAG) $G(N, A')$, where A' is the set of directed links, and sink node t is the root of the DAG. For any such DAG, we can find a *separation* $s = (N_A | N_B | N_C)$ to partition the node set N into three subsets N_A, N_B, and N_C, where N_B is the *cut set* that separates N_A and N_C into two disjoint sets. Without loss of generality, let the sink node be located in the subset N_C. Two possible scenarios are illustrated in Fig. 1, where Fig. 1(a) shows the case that the sink node is located in the interior region, and Fig. 1(b) shows the case that the sink node is located at the boundary region.

Normally, there are many such separations for a given DAG. For a separation s_i, there is a set of routing variables $\phi(s_i)$ for nodes in N_{A_i} and N_{B_i} that minimize the maximum energy consumption of the subset N_{B_i}, which we denote as $w(s_i) = \min \, max\{w_l, l \in N_{B_i}\}$. Among many such separations, there always exists a separation s^* with the largest minimax energy consumption rate $w(s^*)$, i.e., $s^* = \arg \, max\{w(s_1), w(s_2), \cdots\}$. We call the corresponding cutset N_B^* as the *bottleneck set* since this is the node set that limits the lifetime of the network. In practice, this set of bottleneck nodes are not known *a priori*, but can be identified
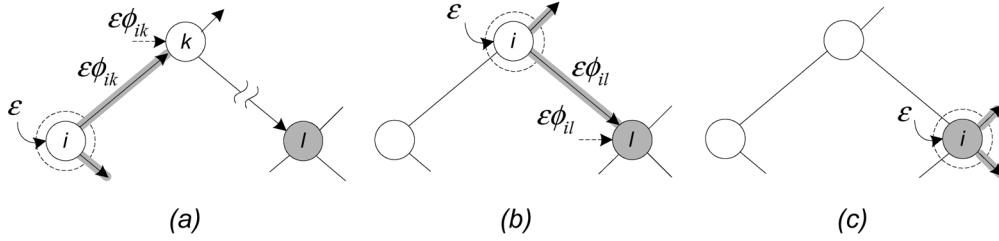
Fig. 2. Three possible relations of node i, k, and l. (a) Source node i and bottleneck node l are nonadjacent. (b) Source node i and bottleneck node l are adjacent. (c) Source node i and bottleneck node l are colocated.

dynamically during the run of algorithm, the details will be discussed in next section. Here, we assume the existence of such set of nodes, then the original MLR problem (7) is equivalent to minimizing of the maximum energy consumption with respect to the set of bottleneck nodes. More formally, we have

$$\text{minimize} \quad \max_{l \in N_B^*} w_l$$
$$\text{subject to} \quad \phi_{ik} \geq 0, \sum_{k \in S_i} \phi_{ik} = 1, \forall i. \tag{8}$$

Notice that (8) differs from (7) in only two aspects.
- The problem size of (8) is reduced from $|N|$ to $|N_B^*|$, where $|N|$ and $|N_B^*|$ are the size of the sets N and N_B^*, respectively.
- The set of values w_l, $l \in N_B^*$ may have smaller variance than those in N because they belong to the same cutset.

The reason for the second aspect is that, for any two nodes $i, j \in N_B^*$, if $w_i > w_j$, the difference of them can be minimized by adjusting the routing of upstream nodes to reduce the traffic of node i and increase that of node j, and *vice versa* if $w_i < w_j$. This may not be true for any two nodes $i, j \in N$.

The $\max$ function in (8) is still not differentiable. However, it can be approximated by the following smoothing function:

$$U(w, c) = \mu^2 + \frac{c}{|N_B^*|} \sum_{l \in N_B^*} (w_l - \mu)^2 \tag{9}$$

where $\mu = \sum_{l \in N_B} w_l / |N_B^*|$ is the mean power consumption of the bottleneck set and c is a positive scalar parameter.

The smoothing function $U(w, c)$ has two terms representing the mean and the variance of the power consumption of nodes in N_B^*. Minimizing the first term μ^2 will enforce data aggregation at the intermediate nodes so that the traffic passing through the bottleneck nodes is minimized. This causes the mean power consumption of these nodes to be minimized too. Minimizing the second term of $U(w, c)$ will cause the power consumption of the set of bottleneck nodes to be equalized, which has the effect of maximizing the network lifetime.

With c as a control parameter, the smoothing function $U(w, c)$ can be minimized successively using the method of multipliers [30, p. 244]. Let $c(n)$ and $\phi(n)$ denote the values of c and ϕ at the nth iteration, and $w(n)$ is the corresponding power consumption vector. We choose $c(n)$ to be an increasing sequence of positive numbers and update the routing variables as

$$\phi(n + 1) = \arg\min_{\phi \succeq 0} U(w(n), c(n)). \tag{10}$$

The iteration stops when a prescribed accuracy criterion is satisfied. The solution of (10) requires to derive the optimality conditions for $U(w(n), c(n))$, which will be described in the next subsection.

Thus, instead of solving problem (8), we can solve the following approximate problem:

$$\text{minimize} \quad U(w, c)$$
$$\text{subject to} \quad \phi_{ik} \geq 0, \sum_{k \in S_i} \phi_{ik} = 1, \forall i. \tag{11}$$

B. Optimality Conditions

To solve (11) in a distributed manner, using ϕ as the control variable, we extend the techniques in [31] to obtain the necessary and sufficient conditions for achieving the optimality of the smoothing function $U(w, c)$. Note the discussion in this section is applicable to nonbottleneck node cuts as well. So, without abusing the notations, we use N_A, N_B, and N_C to denote three corresponding subsets.

First of all, we can rewrite the smoothing function (9) as

$$U = \mu^2 + \frac{c}{|N_B|} \sum_{l \in N_B} (w_l - \mu)^2$$
$$= \frac{c}{|N_B|} \sum_{l \in N_B} w_l^2 - \frac{(c-1)}{|N_B|^2} \left(\sum_{l \in N_B} w_l \right)^2$$

where we use the fact that $\mu = \sum_{l \in N_B} w_l / |N_B|$. Since w is a function of the routing variable $\phi = \{\phi_{ij}\}$, by using the chain rule, we obtain

$$\frac{\partial U}{\partial \phi_{ik}} = \frac{2c}{|N_B|} \sum_{l \in N_B} w_l \frac{\partial w_l}{\partial \phi_{ik}} - \frac{2(c-1)}{|N_B|^2} \sum_{l \in N_B} w_l \sum_{l \in N_B} \frac{\partial w_l}{\partial \phi_{ik}}$$
$$= \frac{2}{|N_B|} \sum_{l \in N_B} (cw_l - (c-1)\mu) \frac{\partial w_l}{\partial \phi_{ik}}. \tag{12}$$

In order to find $\partial w_l / \partial \phi_{ik}$, we introduce a set of dummy variables r_i''s which can be interpreted as the dummy traffic injected into node i that follows the same set of routing as r_i, but without data aggregation. The partial derivative $\partial w_l / \partial \phi_{ik}$ involves three nodes i, k, and l, and we need to consider three possible relations of the source node i, its next-hop neighbor k, and the bottleneck node l as shown in Fig. 2.

