

Research Article

Energy-Balanced Separating Algorithm for Cluster-Based Data Aggregation in Wireless Sensor Networks

Shuai Fu,¹ Jianfeng Ma,¹ Hongtao Li,¹ and Changguang Wang²

¹ School of Computer Science and Technology, Xidian University, Xi'an 710071, China

² College of Information Technology, Hebei Normal University, Shijiazhuang 050016, China

Correspondence should be addressed to Shuai Fu; ljhs0803@126.com

Received 23 September 2012; Revised 16 December 2012; Accepted 22 December 2012

Academic Editor: Jiman Hong

Copyright © 2013 Shuai Fu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Clustering provides an effective way to prolong the lifetime of wireless sensor networks. However, the head node may die much faster than other nodes due to its overburden. In this paper, we design a method to mitigate the uneven energy dissipation problem. Considering the relaying load undertaken by each cluster, we use the network topology and energy consumption to calculate a cluster radius for obtaining the intercluster energy balancing. A new cluster-leader election algorithm is proposed wherein the task of a single cluster head is separated to two nodes so that the critical nodes in each cluster will not exhaust their power so quickly. Furthermore, cross-level data transmission is used to prolong network lifetime. Extensive simulation experiments are carried out to evaluate the method with several performance criteria. Our simulation results show that this method obtains satisfactory performance on balancing energy dissipation and prolonging the networks lifetime.

1. Introduction

Continued advances of microelectromechanical systems (MEMS) and wireless communication technologies have enabled the deployment of large-scale wireless sensor networks (WSNs) [1]. Due to limited and nonrechargeable energy provision of sensors, improving energy efficiency and maximizing the network lifetime by decreasing energy consumption of the individual nodes and balancing energy consumption of all nodes are the major challenges in the research of data aggregation algorithms in WSNs [2].

The energy of a sensor node is mainly consumed by the communication unit, computing unit, and sensor, out of which the wireless transceiver uses a large portion of the energy. The traffic follows a multihop pattern, where intermediate nodes deplete their energy faster when taking more tasks, which leads to what is known as an energy hole [3]. Therefore, unbalanced energy consumption is an inherent problem, which needs to be solved to prolong the network lifetime.

Clustering method for data aggregation in wireless sensor network has attracted great attention for its high efficiency [4–6]. The data traffic (as well as the data transmission

and reception energy) can be greatly reduced by applying data aggregation at cluster heads. It significantly reduces the battery drainage of individual sensors and also has other advantages in terms of simplifying network management, improving security, and achieving better scalability. Recently, some studies have been done to address issues related to energy efficiency and prolonging the lifetime [7] of the WSNs. In this work, our main focus is rather on balancing the energy dissipation of the whole network and making energy-efficient routing during data aggregation. It is considered from two aspects: intercluster energy balancing and intracluster energy balancing. The motivation and main contributions of this paper are listed in the following.

An analysis of energy balancing problem is made in WSNs under cluster hierarchy. This problem deals with both intercluster and intracluster. As to the former, we try to allow each cluster to consume approximately the same amount of energy through arranging cluster sizes. As for the latter, we design an algorithm from the task separation perspective. Through this approach, the load imposed on a single cluster head can be alleviated in each cluster. Although a lot of literatures on dividing the network into clusters cope with the problem of unbalanced power consumption in WSNs,

none of the existing algorithms consider assigning the tasks of CHs to two nodes for intracluster energy balancing. The main contributions of this paper are summarized as follows.

- (i) Arranging cluster sizes based on the equal inter-cluster energy consumption. According to the relaying load of each cluster, the cluster radius is calculated. We only consider the energy consumption on data aggregation and transmission. It is assumed that the total energy consumed by each cluster is approximately the same. As leaders of clusters near the BS will relay more data than those located far away from the BS, their radiuses will be smaller accordingly.
- (ii) Designing the intracluster communication algorithm from the task separation perspective. To slow the energy consumption of critical nodes in each cluster, the algorithm is designed with consideration of task separation. Both the data gathering and aggregation are performed by a sensor named processor, and the report to the base station will be done by another sensor named forwarder in the same cluster. The election procedure of the processor and the forwarder is performed simultaneously, thus avoiding wasting the bandwidth caused by transmitting messages too many times.

