Minimizing Energy Expense for Chain-Based Data Gathering in Wireless Sensor Networks*

Li-Hsing Yen, Ming-Zhou Cai, Yang-Min Cheng, and Ping-Yuan Yang
Department of Computer Science and Information Engineering
Chung Hua University, Taiwan 300, R.O.C.
E-mail: lhyen@chu.edu.tw

Abstract

This paper aims to minimize energy expense for chain-based data gathering schemes, which is essential to prolong operation lifetime of wireless sensor networks. We propose the concept of virtual chain, where an edge may correspond to a multi-hop data propagation path to conserve power. In contrast, an edge in previous work can only be a costly direct communication link. Furthermore, we propose a power-efficient leader scheduling scheme which selects the node that has the maximum residual power to be the leader of the chain. In contrast, nodes in previous work play the role of leader by turns, which results in non-uniform energy consumption among sensors. Simulation results show that our strategies successfully conserve power.

1. Introduction

Rapid progress in wireless communications and micro-sensing MEMS technology has enabled the deployment of wireless sensor networks. A wireless sensor network consists of a large number of sensor nodes deployed in a region of interest. Each sensor node is capable of collecting, storing, and processing environmental information, and communicating with other sensors.

Data gathering refers to the process of collecting sensed data from every sensor to a distant base station (BS), where end users can access the data [10]. Since sensor nodes are usually battery powered, power-conserving techniques are essential to prolong operation lifetime of sensor network. One such technique is data fusion [8], which is the process of automatic combining or aggregating sensed data. Another technique is multi-hop transmission, which replaces the otherwise direct transmission between every sensor and the BS. Multi-path transmission consumes less energy than corresponding direct transmission does since radio signal attenuation varies nonlinearly with distance [10].

To facilitate data fusion and multi-hop transmission, many existing data gathering approaches organize nodes into clusters [10, 9], a tree [1, 17], or a chain [13, 7]. Cluster-based approaches are inherently distributed, but they may not effectively minimize power dissipation [13]. Both tree-based and chain-based approaches have reported less energy consumption when compared with their cluster-based counterparts. Tree-based approaches allow simultaneous data transmissions so the data collection latency is expected to be low. However, simultaneous transmission requires sophisticated slot/code scheduling to prevent potential transmission collisions [1, 17]. In a chain-based scheme, nodes take turns in transmitting so simultaneous transmission is impossible. The transmission is thus collision free.

Once a chain has been formed, data are propagated from both ends of the chain toward a designated sensor node called *leader*. The leader then transmits the aggregated data directly to the BS. Energy expense in each round of data collection thus consists of two parts. One is for inter-sensor communication that depends on the structure of the chain. The other is for leader-BS communication that mainly depends on the in-between distance.

This paper aims to minimize energy expense for chain-based data gathering schemes. Finding an energy-optimal chain structure is similar to the traveling salesperson problem on a complete graph and thus NP-hard[1], so existing chain-construction algo-

* This work has been jointly supported by National Science Council, Taiwan, under contract NSC-93-2213-E-216-024 and by Chung Hua University under grant CHU-93-TR-004.

1 Not returning to the starting node (as a chain does) dose not

rithms [13, 7] take heuristic approaches. In these methods, every edge of the chain corresponds to a direct radio transmission between two nodes, which is simple but not an optimal approach in terms of energy use. In this paper, we propose the concept of virtual chain, where each edge of the chain corresponds to a multi-hop data propagation path to conserve power. In this way, the chain structure is independent of the actual data propagation paths among nodes: the topology superimposed by all data propagation paths is generally a graph rather than a chain.

The introduction of virtual chain is independent of chain construction methods: all existing methods can be used to form virtual chains. Nevertheless, we propose an additional chain construction scheme that converts a minimum spanning tree into a chain. This method yields more energy-efficient chain than PEGASIS does and has lower computational cost than [7].

When the BS is distant from sensors, leader node will dissipate more power then non-leader nodes. This would lead to early power exhaustion of the leader, decreasing network life time. As a remedy, nodes in previous work play the role of leader by turns. Unfortunately, such a round-robin leader scheduling still results in non-uniform energy consumption among sensors as their distances to the BS vary. We have formulated the problem of optimal leader scheduling as a linear programming problem and proposed a simple scheduling rule called Maximum Residual Power First (MRPF). As the name suggests, MRPF selects the node that has the maximum residual power to be the leader in each round of data collection. Simulation results show that MRPF performs only slightly worse than the optimal scheduling.