1) Source Node and Bottleneck Node Are Nonadjacent: If the source node i is not adjacent to the bottleneck node l as shown in Fig. 2(a), let us consider a small increment ϵ to the input rate r_i'. This will cause an increment $\epsilon \phi_{ik}$ to the transit

rate of its next-hop neighbor k. Since node k is not a bottleneck node, 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 to all next-hop neighbors. Summing up over all $k \in S_i$ gives

$$\frac{\partial w_l}{\partial r'_i} = \sum_{k \in S_i} \phi_{ik} \frac{\partial w_l}{\partial r'_k}. \tag{13}$$

Now let us fix the aggregated traffic at node i and consider an increment ϵ to the routing variable ϕ_{ik}, which will cause an increment $\epsilon(R_i + r_i)$ to node k. Considering the data aggregation applied to r_i, this is equivalent to an increment of $\epsilon(R_i + r_i(1 - q_{ik}))$ to the input rate r'_k. Applying the similar reasoning as above, we obtain

$$\frac{\partial w_l}{\partial \phi_{ik}} = [R_i + r_i(1 - q_{ik})] \frac{\partial w_l}{\partial r'_k}. \tag{14}$$

2) Source Node and Bottleneck Node Are Adjacent: If the source node i is adjacent to the bottleneck node l as shown in Fig. 2(b), 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 above. Taking into account the indirect increment from other non-adjacent neighbor $k \neq l$ as derived above, we obtain

$$\frac{\partial w_l}{\partial r'_i} = \sum_{k \in S_i, k \neq l} \phi_{ik} \frac{\partial w_l}{\partial r'_k} + \phi_{il}\left(\frac{e_r}{E_l} + \frac{\partial w_l}{\partial r'_l}\right)$$
$$= \sum_{k \in S_i} \phi_{ik} \frac{\partial w_l}{\partial r'_k} + \frac{\phi_{il}e_r}{E_l}. \tag{15}$$

Similarly, an increment ϵ to ϕ_{il} leads to an increment of $\epsilon(R_i + r_i)$ to node l, and therefore

$$\frac{\partial w_l}{\partial \phi_{il}} = (R_i + r_i)\frac{e_r}{E_l} + [R_i + r_i(1 - q_{il})] \frac{\partial w_l}{\partial r'_l}. \tag{16}$$

3) Source Node and Bottleneck Node Are Colocated: If the source node i is also a bottleneck node as shown in Fig. 2(c), note that r'_i is the dummy traffic, so we do not consider the power consumption for generating traffic ϵ. Taking the derivative directly from (4), we have

$$\frac{\partial w_i}{\partial r_i} = \sum_{k \in S_i} \frac{e_{ik}\phi_{ik}}{E_i} \tag{17}$$

$$\frac{\partial w_i}{\partial \phi_{ik}} = \frac{(R_i + r_i)e_{ik}}{E_i}. \tag{18}$$

Another case is $i \in N_C$ and $l \in N_B$. However, this case is not interesting because both $\partial w_l/\partial r'_i$ and $\partial w_l/\partial \phi_{ik}$ are zeros since w_l does not depend on r'_i.

We can now combine the above results to derive $\partial U/\partial \phi_{ik}$ of (12) by considering the following four cases.

- **Case 1**: If $i, k \in N_A$, then none of the bottleneck nodes are adjacent to node i, so we can obtain $\partial w_l/\partial \phi_{ik}$ from (14) for all $l \in N_B$. Substituting these into (12), we have

$$\frac{\partial U}{\partial \phi_{ik}} = \frac{2[R_i + r_i(1 - q_{ik})]}{|N_B|} \sum_{l \in N_B} [cw_l - (c-1)\mu] \frac{\partial w_l}{\partial r'_k}. \tag{19}$$

- **Case 2**: If $i \in N_A$, $k \in N_B$, then node k is a bottleneck node adjacent to node i. Therefore, $\partial w_k/\partial \phi_{ik}$ is given by (16), while for other bottleneck nodes $l \neq k$, $\partial w_l/\partial \phi_{ik}$ is given by (14). Substituting these into (12), we have

$$\frac{\partial U}{\partial \phi_{ik}} = \frac{2[R_i + r_i(1 - q_{ik})]}{|N_B|} \sum_{l \in N_B} [cw_l - (c-1)\mu] \frac{\partial w_l}{\partial r'_k}$$
$$+ \frac{2(R_i + r_i)}{|N_B|} \frac{e_r}{E_k} [cw_k - (c-1)\mu]. \tag{20}$$

- **Case 3**: If $i, k \in N_B$, then node i and k are adjacent bottleneck nodes, so $\partial w_i/\partial \phi_{ik}$ and $\partial w_k/\partial \phi_{ik}$ are given by (18) and (16), respectively. Therefore

$$\frac{\partial U}{\partial \phi_{ik}} = \frac{2[R_i + r_i(1 - q_{ik})]}{|N_B|} \sum_{l \in N_B} [cw_l - (c-1)\mu] \frac{\partial w_l}{\partial r'_k}$$
$$+ \frac{2(R_i + r_i)}{|N_B|}\left(\frac{e_{ik}}{E_i}[cw_i - (c-1)\mu]\right.$$
$$\left. + \frac{e_r}{E_k}[cw_k - (c-1)\mu]\right). \tag{21}$$

- **Case 4**: If $i \in N_B$ and $k \in N_C$, the source node i is also a bottleneck node, so $\partial w_i/\partial \phi_{ik}$ is given by (18). Therefore, we have

$$\frac{\partial U}{\partial \phi_{ik}} = \frac{2[R_i + r_i(1 - q_{ik})]}{|N_B|} \sum_{l \in N_B} [cw_l - (c-1)\mu] \frac{\partial w_l}{\partial r'_k}$$
$$+ \frac{2(R_i + r_i)}{|N_B|} \frac{e_{ik}}{E_i} [cw_i - (c-1)\mu]. \tag{22}$$

Now all that is required to minimize $U(w, c)$ is to find the stationary points for the routing variable ϕ. Applying the Lagrange multiplier ν_i to the constraints $\sum_{k \in S_i} \phi_{ik} = 1$ and taking into account the constraint $\phi_{ik} \geq 0$, the necessary condition for ϕ to be a minimizer of $U(w, c)$ is given by the following theorem.

Theorem 2: (Necessary Condition): Let $\partial U/\partial \phi_{ik}$ be given by (19)–(22), then the necessary condition for a minimum of $U(w, c)$ with respect to ϕ^* to exist for all $i \in N_A \cup N_B$, $k \in S_i$ is

$$\frac{\partial U}{\partial \phi^*_{ik}} = \begin{cases} = \nu_i, & \phi^*_{ik} > 0 \\ \geq \nu_i, & \phi^*_{ik} = 0. \end{cases} \tag{23}$$

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

We prove further that the sufficient condition to minimize U with respect to ϕ for all $i \in N_A \cup N_B$, $k \in S_i$ is given by the following theorem.