The remainder of this paper is organized as follows. Section 2 summarizes related work. Section 3 describes the network model and elaborates the imbalanced energy consumption problem that we address in this work. In Section 4, the proposed method for arranging cluster size is described in detail. Section 5 shows the new algorithm for cluster-leader election. Section 6 gives a performance analysis of the proposed algorithm. We make a theoretical energy consumption analysis in Section 7. Then in Section 8, we evaluate the performance of our approach by simulation and make a comparison of it with LEACH, MR-LEACH, EECA, and ACT. Finally, we conclude the paper in Section 9.

2. Related Work

In this section, four steps of the hierarchical routing protocol and related works are introduced: CH election, cluster formation, intracluster communications, and intercluster communications.

In CH election, many typical protocols adopt different approaches. The first proposed cluster-based algorithm for WSNs is LEACH [8]. It divides the operation into rounds and randomly selects new CHs in each round to distribute the energy load among all nodes. In the data transmission phase, each cluster head forwards an aggregated packet to the base station directly. One common issue with LEACH is that the energy of sensor nodes which are located far away from their CHs can be easily used up for the transmission of packets to their CHs.

Several variants of LEACH protocol are proposed to make an improvement on it that further decreases the power consumption. LEACH-C [8] is a centralized version of LEACH.

It uses the BS central control to form clusters. During the set-up phase, each node sends information about its current location and energy level to the BS. Then the BS computes the average node energy and chooses those whose energy level is above this average as candidates for CHs. For minimizing the total sum of squared distances between all the non-CHs and the closest CH, LEACH-C uses the simulated annealing algorithm to find the optimal clusters. Fan and Song [9] introduce the energy-LEACH protocol (E-LEACH), which chooses CHs based on remaining energy. Multihop communication mode among cluster heads is adopted to avoid the whole network from dying quickly and prolong the network lifetime. Loscri et al. [10] build a two-level hierarchy for LEACH (TL-LEACH). TL-LEACH considers a randomized rotation of the CHs and chooses one of the CHs that lies between the current CH and the BS as a relay station. This allows CHs to better distribute the energy load among sensor nodes when the network density is higher. Yassein et al. [11] present the concept of vice-CH (V-LEACH), a sensor node which will become a CH if the current CH uses up its energy. This ensures that cluster nodes data will always reach the BS. Farooq et al. [12] present a multihop routing with low energy adaptive clustering hierarchy (MR-LEACH) protocol. The CH election in MR-LEACH is based on the available energy, and it partitions the network into different layers of clusters. CHs in each layer are responsible for relaying data for CHs at lower layers to transmit data to the BS. Thus, MR-LEACH follows multihop routing from cluster heads to the BS to conserve energy.

During cluster formation, CHs broadcast messages and non-CH nodes determine which cluster to join according to the signal strength received. As all nodes tend to join the closest cluster, clusters are formed in various sizes. The greater the cluster is, the heavier the load of its CH is. An energy-efficient clustering scheme (EECS) [13] presented by Ye et al. takes into account the unbalanced energy dissipation. In EECS, during the cluster formation phase nodes decide to associate with a CH based on a weighted cost factor that is composed of three functions. A new scheme was given to avoid the energy hole problem with unequal clustering mechanism in [14]. Its core is an energy-efficient uneven clustering algorithm (UCR) for network topology organization, in which tentative cluster heads use uneven competition ranges to construct clusters of uneven sizes. Some other works attempt to take measures to adjust the size of each cluster so as to reduce the differences of loads between CHs [15, 16]. Paper [17] proposed a novel cluster-based routing protocol named ACT, which aims to reduce the size of clusters near the base station. It provides a method to arrange cluster size, allowing each CH to consume approximately the same amount of energy. However, the CHs are determined as soon as the cluster radius is obtained. Their locations are closest to the ideal but may not be the best.

As for intracluster communications, some studies suggest that the sleep mode of sensor nodes should be adopted in intracluster communications to save energy. That means there is only one node or several nodes in a cluster that are active while the others enter sleep mode (e.g., cluster members take turns collecting data). However, scheduling sleep time is a major issue worthy of discussion [18].