2. Problem Definition and Related Work

We are given a set of n sensor nodes that are assumed to be static (no mobility). A BS distant from these sensors is capable of communicating with them directly by radio. The BS is aware of all sensor's positions so that it can run a chain construction algorithm and broadcast the result to all sensors. Each sensor node is assumed to have power control capability so that minimum energy is expended to reach the intended recipients. A round of data collection is completed when all sensed data are sent to the BS. The leader in each round of data collection is selected by the BS.

change the computational complexity of the problem.

PEGASIS [13] uses a greedy algorithm for chain construction. The farthest node from BS is first added into the chain as a head. Then the node not in the chain but closest to the head is appended to the chain and becomes the new head. The process repeats until all nodes are included in the chain. This simple method has $O(n^2)$ time complexity, but the chains it produces are typically not power optimal. Du *et al.* [7] proposed an improved version of the chain construction algorithm. Unlike PEGASIS, where a non-chain node can only be appended to the end of the chain, in [7] the node can be considered inserting into any position within the chain to minimize the increase of energy use due to the addition of the node. In each round of the chain construction process, the node that increases energy to the minimum possible extent will be added into the chain. The constructed chains are generally power efficient, but the time complexity of this method is $O(n^3)$.

Virtual chains can be formally defined as follows. Consider two arbitrary nodes X and Y. Let a data propagation path starting at X and ending at Y be denoted by $P_{X,Y}$, which is a sequence of nodes $X = x_i$, $x_{i+1}, \ldots, x_j = Y$, where $j \geq i + 1$. The length of $P_{X,Y}$, $|P_{X,Y}|$, is defined to be the number of elements in $P_{X,Y}$ minus one. A sequence of n nodes $x_1, x_2, \ldots, x_n$ with $VP = \{P_{x_i,x_{i+1}} | 1 \leq i \leq n\}$ is a virtual chain if there exists some $P_{x_i,x_{i+1}} \in VP$ such that $|P_{x_i,x_{i+1}}| > 1$. It is a conventional chain otherwise.

We use the same model described in [10] to express energy dissipation caused by radio transmission, which has been commonly adopted [9, 13, 7, 17]. The radio dissipates $E_{elec} = 50$ nJ/bit to run the transmitter or receiver circuitry and the transmitter amplifier spends $\epsilon_{amp} = 100$ pJ/bit/m$^\alpha$ to achieve an acceptable signal-to-noise ratio, where α is the path exponent that indicates the rate at which the pass loss increases with distance. The value of α typically ranges from 2 to 4, depending on the characteristics of the communication environment [16]. All above-mentioned papers assume that $\alpha = 2$, which is the case in free space. We assume that $\alpha = 3$, which is typically the path loss exponent obtained in noisy urban area [16] and thus is more realistic.

We assume that data fusion is used so that every data message has k bits. It follows that if node x transmits a message to node y, x consumes energy $kE_{elec} + k\epsilon_{amp}d(x, y)^\alpha$, where $d(x, y)$ denotes the distance between x and y, while y expends kE_{elec}. The energy dissipation per transmission therefore consists of two parts. One part is of fixed quantity denoted by $\delta_k = 2kE_{elec}$. The other depends on α and on the distance between transmitter and receiver.

Given a data propagation path $X = x_i, x_{i+1}, \ldots, x_j = Y$, the cost of $P_{X,Y}$ is defined to be the total energy expense for propagating a k-bit message from X to Y, i.e.,

$$c(P_{X,Y}) = (j - i + 1)\delta_k + k\epsilon_{amp} \sum_{t=i}^{j-1} d(x_t, x_{t+1})^{\alpha}.$$

Let $\Phi(X, Y) = \{P_{X,Y}\}$ be the set of all possible data propagation paths from X to Y. Define $mcp(X, Y) = \{p | p \in \Phi(X, Y) \wedge \forall p' \in \Phi(X, Y) : c(p) \leq c(p')\}$ be the set of minimum-cost data propagation paths from X to Y. Given a virtual chain $\{N_i\}_{i=1}^{n}$ and associated data propagation path set $\{P_{N_i,N_{i+1}}\}_{i=1}^{n}$, the cost of the virtual chain is defined to be

$$\sum_{i=1}^{n-1} c(P_{N_i,N_{i+1}}).$$

The chain has the lowest cost if $P_{N_i,N_{i+1}} \in mcp(N_i, N_{i+1})$ for all i. The optimal virtual chain problem is to find a virtual chain whose lowest cost is the minimum among all possible ones. This is an NP-hard problem.