Theorem 3: (Sufficient Condition): Let $\partial U/\partial\phi_{ik}$ be given by (19)–(22) and define $\partial U/\partial r_k' = \sum_{l\in N_B}[cw_l - (c-1)\mu]\partial w_l/\partial r_k'$, it is then sufficient for ϕ to be the minimizer of $U(w,c)$ if for all $i \in N_A \cup N_B$, $k \in S_i$, there is

$$(R_i + r_i(1-q_{ik}))\frac{\partial U}{\partial r_k'}$$
$$\geq (R_i + r_i)\frac{\partial U}{\partial r_i'}$$
$$(R_i + r_i(1-q_{ik}))\frac{\partial U}{\partial r_k'} + (R_i + r_i)\frac{e_r}{E_k}[cw_k - (c-1)\mu]$$
$$\geq (R_i + r_i)\frac{\partial U}{\partial r_i'}$$
$$(R_i + r_i(1-q_{ik}))\frac{\partial U}{\partial r_k'} + (R_i + r_i)\frac{e_{ik}}{E_i}[cw_i - (c-1)\mu]$$
$$+ (R_i + r_i)\frac{e_r}{E_k}(cw_k - (c-1)\mu)$$
$$\geq (R_i + r_i)\frac{\partial U}{\partial r_i'}$$
$$(R_i + r_i(1-q_{ik}))\frac{\partial U}{\partial r_k'} + (R_i + r_i)\frac{e_{ik}}{E_i}[cw_i - (c-1)\mu]$$
$$\geq (R_i + r_i)\frac{\partial U}{\partial r_i'} \qquad (24)$$

which correspond to the four cases given by (19)–(22), respectively.

The proofs of Theorems 2 and 3 are given in Appendix A and B, respectively. Note that the sufficient condition is always satisfied for a node i if both the transit traffic rate R_i and the local data rate r_i are zeros. This may lead to inflection points to $U(w,c)$ as a function of ϕ_{ik}'s. To avoid this problem, it is necessary that the local data rates of all nodes should be non-zero. This assumption is reasonable for the sensor network in our study since all sensors are expected to generate data constantly.

Let us define two indicator variables I_i and I_k, where I_i is 1 if $i \in N_B$ and 0 otherwise, and I_k is 1 if $k \in N_B$ and 0 otherwise. Let $Z_{ik} = I_i e_{ik}[cw_i-(c-1)\mu]/E_i + I_k e_r[cw_k-(c-1)\mu]/E_k$, then the sufficient condition in (24) can be simplified as

$$[R_i + r_i(1-q_{ik})]\frac{\partial U}{\partial r_k'} + (R_i + r_i)Z_{ik} \geq (R_i + r_i)\frac{\partial U}{\partial r_i'} \quad (25)$$

for all $i \in N_A \cup N_B$, $k \in S_i$, where the equality is achieved for k whose routing variable ϕ_{ik} is greater than 0. In other words, traffic is only distributed over those links with the smallest and identical values of $[R_i + r_i(1-q_{ik})](\partial U/\partial r_k') + (R_i + r_i)Z_{ik}$ when the optimality has been achieved,

IV. DISTRIBUTED ALGORITHM AND PROTOCOL

Here, we present a routing adaptation algorithm for every node to compute the routing variables based on the sufficient conditions derived in previous section. For each node i, the algorithm requires the feedback of the following information from its downstream neighbors:
- the set of bottleneck nodes N_B;

- the power consumption w_l of nodes $l \in N_B$;
- the power consumption rate $\partial w_l/\partial r_i'$ of nodes $l \in N_B$;
- the data correlation coefficient q_{ik} of next-hop neighbors $k \in S_i$.

In the following, we first discuss the procedure for identifying the bottleneck nodes and the computation of $\partial U/\partial r_k'$. We then present the routing adaptation algorithm and explain the procedure of the MLR protocol.

A. Bottleneck Node Identification

In the previous section, we defined the bottleneck node set N_B as the set of nodes that have the largest minimax power consumption among all possible cutsets. This suggests that we can determine whether a node belongs to the bottleneck set by comparing its power consumption with that of its downstream neighbors. Let each node i maintain two variables: 1) its power consumption w_i and 2) the weighted power consumption of the set of bottleneck nodes known by node i, denoted by $w_{B,i}$, which is given by

$$w_{B,i} = \max\left\{w_i, \sum_{k\in S_i}\phi_{ik}w_{B,k}\right\} \qquad (26)$$

that is, if $w_i > \sum_{k\in S_i}\phi_{ik}w_{B,k}$, node i labels itself as a bottleneck node and set $w_{B,i} = w_i$, otherwise it sets $w_{B,i} = \sum_{k\in S_i}\phi_{ik}w_{B,k}$ and passes it to the upstream neighbors.

B. Computing $\partial U/\partial r_k'$

Since the network is a DAG, it is possible that only a subset of the bottleneck nodes are located on the downstream paths of a source node. If a bottleneck node l is not on the downstream paths of a node k, the input rate r_k will have no contribution to the power consumption of node l, so $\partial w_l/\partial r_k = 0$. Let $N_{B,k}$ denote the set of downstream bottleneck nodes known by node k. Then, the overall power consumption rate $\partial U/\partial r_k$ in sufficient condition (25) needs only to include those terms in $N_{B,k}$ or

$$\frac{\partial U}{\partial r_k'} = \sum_{l\in N_{B,k}}[cw_l - (c-1)\mu]\frac{\partial w_l}{\partial r_k'}. \qquad (27)$$

Since only the values w_l from the subset $N_{B,k}$ are available at node k, we can approximate the global mean μ with $\mu \approx \sum_{l\in N_{B,k}} w_l/|N_{B,k}|$.

C. Routing Adaptation Algorithm

Every node executes a routing adaptation algorithm to update its routing variables according to the received information from downstream neighbors. The algorithm is operated in following steps.
1) Calculate $[R_i + r_i(1-q_{ik})]\partial U/\partial r_k' + (R_i + r_i)Z_{ik}$ for every neighbor $k \in S_i$ and find the best neighbor $k_{\min}$ such that

$$k_{\min} \leftarrow \arg\min_{k\in S_i}\left\{[R_i + r_i(1-q_{ik})]\frac{\partial U}{\partial r_k'} + (R_i + r_i)Z_{ik}\right\}.$$
$$(28)$$

2) Calculate the amount of reduction $\Delta\phi_{ik}$ to each $\phi_{ik}(k \neq k_{\min})$. Let a_{ik} be the gradient difference between each neighbor k and $k_{\min}$, that is,

$$
\begin{aligned}
a_{ik} = & [R_i + r_i(1 - q_{ik})] \frac{\partial U}{\partial r'_k} + (R_i + r_i)Z_{ik} \\
& - \left\{ [R_i + r_i(1 - q_{ik_{\min}})] \frac{\partial U}{\partial r'_{k_{\min}}} + (R_i + r_i)Z_{ik_{\min}} \right\}.
\end{aligned}
\tag{29}
$$

The amount of routing reduction $\Delta\phi_{ik}$ to ϕ_{ik} should be proportional to a_{ik} and inversely proportional to $R_i + r_i$ so that the change of link traffic will not greatly affect the objective function. In addition, taking into account that the routing variable ϕ_{ik} cannot be negative, $\Delta\phi_{ik}$ is given by

$$
\Delta\phi_{ik} = \min\left\{ \phi_{ik}, \gamma a_{ik}/(R_i + r_i) \right\}, \quad k \in S_i
\tag{30}
$$

where γ is a positive scale parameter.