During intercluster communications, the farther the messages to be transmitted, the greater the energy dissipation will be. In [19], the authors proposed an energy-efficient clustering algorithm (EECA) in which the data aggregation tree is constructed by determining the weight of CHs, but this many-to-one communication mode still possesses the imbalance power dissipation problem. Distributed clustering algorithms were proposed in [20], with the objective of minimizing the energy spent in communicating information to the sink. It should be noted that minimizing the total energy consumption is not equivalent to maximizing coverage time, as the former criterion does not guarantee balanced power consumption at various CHs.

Unlike previous approaches, we try to solve the unbalanced energy consumption problem from the perspective of both intercluster communication and intracluster communication. Arranging cluster radius based on the assumption of equal total energy dissipation ensures the energy balance among clusters, while the new separating cluster-based algorithm (SCA) obtains the energy balance among sensors within a cluster. The separation of the CH role alleviates the burden of a single CH, thus avoiding early network collapse due to the death of critical nodes and prolonging the network lifetime.

3. Preliminaries

3.1. Network Model. In this paper, we consider a sensor network consisting of N sensor nodes uniformly dispersed in the service area of the network whose coverage area is a rectangular region of $L \times W$. We make some assumptions about the sensor nodes and the underlying network model.

- (1) The positions of BS and sensor nodes are fixed. Nodes are uniformly distributed in the sensor field with density ρ .
- (2) Each node is assigned a unique identifier (ID). Sensors are with the same initial energy and their transmit power is controllable. The maximum power level can be used in transmitting data to BS directly.
- (3) Links are symmetric. A node can compute the approximate distance to another node based on the received signal strength, if the transmitting power is known.
- (4) Sensor nodes can recognize their geographical position and the BS's position via exchanging information.
- (5) All sensors are sensing the environment at the same fixed rate and thus always have data to send to the end-user. The size of each data packet is the same.

We use the typical energy consumption model [8]. The energy spent for transmitting an l -bit message over distance d is

$$E_{TX}(l, d) = \begin{cases} l \times E_{elec} + l \times \epsilon_{fs} \times d^2, & d < d_0, \\ l \times E_{elec} + l \times \epsilon_{amp} \times d^4, & d \geq d_0, \end{cases} \quad (1)$$

where E_{elec} is the energy dissipated per bit to run the transmitter or the receiver circuit, ϵ_{fs} and ϵ_{amp} are the energy dissipated per bit to run the transmit amplifier depending on the distance between the transmitter and receiver. If the distance is less than a threshold d_0 , the free space (fs) model is used; otherwise, the multipath (mp) model is used.

To receive this message, the expended energy is

$$E_{RX}(l) = l \times E_{elec}. \quad (2)$$

The consumed energy of aggregating a message with l -bit is

$$E_A(m, l) = l \times E_{DA}, \quad (3)$$

where E_{DA} is the energy dissipated per bit to aggregate message signal.

3.2. Related Definition

- (1) We denote the i th sensor by S_i and the corresponding sensor node set $S = \{S_1, S_2, \dots, S_N\}$, where $|S| = N$. For a random node S_i , make its residual energy E_{ri} and its coordinate $L(X_i, Y_i)$.
- (2) The neighboring node set R_{CH} of any node S_m is defined as

$$\begin{aligned} S_m - R_{CH} &= \{S_n \mid d(S_m, S_n) \leq \theta R_m, d(S_n, BS) < d(S_m, BS)\}, \end{aligned} \quad (4)$$

where θ is the minimum integer that lets $S_m - R_{CH}$ contain at least one item (if there does not exist such a θ , define $S_m - R_{CH}$ as a null).

- (3) Define E_{res_MAX} as the threshold of the residual energy of node S_i . If a node's residual energy is less than E_{res_MAX} , it will give up the competition for processor and forwarder. According to (1)–(3), the value of E_{res_MAX} could be estimated by

$$\begin{aligned} E_{res_MAX} &= \mu \times [mE_{elec} + (m+1)E_{DA} \\ &\quad + (1-\lambda)(m+1)l(E_{elec} + \epsilon_{fs}d^2)], \end{aligned} \quad (5)$$

where μ represents the number of times of each turn of data acquisition, λ is the compression ratio of data aggregation, d represents the distance between current processor and its parent node, and m represents the number of neighbor nodes.