3. Proposed Scheme

Our energy-conservation approach for inter-sensor communications consists of two independent parts. The first is to compute and store the costs of every possible pair of nodes. Based on the cost information, the second part constructs a logical chain among all sensor nodes. The issue of leader scheduling is discussed in Sec. 3.3.

3.1. Costs of Node Pairs

Conventionally, the cost of every pair of nodes is simply the energy expense of a direct transmission between them [13, 7]. Let matrix M_d keep the cost such that $M_d(i, j)$ is the energy expense of a direct transmission between nodes i and j. To allow virtual chain, the costs should be associated with data propagation paths rather than direct links. Let M_p be the minimum cost matrix such that $M_p(i, j) = c(P_{i,j})$ for some $P_{i,j} \in mcp(i, j)$. Such a $P_{i,j}$ for every i and j can be found by running an all-pair shortest-path algorithm (e.g., Floyd-Warshall algorithm [4]) on input M_d. As an example, Fig. 1(a) represents M_d graphically for a four-sensor network, where each edge is labeled with the direct transmission cost between nodes on the two ends. Fig. 1(b) shows M_p that corresponds to all-pair shortest paths given M_d.

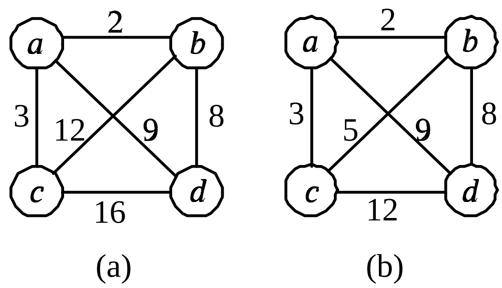

Figure 1. (a) M_d. (b) M_p that corresponds to the shortest paths given (a).

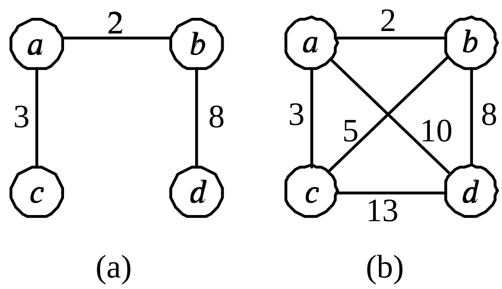

Figure 2. (a) MST of Fig. 1(a). (b) M_t that corresponds to (a).

All-pair shortest-path algorithms are time expensive ($O(n^3)$ in case of Floyd-Warshall algorithm). Alternatively, we may find first a minimum-cost spanning tree (MST) on the weighted complete graph corresponding to M_d. Then $P_{i,j}$ is designated to be the shortest path (actually the only path) traversing along the tree from i to j. We denote the matrix the keeps such costs by M_t. With this approach, the data propagation paths found may not be optimal. However, the time complexity of constructing an MST and traversing it from every node is only $O(n^2)$.

Take Fig. 1(a) as an example. Fig. 2(a) shows an MST of Fig. 1(a). M_t that corresponds to the MST is shown graphically in Fig. 2(b). Here $M_t(c, d) = 13$ because the data propagation path from c to d is confined to be that along the tree (i.e., c, a, b, d). Observe that this is not a minimum-cost path.

It is interesting and also important to note the property of triangle inequality in these cost matrices. Triangle inequality refers to the condition that the cost between any two nodes A and B must be at most the cost between A and any other node C plus the cost between C and B. Triangle inequality does not hold if M_d

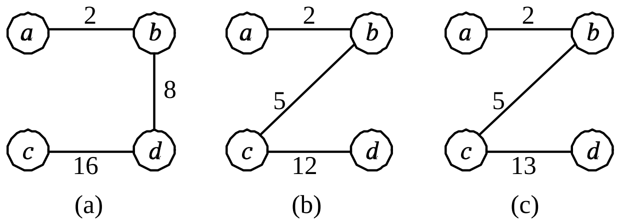

Figure 3. Various chains found by running PEGASIS on (a) M_d of Fig. 1(a); (b) M_p of Fig. 1(b); (c) M_t of Fig. 2(b).