3) Update routing variables as follows:

$$
\phi_{ik} \leftarrow \phi_{ik} - \Delta\phi_{ik}, \quad k \in S_i, k \neq k_{\min}
\tag{31}
$$

and

$$
\phi_{ik_{\min}} \leftarrow \phi_{ik_{\min}} + \sum_{k \in S_i, \neq k_{\min}} \Delta\phi_{ik}.
\tag{32}
$$

Using this algorithm, each node i gradually decreases the routing variables for which the values $[R_i + r_i(1 - q_{ik})]\partial U/\partial r'_k + (R_i + r_i)Z_{ik}$ are larger and increases the routing variable for which the value is the smallest until the sufficient condition (25) is satisfied.

D. Summary of MLR Protocol

Each node i maintains a table for the bottleneck nodes with $|N_{B,i}|$ entries. Each entry contains the node identity l, the power consumption w_l and the power consumption rate $\partial w_l/\partial r'_i$ for a bottleneck node l.

In each iteration, the MLR protocol is operated as follows by each node i.

1) Wait until receiving the table from all of its downstream neighbors and merge the bottleneck set $N_{B,k}$ of neighbor k with the local bottleneck set $N_{B,i}$ to give $N_{B,i} \leftarrow N_{B,i} \cup N_{B,k}$.

2) Calculate new routing variables using the routing-adaptation algorithm.

3) Calculate the power consumption w_i and $w_{B,i}$ with (4) and (26) to perform bottleneck node identification, and
   a) If node i is not identified as a bottleneck node, calculate the power consumption rate $\partial w_l/\partial r'_i$ for all $l \in N_{B,i}$ using the recursive (13).
   b) If node i is a bottleneck node, create a bottleneck table with a single entry and fill the table with w_i and $\partial w_i/\partial r'_i$.

4) Pass the bottleneck table to upstream neighbors.

Each iteration of the MLR algorithm involves the communication between the neighboring nodes, and the communication overhead is bounded by the number of bottleneck nodes at the downstream paths. Thus, each node can run the algorithm in a decentralized manner, and the energy consumption for running this algorithm is determined by the number of the iterations until the algorithm converges.

V. PERFORMANCE EVALUATION

A. Simulation Setup

Here, we compare the performance of MLR algorithm (centralized and distributed) with the minimum energy gathering algorithm (MEGA) [17] and the minimum energy routing (MER) algorithms.

1) MLR (centralized)—The results for centralized MLR algorithm are obtained by directly solving the MLR problem (7) using the $fminimax$ function in MATLAB.

2) MLR (distributed)—The distributed MLR algorithm is the one presented in previous section.

3) 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 shortest path tree (SPT). The coding tree is for data aggregation and constructed with directed minimum spanning tree (DMST) algorithm, and the SPT is for delivery of compressed data.

4) MER—This algorithm tries to minimize the overall energy consumption of delivery of a packet by using the shortest path from the source node to the sink node in term of energy cost. For fair comparison, we revised the MER algorithm to take into account the data correlation effects, that is, raw data packet is firstly compressed at the nexthop node along the shortest path. After that, the compressed data is delivered through the shortest path.

We evaluate these four algorithms over a set of sensor networks with the number of nodes ranging from 20 to 80. For the same number of nodes, we randomly generate twenty network topologies and run these algorithms over them to obtain the average results. In each network, the sensor nodes are randomly distributed on a 100 m × 100 m square. 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 $\alpha = 2$. For data correlation setting, we adopt the *Gaussian random field* model [16] 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 from $\alpha = 0.001/\text{m}^2$ (high correlation) to $\alpha = 0.01/\text{m}^2$ (low correlation) in the experiments. All nodes have uniform battery energy of $E_i = 1$ kJ and the data generating rate is 1 kbps. Also, a decreasing sequence of step size γ and an increasing sequence of c are used for the distributed MLR algorithm in the experiments.

B. Network Lifetime

Fig. 3 shows the network lifetime under two data correlation settings ($\alpha = 0.001$ and $\alpha = 0.01$). From this figure, we first see that the results of the distributed MLR algorithm are very
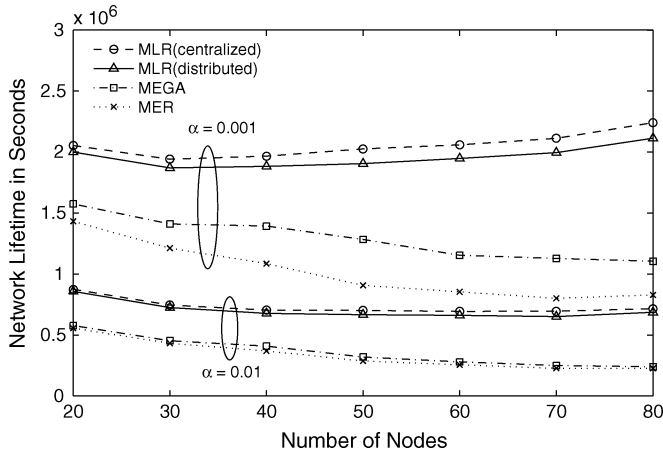
Fig. 3. Network lifetime versus network size.

Fig. 4. Sink node data rate versus network size.

Fig. 5. Network lifetime versus correlation parameter.

close to those of the optimal (centralized) MLR algorithm under all situations. This demonstrates that the smoothing function is a good approximation to the original objective function since the network lifetime of sensor networks is determined by the bottleneck nodes due to the special structure of sensor networks.

We also see from the figure that the network lifetime obtained by MLR algorithm is almost twice of that obtained by MEGA and MER algorithms. In particular, the network lifetime returned by MLR algorithm increases gradually as the network size grows, while those of MEGA and MER algorithms decrease continuously. The reason is that the overall source data rate is proportional to the number of nodes in the network. Therefore, it is expected that the network lifetime should decrease with the increase of network size as more traffic is generated. However, the increase of nodes in the network also drives the network topology from sparse to dense, which has two effects. First, the distance between neighboring nodes 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 with data aggregation. Both effects help reduce the energy consumption. However, MEGA and MER algorithms fail to take advantage of this feature and the network lifetime returned by both algorithms 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 does not. 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 of 80 nodes, the network lifetime obtained by MLR algorithm is around twice that given by MEGA algorithm and 3 times that given by MER algorithm with $\alpha = 0.001$. For $\alpha = 0.01$, the network lifetime of MLR algorithm is around three times of those given by both MEGA and MER algorithms.