3.3. Problem Statement. A fundamental issue in WSN is maximizing the network lifetime subject to a given energy constraint. Notice that the BS is usually located far away from the monitoring area. Previous research has shown that multihop intercluster communication mode is usually desirable because of its power-consumption advantage over direct (CH-to-sink) communication (e.g., [21]). However, the energy hole situation is essentially caused because of the

different loads among nodes when the multihop forwarding mode is adopted in intercluster communication.

Based on the given model, balancing the energy consumption to the maximum is our optimization objective. A complete data collection process involves two steps: collecting data from all sensor nodes and delivering the data to the BS. This problem can be formulated as follows:

$$\begin{aligned} & \min \sum_{i \in C_j} \left[E_{ri} - \overline{E_{res}^j} \right]^2 \\ & \min \sum_{k \in M} \left[\overline{E_{ri}(k)} - \overline{E_{res}} \right]^2, \end{aligned} \quad (6)$$

where E_{ri} is the residual energy of node i in cluster j and $\overline{E_{res}^j}$ represents the average remaining energy of nodes in cluster j . $\overline{E_{ri}(k)}$ represents the average remaining energy of all nodes in cluster k , and $\overline{E_{res}}$ is the average remaining energy of all sensor nodes.

The above optimization problem can be solved as two subproblems as follows.

- (a) How to balance the energy dissipation among nodes within the same cluster? This is referred to as the problem of intracluster energy consumption balancing.
- (b) How to balance energy dissipation among different clusters? This is referred to as the problem of intercluster energy consumption balancing.

It will be described in detail in the following two sections.

To enable readers to more easily understand this paper, Table 1 summarizes the notations used in this paper.

4. Intercluster Energy Balancing

4.1. Arranging Cluster Radius. In the proposed algorithm, we hope to balance the energy consumption between clusters, and this can be achieved by applying (1) and (3) to calculate the radius of each cluster. It is supposed that the tentative network consists of clusters with M different sizes, and each cluster member passes one bit of data to cluster leaders (see Figure 1). The transmission range is regarded as the distance between the centers of two clusters for simplicity in calculations, except in the 1st level (i.e., $(r_m + r_{m-1})$ in M th level, $(r_{m-1} + r_{m-2})$ in $(M - 1)$ th level, and so forth).

We assume that the nodes are deployed in each cluster with density ρ . As each cluster leader in the outermost level (M th level) does not need considering the relay data, it only takes care of the data transmitted by its own cluster members. Its transmission range is $(r_m + r_{m-1})$, and thus the total energy dissipation of each cluster leader in M th level is

$$\pi r_m^2 \rho E_{DA} + (1 - \lambda) \pi r_m^2 \rho \left[E_{elec} + \varepsilon_{fs}(r_m + r_{m-1})^2 \right], \quad (7)$$

where the first part represents the aggregation energy consumption in the M th level and the second the transmission energy consumption from the M th level to the $(M - 1)$ th level.

However, cluster leaders in the $(M - 1)$ th level not only process data given by their members, but they also perform

TABLE 1: Notations.

Symbols	Description
N	The number of sensor nodes
A	The area of the sensing field and $A = L \times W$
ρ	The node's density
E_{elec}	The energy dissipated per bit to run the transmitter or the receiver circuit
ε_{fs}	The energy dissipated per bit to run the transmit amplifier when the distance between the transmitter and receiver is less than a threshold
ε_{amp}	The energy dissipated per bit to run the transmit amplifier when the distance between the transmitter and receiver is greater than a threshold
d_0	The threshold distance determining which model to use to calculate the energy consumption of nodes
E_{DA}	The energy dissipated per bit to aggregate message signal
E_{ri}	The residual energy of a node i
E_{res_MAX}	The threshold of residual energy of a node
λ	The compression ratio of data aggregation
m	The number of neighbor nodes
M	The number of different sizes of clusters
r_j	The radius of cluster j
CP_j	The competition bids of being elected as processor for node S_j
CF_j	The competition bids of being elected as forwarder for node S_j
ΔE	The difference value between residual energy of two different clusters
ΔE_T	The difference value between the energy consumption for sending the same data packets to two different clusters