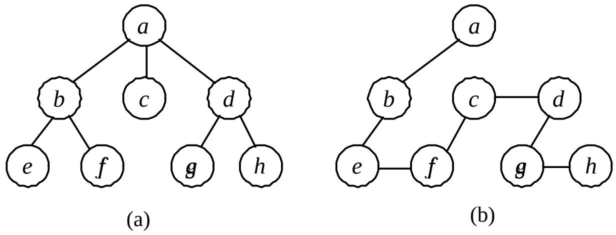

Figure 4. (a) A tree rooted at a. (b) The chain corresponds to the prefix traverse of (a).

is used as the cost matrix in our problem setting (due to the nonlinear attenuation properties of radio signals). That is, $M_d(i,j)$ can be larger than $M_d(i,k)+M_d(k,j)$ for any i, j, and k. Triangle inequality does hold in case of M_p, as it is a property of shortest paths [5]. For M_t that is computed based on an MST, triangle inequality still hold by the following theorem.

Theorem 1 *Let T_d be an MST built on the graph corresponding to M_d. If M_t is computed based on T_d, we have $M_t(i,j) \leq M_t(i,k) + M_t(k,j)$ for any i, j, and k.*

Proof: For any two nodes i and j in a tree, there exists exactly one unique simple path[2] from i to j. The path from i to k and then to j is either the same path from i to j, for which the equality of cost holds, or a path that is not simple. In the latter case, an edge incident with k must be included in the path twice, one immediately followed by the other (one coming into k and the other leaving k). If the occurrences of this edge are removed from the path, the path becomes either the exact simple path from i to j or a non-simple path with lower cost which can be further shrank by the above argument. The conclusion thus follows. □

3.2. Virtual Chain Formation

Once M_p (or M_t) and every $P_{i,j}$ have been obtained, a virtual chain can be formed using any conventional chain construction algorithm such as those proposed in [13, 7]. The only difference is that the algorithm may run on M_p or M_t instead of M_d. Fig. 3 shows various chains obtained by running the appending-based chain construction algorithm of PEGASIS [13] on different cost matrices.

Although the insertion-based chain construction algorithm [7] usually performs well, here we consider an MST-based chain construction heuristic which is more time efficient. The basic idea is to find an MST first (on the weighted complete graph representing M_d, M_t, or M_p) and then convert it to a chain. A tree can be converted to a chain by traversing the tree form the root in prefix order. The node visiting sequence then corresponds to a chain. Fig. 4 shows an example. Time complexity of this approach is $O(n^2)$.

This heuristic has been devised for the traveling salesperson problem (TSP), and is often accompanied with the assumption of triangle inequality. It can be shown that, thanks to the triangle inequality, the heuristic creates a TSP tour whose cost is not more than twice the cost of the MST [6]. The cost can be further reduced to at most 1.5 times of the minimum cost [3]. However, constant performance ratio is impossible without triangle inequality.

In summary, we have one design choice among three cost metrics and another design choice among three chain construction algorithms. Table 1 lists all possible combinations. Among them, the operations of MST-based chain constructions are detailed in Fig. 5. Procedure MST-MST can be further simplified by the following theorem.

Theorem 2 *Let T_d be an MST built on the graph corresponding to M_d. Assume that M_t is computed based on T_d. Let T_t be an MST on the graph corresponding to M_t. The cost of T_t is equal to that of T_d.*

Proof: For every edge $(i,j) \in T_t$, let $P_{i,j}$ denote the data propagation path from i to j that traverses along T_d. If $|P_{i,j}| = 1$, edge (i,j) must be an edge of T_d also. So if we can prove that $|P_{i,j}| = 1$ for every edge $(i,j) \in T_t$, the cost of T_t will be equal to that of T_d. Suppose, by way of contradiction, that there exists an edge $(i,j) \in T_t$ with $|P_{i,j}| > 1$. It follows that there is at least one intermediate node k on $P_{i,j}$. Since $P_{i,j}$ corresponds to the shortest path traversing along T_d from i to j, it must be a simple path. Therefore, for any k

2 A path is simple if it does not include the same edge twice [14].

Cost matrix	Chain construction		
	Greedy appending	Greedy insertion	MST traverse
M_d (direct transmission)	PEGASIS [13]	Direct-insertion [7]	Direct-MST
M_p (all-pair shortest paths)	Shortest-appending	Shortest-insertion	Shortest-MST
M_t (paths confined to MST)	MST-appending	MST-insertion	MST-MST

Table 1. All possible cost-metric/chain-construction combinations.

Direct-MST

1. Compute and store in M_d direct communication costs of all node pairs.
2. Find an MST T_d on M_d.
3. Convert T_d to a chain.

Shortest-MST

1. Compute and store in M_d direct communication costs of all node pairs.
2. Compute cost matrix M_p by running an all-pair shortest-path algorithm on M_d.
3. Find an MST T_p on M_p.
4. Convert T_p to a chain.