C. Aggregated Data Rate at Sink Node

Fig. 4 shows the aggregated data rate at the sink node. We can see that MLR algorithm has better aggregation results than
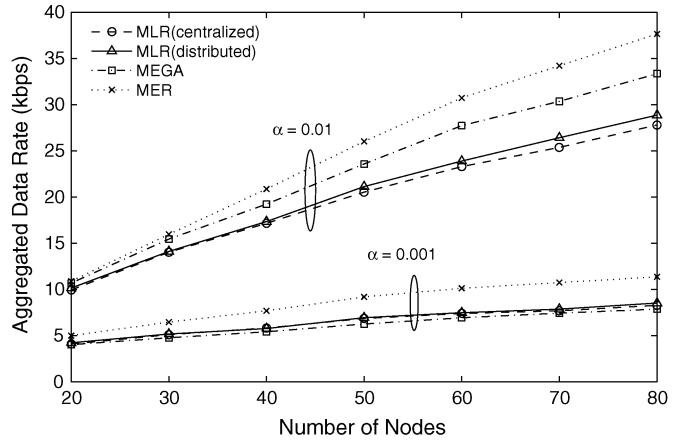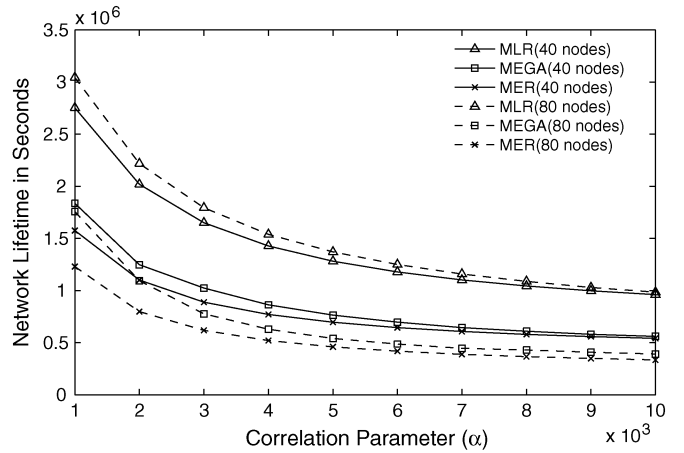
MER algorithm. For MEGA algorithm, its aggregated rate is comparable to that of the MLR algorithm under higher correlation condition ($\alpha = 0.001$), but is higher than MLR algorithm under lower correlation condition. Comparing with the results from Fig. 3, we can see that MEGA algorithm successfully optimizes data aggregation paths, but fails to balance the traffic since it use the shortest path to deliver compressed data, which tends to overwhelm the hotspot nodes. Therefore, under lower correlation condition, the network lifetime of MEGA and MER algorithms are quite similar.

D. Impact of Data Correlation

In Fig. 5, we show the average network lifetimes given by MLR, MEGA, and MER algorithms as the correlation parameter α increases from 0.001 to 0.01. We can see that 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 degenerate as the data correlation becomes smaller. This is observed in Fig. 3. On the other hand, the network lifetime of MLR increases with network size. This difference diminishes as the correlation parameter grows larger(which reduces the data correlation). Under the same settings, we show in Fig. 6 the aggregated data rate at the
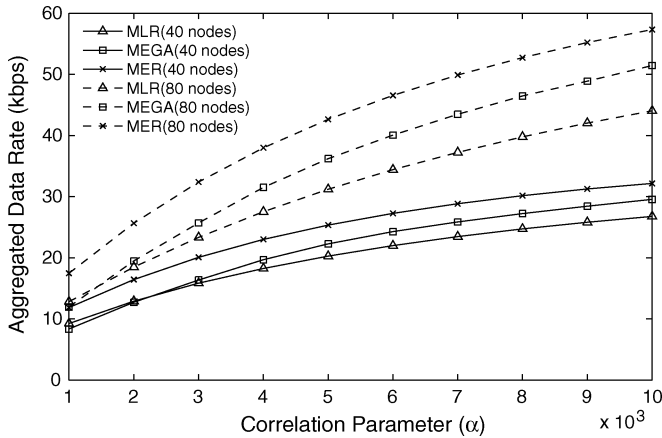
Fig. 6. Sink node data rate versus correlation parameter.

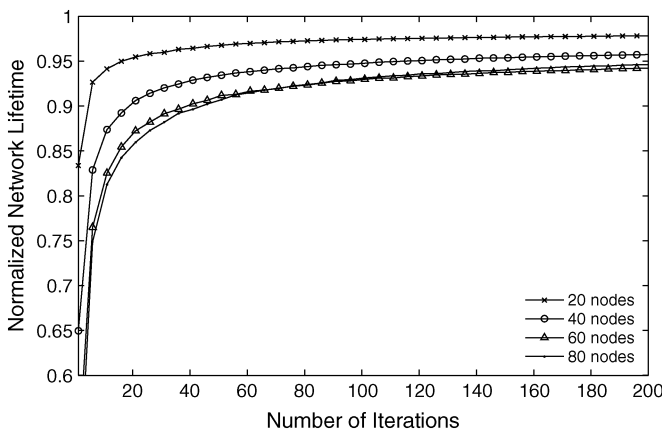Fig. 8. Normalized sink node data rate versus iteration.

Fig. 7. Normalized network lifetime versus iteration.

sink node with these algorithms. We see that MLR algorithm effectively reduces the network traffic compare with MEGA and MER algorithms.

E. Convergence of the Distributed Algorithm

In addition to the effectiveness of the MLR algorithm, we are also interested in knowing the efficiency of the distributed algorithm, or how fast the algorithm can converge to the optimal values given by the centralized algorithm. Fig. 7 shows the normalized network lifetime obtained by distributed 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 obtained by the centralized MLR algorithm. We see that the distributed algorithm can converge to the optimal values efficiently. The number of iterations required for the network lifetime to converge to over 90% of the optimal values is 5, 20, 35, and 40 iterations respectively for network size ranging from 20 to 80 nodes. The final network lifetimes are around 95% of the optimal values all network sizes. The effectiveness of the distributed MLR algorithm can also be observed from Fig. 8 which shows the normalized aggregated data rate at the sink node for various network sizes. The aggregated data rate is normalized with respect to the optimal value obtained by the centralized MLR algorithm. Again, we see that the distributed MLR algorithm successfully reduces the data rate
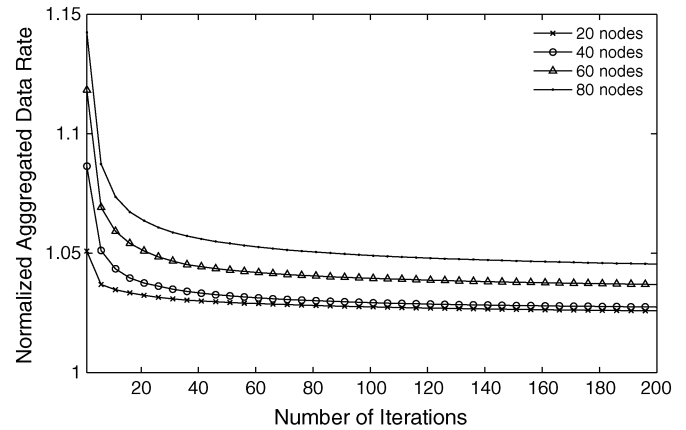
and achieves closer approximation ratios (below 105%) of the optimal results returned by the centralized algorithm.

F. Bottleneck Nodes Identification

In Section IV, we propose a heuristic bottleneck node identification procedure. We now illustrate the effectiveness of this algorithm under two different settings: 1) uniform battery energy and 2) nonuniform battery energy.

For uniform battery energy setting, we choose a network topology with 80 nodes and setup the same battery energy for all nodes as previous experiments. We run the MLR algorithm (centralized and distributed algorithm) and record the lifetimes of all nodes. In Fig. 9, we plot the node locations and indicate the bottleneck nodes returned by both algorithms, where the node with star mark is the sink node locating at the left-bottom corner. The set of bottleneck nodes (nodes with the smallest lifetime) returned by the centralized and distributed MLR algorithms are indicated by circle and cross marks, respectively. As expected, the nodes close to the sink node are the bottleneck nodes because these nodes have to forward traffic for those nodes far away from the sink node. Since the initial batter energy is the same for all nodes, these nodes tend to run out of energy first. From this figure, we also see that the distributed MLR algorithm is effective as it can identify most of the bottleneck nodes found by the centralized MLR algorithms.

For nonuniform setting, we choose three nodes that are far away from the sink node and set their initial battery energy to only 25% of other nodes. We repeat the same experiments and plot the results in Fig. 10. We can see that some bottleneck nodes shown in Fig. 9 are no longer found in Fig. 10, but some new ones are identified. More importantly, three chosen nodes labeled with A, B, and C are identified as bottleneck nodes by both algorithms since they have lower battery energy, even though they are far way from the sink node. Again, most of the bottleneck nodes identified by the distributed MLR algorithm coincide with those found by the centralized MLR algorithm.

VI. Conclusion

In this paper, we have presented an optimal routing and data aggregation scheme for maximizing the network lifetime of sensor networks. By exploiting the special structure of the
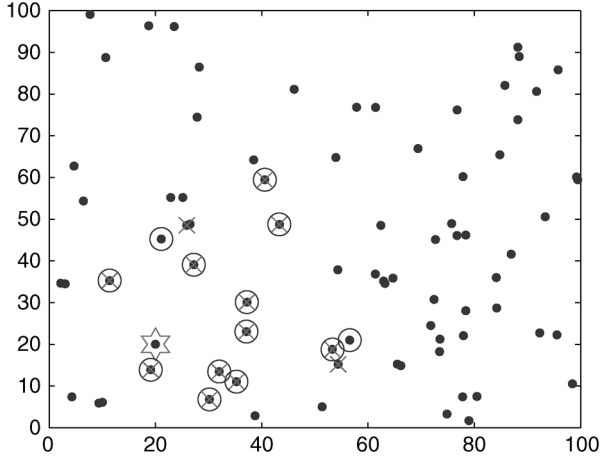
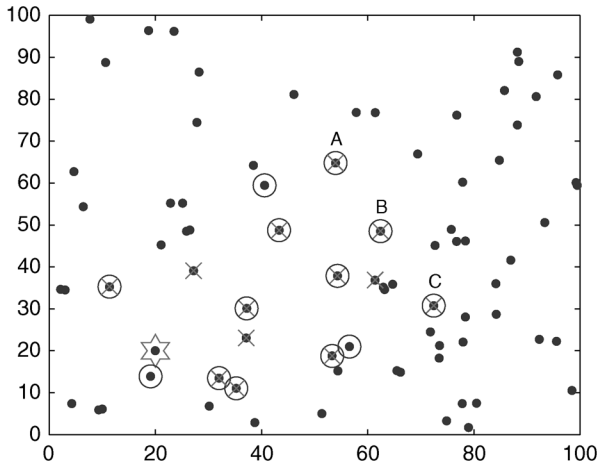Fig. 9. Bottleneck nodes with uniform battery energy.

Fig. 10. Bottleneck nodes with nonuniform battery energy.

sensor network, we have proposed a smoothing approximate function to overcome the nondifferentiability of original optimization problem so that the distributed solution is possible. The optimality conditions are derived and a distributed algorithm is designed accordingly. We have shown that the scheme can significantly reduce data traffic and improve the network lifetime. The distributed algorithm can converge to the optimal value efficiently. Extension of our work for multiple sink nodes and for nodes with sleeping mode would be of interest, but these are beyond the scope of this paper.

APPENDIX A

A. Proof of Necessary Conditions

Proof: We prove that (23) is the necessary condition to minimize U by defining the following Lagrange function:

$$U(w, c, \nu, \mu) = U + \sum_{i \in N} \nu_i \left(1 - \sum_{k \in S_i} \phi_{ik} \right) - \sum_{i \in N, k \in S_i} \mu_{ik} \phi_{ik}$$
(A1)

where $\nu = (\nu_1, \cdots, \nu_N)$ and $\mu = \{\mu_{ik}\}$ are the Lagrange multipliers. According to the Kuhn–Tucker theorem, the necessary

condition for a ϕ^* to be a minimizer of $U(w, c, \nu, \mu)$ is that there exist Lagrange multipliers ν_i^*, $i \in N$ and μ_{ik}^*, $i \in N, k \in S_i$ such that

$$\frac{\partial U}{\partial \phi_{ik}^*} - \nu_i^* - \mu_{ik}^* = 0$$
$$\mu_{ik}^* = 0, \text{ if } \phi_{ik}^* > 0, \quad \forall i, k$$
$$\mu_{ik}^* > 0, \text{ if } \phi_{ik}^* = 0, \quad \forall i, k. \tag{A2}$$

Rearranging the first equation as $\partial U / \partial \phi_{ik}^* = \nu_i^* + \mu_{ik}^*$ and taking into account the second and third conditions will complete the proof of (23). ∎

B. Proof of Sufficient Conditions

Proof: Suppose that there is a set of routing variables ϕ^* satisfying (24), then the corresponding node flows are R^* and link flows are f^*, where $f_{ik} = (R_i + r_i)\phi_{ik}, i \in N, k \in S_i$. Let ϕ be any other set of routing variables with the corresponding node flows R and link flows f. Define $f(\theta)$ as the convex combination of f^* and f with respect to a variable θ, that is,

$$f_{ik}(\theta) = (1 - \theta) f_{ik}^* + \theta f_{ik}. \tag{A3}$$

Therefore, each $w_l, l \in N_B$ can be represented by the link flow f, which in turn is a function of θ, so U is also a function of θ. We rewrite the smoothing function (9) as

$$U(\theta) = \frac{c}{|N_B|} \sum_{l \in N_B} w_l^2(\theta) - \frac{(c-1)}{|N_B|^2} \left(\sum_{l \in N_B} w_l(\theta) \right)^2. \tag{A4}$$

Since each $w_l(\theta)$ is a convex function of the node flow f, therefore $U(\theta)$ is also a convex function with respect to θ, so it is obvious that

$$\left. \frac{dU(\theta)}{d\theta} \right|_{\theta=0} \leq U(\phi) - U(\phi^*). \tag{A5}$$

Since ϕ is an arbitrary set of routing variable, it will complete the proof by proving that $dU(\theta)/d\theta \geq 0$ at $\theta = 0$.