MST-MST

1. Compute and store in M_d direct communication costs of all node pairs.
2. Find an MST T_d on M_d.
3. Compute cost matrix M_t based on T_d.
4. Find an MST T_t on M_t.
5. Convert T_t to a chain.

Figure 5. Operations of MST-based chain constructions.

we have $M_t(i,k)+M_t(k,j) = M_t(i,j)$.[3] There are four possible cases depending on the relation among i, j, and k.

- Both edges (i,k) and (k,j) are included in T_t. This is impossible since the inclusion of these edges plus (i,j) creates a cycle in T_t.
- Edge (i,k) but (k,j) is included in T_t. We can form T'_t by first removing (i,j) from T_t and then adding (k,j) into T_t. Note that T'_t does not contain cycle and the cost of T'_t is lower than that of T_t since we swap (i,j) for a lower-cost edge (k,j). It follows that T'_t is a tree with cost lower than that of T_t.

3 Recall that the equality in Theorem 1 holds when k lies on the path from i to j.

MST-reduced

1. Compute and store in M_d direct communication costs of all node pairs.
2. Find an MST T_d on M_d.
3. Convert T_d to a chain.
4. For every edge (i,j) of the chain, set $P_{i,j}$ to be that specified by T_d.

Figure 6. Operations of MST-reduced.

- Edge (k,j) but (i,k) is included in T_t. Similarly, this leads to another tree whose cost is lower than that of T_t.
- Neither (i,k) nor (k,j) is included in T_t. T_t must contain a path from i to k and another from k to j as T_t is connected. The lengths of these paths must be greater than one. Now consider replacing (i,j) with (i,k) and (k,j) in T_t. Let the result be T'_t. Note that T'_t has the same cost as T_t but contains two cycles, one involving the path from i to k and the other j to k. We can remove any edge from the first path and any other from the second, resulting a tree with cost lower than that of T_t.

All these cases lead to impossibility or contradiction, so we conclude that there exists no edge $(i,j) \in T_t$ with $|P_{i,j}| > 1$. □

Theorem 2 indicates that, in case of MST-MST, we may directly convert T_d instead of T_t to a chain. Procedure MST-reduced in Fig. 6 thus replaces MST-MST.

Table 2 lists the time complexities of all mentioned methods. Among them, PEGASIS, Direct-MST, MST-appending, and MST-reduced are more time efficient then others.

3.3. Leader Scheduling

Given a chain structure, leader scheduling is to determine which node plays the role of leader in each round of data collection process. The goal is to prolong network lifetime, i.e., to maximize the number of

Method	Cost matrix computation	Chain construction	Overall
PEGASIS [13]	$O(n^2)$	$O(n^2)$	$O(n^2)$
Direct-insertion [7]	$O(n^2)$	$O(n^3)$	$O(n^3)$
Direct-MST	$O(n^2)$	$O(n^2)$	$O(n^2)$
Shortest-appending	$O(n^3)$	$O(n^2)$	$O(n^3)$
Shortest-insertion	$O(n^3)$	$O(n^3)$	$O(n^3)$
Shortest-MST	$O(n^3)$	$O(n^2)$	$O(n^3)$
MST-appending	$O(n^2)$	$O(n^2)$	$O(n^2)$
MST-insertion	$O(n^2)$	$O(n^3)$	$O(n^3)$
MST-reduced	$O(n^2)$	$O(n^2)$	$O(n^2)$

Table 2. Time complexities of all methods.

data-collection rounds. In the following, we analyze the maximum number of data-collection rounds that can be achieved before any node exhausts its power. Without loss of generality, we assume that nodes in the chain are numbered sequentially as $1, 2, \ldots, n$. Let e_i be the energy consumed by node i in transmitting a data message to the BS. Let $\rho_{i,j} = kE_{elec} + k\epsilon_{amp}d(i,j)^\alpha$ be the energy consumed by i and $e_r = kE_{elec}$ be that consumed by j when node i transmits a k-bit message to node j.