From (4) and (A3), we can express w_l as a function of the link flow $f(\theta)$ as

$$w_l(\theta) = \frac{1}{E_l} \left(e_s + e_g r_l + \sum_{i \in A_l} f_{il}(\theta) e_r + \sum_{k \in S_l} f_{lk}(\theta) e_{lk} \right). \tag{A6}$$

Differentiating w_l with respect to θ from (A3) and (A6), we obtain

$$\frac{\partial w_l}{\partial \theta} = \sum_{i \in A_l} \frac{e_r}{E_l} \left(f_{il} - f_{il}^* \right) + \sum_{k \in S_l} \frac{e_{lk}}{E_l} \left(f_{lk} - f_{lk}^* \right). \tag{A7}$$

We can calculate $dU(\theta)/d\theta$ directly using (A4) and (A7) to yield

$$\left. \frac{dU(\theta)}{d\theta} \right|_{\theta=0} = \frac{2c}{|N_B|} \sum_{l \in N_B} w_l \frac{\partial w_l}{\partial \theta}$$
$$- \frac{2(c-1)}{|N_B|^2} \sum_{l \in N_B} w_l \sum_{l \in N_B} \frac{\partial w_l}{\partial \theta}$$

$$= \frac{2}{|N_B|} \sum_{l \in N_B} \left[cw_l - \frac{(c-1)}{|N_B|} \sum_{l \in N_B} w_l \right] \cdot \frac{\partial w_l}{\partial \theta}$$

$$= \frac{2}{|N_B|} \sum_{l \in N_B} [cw_l - (c-1)\mu]$$
$$\cdot \left(\sum_{i \in A_l} \frac{e_r}{E_l} (f_{il} - f_{il}^*) \right.$$
$$\left. + \sum_{k \in S_l} \frac{e_{lk}}{E_l} (f_{lk} - f_{lk}^*) \right). \tag{A8}$$

We then first prove that

$$\sum_{l \in N_B} [cw_l - (c-1)\mu] \cdot \left(\sum_{i \in A_l} \frac{e_r f_{il}}{E_l} + \sum_{k \in S_l} \frac{e_{lk} f_{lk}}{E_l} \right)$$
$$\geq \sum_{i \in N_A \cup N_B} r_i \frac{\partial U}{\partial r_i'}. \tag{A9}$$

Note that, from (24), multiplying both sides of these equations with ϕ_{ik}, summing over all $i \in N_A \cup N_B$ and $k \in S_i$, and using the fact that $R_i = \sum_{j \in A_i} (R_j + r_j(1 - q_{ji}))\phi_{ji}$, we can obtain the result for the left-hand side as

$$\text{LHS} = \sum_{i \in N_A \cup N_B} \sum_{k \in S_i} (R_i + r_i(1 - q_{ik})) \phi_{ik} \frac{\partial U}{\partial r_k'}$$
$$+ \sum_{i \in N_A} \sum_{k \in S_i, k \in N_B} \left(\frac{(R_i + r_i)\phi_{ik} e_r}{E_k} [cw_k - (c-1)\mu] \right)$$
$$+ \sum_{i \in N_B} \sum_{k \in S_i, k \in N_B} \left(\frac{(R_i + r_i)\phi_{ik} e_{ik}}{E_i} [cw_i - (c-1)\mu] \right)$$
$$+ \sum_{i \in N_B} \sum_{k \in S_i, k \in N_B} \left(\frac{(R_i + r_i)\phi_{ik} e_r}{E_k} [cw_k - (c-1)\mu] \right)$$
$$+ \sum_{i \in N_B} \sum_{k \in S_i, k \in N_C} \left(\frac{(R_i + r_i)\phi_{ik} e_{ik}}{E_i} [cw_i - (c-1)\mu] \right)$$
$$\tag{A10}$$

and the right-hand side as

$$\text{RHS} = \sum_{i \in N_A \cup N_B} r_i \frac{\partial U}{\partial r_i'}$$
$$+ \sum_{i \in N_A \cup N_B} \sum_{j \in A_i} (R_j + r_j(1 - q_{ji})) \phi_{ji} \frac{\partial U}{\partial r_i'}. \tag{A11}$$

Now let us look at the first term of the left-hand side in (A10), which sums over all links directed from nodes $i \in N_A \cup N_B$. Similarly, the second term of the right-hand side in (A11) sums over all in links directed to nodes $i \in N_A \cup N_B$. Recalling that the network is directed acyclic, canceling the common part of these two terms, the remaining part of the first term of (A10) is the sum over all links (i, k), $i \in N_B$, $k \in N_C$, which is zero because $\partial w_i / \partial r_k$ are zero for these links. In other words, we can totally cancel out the first term of (A10) and the second term of (A11).

Rearranging the summation of the second, third, and fourth terms of the left-hand side in (A10) and recalling the inequality between (A10) and (A11), we obtain

$$\sum_{l \in N_B} [cw_l - (c-1)\mu] \left(\sum_{i \in A_l} \frac{e_r}{E_l} (R_i + r_i)\phi_{il} + \right.$$
$$\left. \sum_{k \in S_l} \frac{e_{lk}}{E_l} (R_l + r_l)\phi_{lk} \right) \geq \sum_{i \in N_A \cup N_B} r_i \frac{\partial U}{\partial r_i'}. \tag{A12}$$

Note that $f_{il} = (R_i + r_i)\phi_{il}$, substituting this into (A12), we can obtain (A9).

Following the same derivation procedure, if R^* and ϕ^* are substituted for R and ϕ, this becomes an equality from the equations for $\partial U / \partial r_i$ in (24), that is,

$$\sum_{l \in N_B} [cw_l - (c-1)\mu] \left(\sum_{i \in A_l} \frac{e_r f_{il}^*}{E_l} + \sum_{k \in S_l} \frac{e_{lk} f_{lk}^*}{E_l} \right)$$
$$= \sum_{i \in N_A \cup N_B} r_i \frac{\partial U}{\partial r_i'}. \tag{A13}$$

Substituting (A9) and (A13) into (A8), we see that $dU(\theta)/d\theta \geq 0$ at $\theta = 0$, which completes the proof. ∎

REFERENCES

[1] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie, "Protocols for self-organization of a wireless sensor network," *IEEE Pers. Commun.*, vol. 7, no. 5, pp. 16–27, Oct. 2000.

[2] Y. Xu, J. S. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *Proc. MobiCom*, 2001, pp. 70–84.

[3] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," in *Proc. MobiCom*, 2001, pp. 85–96.

[4] L. Li, J. Y. Halpern, P. Bahl, Y.-M. Wang, and R. Wattenhofer, "Analysis of a cone-based distributed topology control algorithm for wireless multi-hop networks," in *Proc. ACM Symp. Principles Distrib. Comput.*, 2001, pp. 264–273.

[5] N. Li, J. C. Hou, and L. Sha, "Design and analysis of an MST-based topology control algorithm," in *Proc. IEEE INFOCOM*, 2003, pp. 1702–1712.

[6] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. Int. Conf. Syst. Sci.*, 2000.

[7] S. Singh, M. Woo, and C. S. Raghavendra, "Power-aware routing in mobile ad hoc networks," in *Proc. MobiCom*, 1998, pp. 181–190.

[8] T. H. Meng and V. Rodoplu, "Minimum energy mobile wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 8, pp. 1333–1344, Aug. 1999.

[9] J.-H. Chang and L. Tassiulas, "Energy conserving routing in wireless ad hoc networks," in *Proc. IEEE INFOCOM*, 2000, pp. 22–31.

[10] A. Sankar and Z. Liu, "Maximum lifetime routing in wireless ad hoc networks," in *Proc. IEEE INFOCOM*, Mar. 2004, pp. 1089–1097.

[11] R. Madan and S. Lall, "Distributed algorithms for maximum lifetime routing in wireless sensor networks," in *Proc. IEEE GLOBECOM*, Nov. 2004, vol. 2, pp. 748–753.

[12] J. Pan, Y. T. Hou, L. Cai, Y. Shi, and S. X. Shen, "Topology control for wireless sensor networks," in *Proc. MobiCom*, 2003, pp. 286–299.

[13] Y. Xue, Y. Cui, and K. Nahrstedt, "A utility-based distributed maximum lifetime routing algorithm for wireless networks," in *Proc. 2nd Int. Conf. QoS Heterogeneous Wired/Wireless Networks*, 2005, p. 18.

[14] K. Kalpakis, K. Dasgupta, and P. Namjoshi, "Maximum lifetime data gathering and aggregation in wireless sensor networks," in *Proc. ICN*, Aug. 2002, pp. 685–696.

[15] S. Pattem, B. Krishnamachari, and Ramesh, "The impact of spatial correlation on routing with compression in wireless sensor networks," in *Proc. IPSN*, Berkeley, CA, 2004, pp. 28–35.

[16] R. Cristescu, B. Beferull-Lozano, and M. Vetterli, "On network correlated data gathering," in *Proc. IEEE INFOCOM*, Hong Kong, 2004, pp. 2571–2582.

[17] P. von Rickenbach and R. Wattenhofer, "Gathering correlated data in sensor networks," in *Proc. DIALM-POMC*, New York, 2004, pp. 60–66.

[18] M. Mauve, J. Widmer, and H. Hartenstein, "A survey on position-based routing in mobile ad hoc networks," *IEEE Network*, vol. 15, no. 6, pp. 30–39, Nov. 2001.

[19] L. Doherty, L. El Ghaoui, and K. S. J. Pister, "Convex position estimation in wireless sensor networks," in *Proc. IEEE INFOCOM*, Apr. 2001, pp. 1655–1663.

[20] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz, "Localization from mere connectivity," in *Proc. MobiCom*, Jun. 2003, pp. 201–212.

[21] C. Q. Hua and T. S. Yum, "Optimal routing and data aggregation for maximizing lifetime of wireless sensor networks," Inf. Eng. Dept., Chinese Univ. of Hong Kong, Tech. Rep., 2005.

[22] M. C. Vuran, O. B. Akan, and I. F. Akyildiz, "Spatio-temporal correlation: Theory and applications for wireless sensor networks," *Comput. Netw.*, vol. 45, no. 3, p. 245, 2004.

[23] M. C. Vuran and O. B. Akan, "Spatio-temporal characteristics of point and field sources in wireless sensor networks," in *Proc. IEEE ICC*, Istanbul, Turkey, Jun. 2006, pp. 234–239.

[24] R. K. Ahuja, "Algorithms for the minimax transportation problem," *Nav. Res. Logist.*, vol. 33, pp. 725–740, 1986.

[25] A. Ben-Tal and M. Teboulle, "A smoothing technique for nondifferentiable optimization problems," in *Proc. Int. Seminar Optimiz.*, New York, 1988, pp. 1–11.

[26] X. Li, "An entropy-based aggregate method for minmax optimization," *Eng. Optimiz.*, vol. 18, pp. 277–285, 1992.

[27] C. Chen and O. L. Mangasarian, "Smoothing methods for convex inequalities and linear complementarity problems," *Math. Program.*, vol. 71, no. 1, pp. 51–69, 1995.

[28] S. I. Birbil, S.-C. Fang, J. B. G. Frenk, and S. Zhang, "Recursive approximation of the high dimensional max function," *Oper. Res. Lett.*, vol. 33, pp. 450–458, 2005.

[29] L. Qi and D. Sun, "Smoothing functions and a smoothing newton method for complementarity and variational inequality problems," *J. Optimiz. Theory Appl.*, vol. 113, pp. 121–147, 2002.

[30] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Belmont, MA: Athena Scientific, 1997.

[31] R. G. Gallager, "A minimum delay routing algorithm using distributed computation," *IEEE Trans. Commun.*, vol. COM-25, no. 1, pp. 73–85, Jan. 1977.
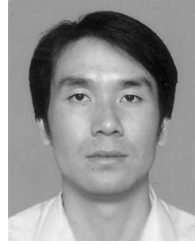
Cunqing Hua received the B.S. degree in electronics engineering from the University of Science and Technology of China in 2000, and the M.Phil. and Ph.D. degrees in information engineering from The Chinese University of Hong Kong in 2002 and 2006, respectively.

His research interests include protocol and algorithm design, analysis and optimization, and performance evaluation for computer networks, wireless mesh networks, community networks, and sensor networks.

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

He joined Bell Telephone Laboratories in April 1978, where he was involved with switching and signaling systems. In 1980, he accepted a teaching appointment at the National Chiao Tung University, Hsinchu, Taiwan, R.O.C. Then, in 1982, he joined the Chinese University of Hong Kong, where he is now Dean of Engineering and a 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 networks, lightwave networks, video distribution networks, and 3G networks. His recent interest is in RFIDs and sensor network technologies and applications.