When some node i is selected to be the leader, every node numbered $j < i$ (if any) expends energy $\rho_{j,j+1}$ in sending data to node $j+1$, at which energy e_r is consumed to receive the data. Likewise, every node numbered $k > i$ (if any) expends $\rho_{k,k-1}$ to send data to node $k-1$, where energy e_r is expended in receiving the data. The leader transmits the collected data to the BS, consuming energy e_i. Suppose that every node i is scheduled to be the leader x_i times, Table 3 shows the energy expense of every sensor node. Optimal leader scheduling problem is to find positive integer values of x_i's as to maximize $\sum_i x_i$ subject to the following constraints:

$$\begin{aligned}
E_1 &\geq (e_1+e_r)x_1 + \rho_{1,2}x_2 + \rho_{1,2}x_3 + \cdots + \rho_{1,2}x_n, \\
&\vdots \\
E_i &\geq (\rho_{i,i-1}+e_r)x_1 + \cdots + (\rho_{i,i-1}+e_r)x_{i-1} \\
&\quad +(e_i+2e_r)x_i + (\rho_{i,i+1}+e_r)x_{i+1} + \cdots + \\
&\quad (\rho_{i,i+1}+e_r)x_n, \\
&\vdots \\
E_n &\geq \rho_{n,n-1}x_1 + \rho_{n,n-1}x_2 + \cdots + (e_n+e_r)x_n,
\end{aligned}$$

where E_i denotes the amount of energy that node i initially has. These constraints can be reformulated as

$$\mathbf{A}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} \leq \begin{pmatrix} E_1 \\ E_2 \\ E_3 \\ \vdots \\ E_n \end{pmatrix},$$

where

$$\mathbf{A} = \begin{pmatrix} e_1+e_r & \rho_{1,2} & \cdots & \rho_{1,2} \\ \rho_{2,1}+e_r & e_2+2e_r & \cdots & \rho_{2,3}+e_r \\ \rho_{3,2}+e_r & \rho_{3,2}+e_r & \cdots & \rho_{3,4}+e_r \\ \vdots & \vdots & \cdots & \vdots \\ \rho_{n,n-1} & \rho_{n,n-1} & \cdots & e_n+e_r \end{pmatrix}.$$

The problem turns out to be a linear programming problem. Round robin leader scheduling (RR) equalizes the values of x_i's, which is generally far from optimal. The authors of PEGASIS also proposed an improvement on RR [13]. This approach sets up a threshold of distance, and nodes are not allowed to be leaders if their distances to their neighbors along the chain are beyond the threshold.

Instead of finding an optimal solution, we propose a simple rule called Maximum Residual Power First (MRPF) for leader selection. As the name suggests, MRPF selects the node that has the maximum residual power to be the leader in each round of data collection. Residual power information can be piggybacked with data messages as a part of the aggregated data. If every node attaches its own power level to data message and let the BS find the maximum value, it will incur an additional $O(n)$ overhead on every message. A better approach is to let every node compare its power level with that attached with incoming data message (if any) and send only the large one. This is similar to existing distributed maximum-finding algorithms on rings [2, 11, 12, 15] and the message overhead is only $O(1)$.

Node id.	In sending messages to the BS	In sending messages to neighbors	In receiving neighbor's messages
1	$e_1 x_1$	$\rho_{1,2} \sum_{j=2}^{n} x_j$	$e_r x_1$
$i \in \{2, \ldots, n-1\}$	$e_i x_i$	$\rho_{i,i-1} \sum_{j=1}^{i-1} x_j + \rho_{i,i+1} \sum_{j=i+1}^{n} x_j$	$e_r(\sum_{j=1}^{i-1} x_j + \sum_{j=i+1}^{n} x_j + 2x_i)$
n	$e_n x_n$	$\rho_{n,n-1} \sum_{j=1}^{n-1} x_j$	$e_r x_n$
x_i: the number of times node i is selected to be the leader; e_i: the amount of energy consumed in transmitting message from node i to the BS; $\rho_{i,j}$: the energy consumed by i in transmitting a message to j; e_r: the energy consumed by any node in receiving a message.			

Table 3. Energy expense of every sensor.

4. Simulations

We conducted simulations to analyze the performances of energy conservation techniques. In all experiments, message size is assumed to be 2000 bits. The positions of sensor nodes are randomly determined by a uniform distribution over the network region.

4.1. Performance of Virtual Chains

We measured the number of data-collection rounds that can be achieved by each approach. Two network sizes are considered: 50×50 and 100×100. The number of nodes are varied as 50, 100, and 200. The BS is located at (50,150), (50,200), or (50,300). The initial power of each sensor is 50J. Figs. 7-9 show the results averaged over 100 experiments. The results of Direct-insertion, Shortest-appending, Shortest-MST, MST-insertion, and MST-reduced are nearly the same and are collectively denoted as 'others' in these figures. We can see that Direct-MST performs better than PEGASIS but worse than others. Also note that the difference of performances increases when the network size increases. This can be explained as when the network is bigger, the average distance between nodes increases and thus enlarges the performance difference between good and bad schemes.

4.2. Performances of Leader Scheduling

We measured and compared the performance gains brought by several leader scheduling schemes including MRPF, RR, and RR with distance-based leader eligibility rule. Here the network size is 50×50 and BS is located at $(25, 150)$ or $(25, 250)$. All nodes are assumed to have power 10J initially. The chains to be tested with leader scheduling schemes were produced by PEGASIS. Fig. 10 shows the results, where each result was obtained by averaging 10 experiments.

It is clear that MRPF performs slightly worse than the optimal result obtained by a linear-programming problem solver. RR generally performs much worse than MRPF. The performance of RR with distance-based leader eligibility rule (RR with threshold) decreases as the threshold value of distance decreases. The reason is that the loads on leader nodes cannot be fairly shared if less nodes are eligible for leaders. We also found that only when the number of sensors is sufficiently large, RR with threshold outperforms RR (results are not shown here). Therefore, a critical is-

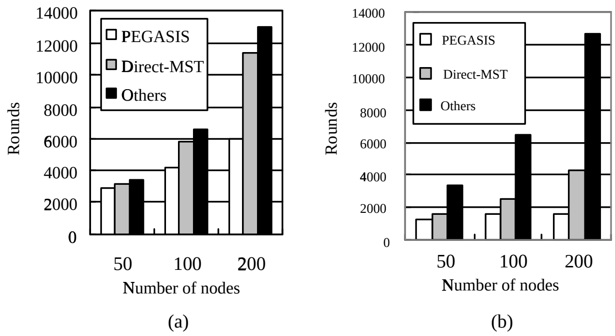

Figure 7. Number of rounds before 1st node exhausts its power in a (a) 50×50 network (b) 100×100 network. The BS is located at (50,150).

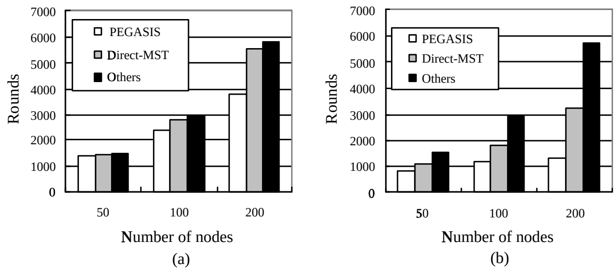

Figure 8. Number of rounds before 1st node exhausts its power in a (a) 50×50 network (b) 100×100 network. The BS is located at (50,200).

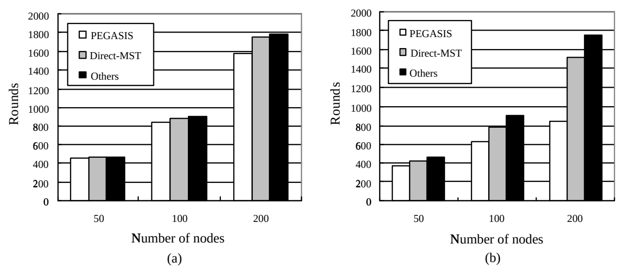

Figure 9. Number of rounds before 1st node exhausts its power in a (a) 50×50 network (b) 100×100 network. The BS is located at (50,300).

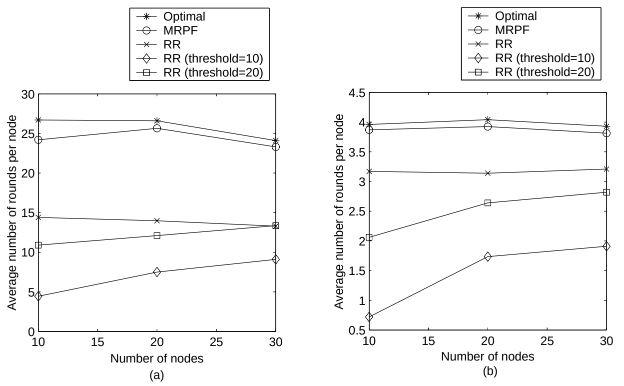

Figure 10. Number of rounds before 1st node exhausts its power. (a) The BS is located at $(25, 150)$. (b) The BS is located at $(25, 250)$.

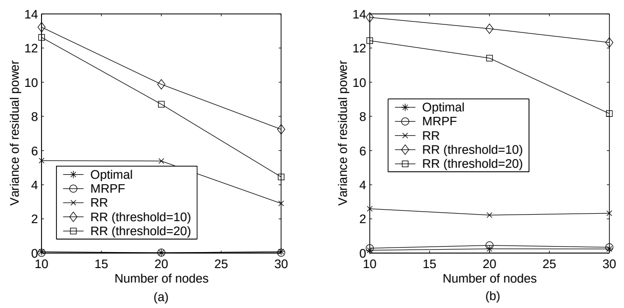

Figure 11. The variance of residual power when the 1st node exhausts its power. (a) The BS is located at $(25, 150)$. (b) The BS is located at $(25, 250)$.

sue of using RR with threshold is to determine an appropriate threshold value so that leader-eligible nodes and others fairly share the communication load, which is untold in the original paper.

Fig. 11 shows the variance of all other's residual power when the first node exhausts its power. We can see that both the optimal and MRPF have very small variances, meaning that they successfully equalize power consumption among all nodes. RR family does not perform well, but the results tend to be acceptable when the population of nodes is getting large.

5. Conclusions

We have considered several energy-conserving techniques for chain-based data gathering. Among them, PEGASIS, Direct-MST, MST-appending, and MST-reduced all have $O(n^2)$ computation time while others have $O(n^3)$. On the other hand, Direct-insertion, Shortest-appending, Shortest-MST, MST-insertion, and MST-reduced perform nearly the same and outperform others. MST-appending and MST-reduced both have the merits of lower time cost and, in the same time, better results and are therefore recommended.

We have cast optimal leader scheduling as a linear programming problem. The proposed leader scheduling algorithm, MRPF, successfully equalizes energy expense among all sensors. Experiment results show that its performance is nearly the same as the optimal scheduling.

References

[1] V. Annamalai, S. K. S. Gupta, and L. Schwiebert. On tree-based convergecasting in wireless sensor networks. In *IEEE Wireless Communications and Networking Conf.*, Mar. 2003.

[2] E. Chang and R. Roberts. An improved algorithm for decentralized extrema-finding in circular configurations of processes. *Comm. ACM*, 22(5):281–283, May 1979.

[3] N. Christodes. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical Report 388, Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, PA, 1976.

[4] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*, pages 558–562. MIT Press, 1990.

[5] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*, page 520. MIT Press, 1990.

[6] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*, pages 969–972. MIT Press, 1990.

[7] K. Du, J. Wu, and D. Zhou. Chain-based protocols for data broadcasting and gathering in the sensor networks.

In *Proc. Int'l Parallel and Distributed Processing Symp.*, Apr. 2003.

[8] D. Hall. *Mathematical Techniques in Multisensor Data Fusion.* Artech House, Boston, MA, 1992.

[9] M. Handy, M. Haase, and D. Timmermann. Low energy adaptive clustering hierarchy with deterministic cluster-head selection. In *Proc. 4th Int'l Workshop on Mobile and Wireless Communications Network*, pages 368–372, Sept. 2002.

[10] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proc. 33rd Annual Hawaii Int'l Conf. on System Sciences*, Jan. 2000.

[11] L. Higham and T. Przytycka. A simple, efficient algorithm for maximum finding on rings. In A. Schiper, editor, *Distributed Algorithms (7th International Workshop, WDAG '93)*, pages 249–263, Lausanne, Switerland, September 1993. Springer-Verlag. Also published in *Lecture Notes in Computer Science*, 725.

[12] D. S. Hirschberg and J. B. Sinclair. Decentralized extrema-finding in circular configurations of processes. *Comm. ACM*, 23(11):627–628, Nov. 1980.

[13] S. Lindsey, C. Raghavendra, and K. M. Sivalingam. Data gathering algorithms in sensor networks using energy metrics. *IEEE Trans. Parallel Distrib. Syst.*, 13(9):924–935, Sept. 2002.

[14] C. L. Liu. *Elements of Discrete Mathematics*, page 145. McGraw-Hill Inc., 2nd edition, 1985.

[15] G. L. Perterson. An $O(n \log n)$ unidirectional distributed algorithm for the circular extrema problem. *ACM Trans. on Programming Languages and Systems*, 4(4):758–762, Oct. 1982.

[16] T. S. Rappaport. *Wireless Communications: Principles and Practice.* Prentice Hall, 2nd edition, 2002.

[17] S. Upadhyayula, V. Annamalai, and S. K. S. Gupta. A low-latency and energy-efficient algorithm for convergecast in wireless sensor networks. In *Proc. Globecom*, San Fransisco, Dec. 2003.