data relaying for M th level. According to its transmission range, the total energy dissipation of cluster leaders in $(M - 1)$ th level is

$$\begin{aligned} & \pi r_{m-1}^2 \rho E_{DA} + \left[(1 - \lambda) \pi r_{m-1}^2 \rho + (1 - \lambda)^2 \pi r_m^2 \rho \right] \\ & \times \left[E_{elec} + \varepsilon_{fs}(r_{m-1} + r_{m-2})^2 \right]. \end{aligned} \quad (8)$$

Similarly, each cluster leader in the $(M - 2)$ th level forwards data generated by its own cluster members while performing data relaying for $(M - 1)$ th level and M th level. Then the total energy dissipation of each cluster leader in $(M - 2)$ th level is

$$\begin{aligned} & \pi r_{m-2}^2 \rho E_{DA} + \left[(1 - \lambda) \pi r_{m-2}^2 \rho + (1 - \lambda)^2 \pi r_{m-1}^2 \rho + (1 - \lambda)^3 \right. \\ & \left. \times \pi r_m^2 \rho \right] \left[E_{elec} + \varepsilon_{fs}(r_{m-2} + r_{m-3})^2 \right]. \end{aligned} \quad (9)$$

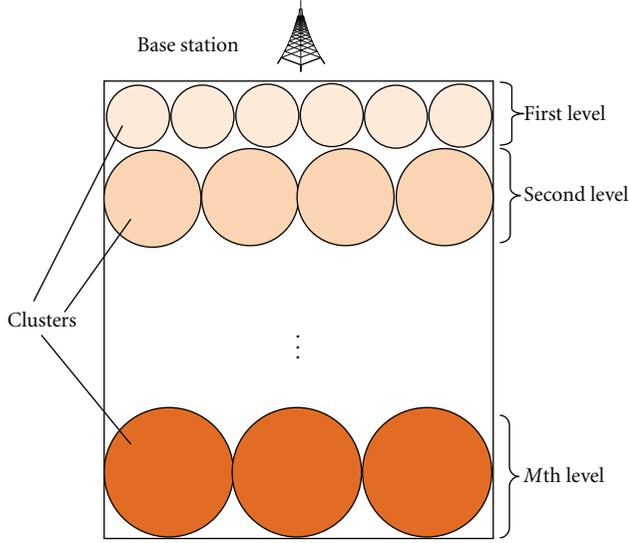


FIGURE 1: Level model of network topology.

In this way, the total energy dissipation of a cluster leader in each level (for one generated message bit) can be calculated as follows:

$$\begin{aligned}
 \text{The } M\text{th level: } E_m &= \pi r_m^2 \rho E_{DA} + (1 - \lambda) \pi r_m^2 \rho \\
 &\quad \times [E_{\text{elec}} + \varepsilon_{fs}(r_m + r_{m-1})^2] \\
 \text{The } (M-1)\text{th level: } E_{m-1} &= \pi r_{m-1}^2 \rho E_{DA} \\
 &\quad + [(1 - \lambda) \pi r_{m-1}^2 \rho \\
 &\quad + (1 - \lambda)^2 \pi r_m^2 \rho] \\
 &\quad \times [E_{\text{elec}} + \varepsilon_{fs}(r_{m-1} + r_{m-2})^2] \\
 &\quad \vdots \\
 \text{The 2nd level: } E_2 &= \pi r_2^2 \rho E_{DA} \\
 &\quad + [(1 - \lambda) \pi r_2^2 \rho + (1 - \lambda)^2 \pi r_3^2 \rho + \dots \\
 &\quad + (1 - \lambda)^{m-1} \pi r_m^2 \rho] \\
 &\quad \times [E_{\text{elec}} + \varepsilon_{fs}(r_2 + r_1)^2] \\
 \text{The 1st level: } E_1 &= \pi r_1^2 \rho E_{DA} + [(1 - \lambda) \pi r_1^2 \rho + (1 - \lambda)^2 \\
 &\quad \times \pi r_2^2 \rho + \dots \\
 &\quad + (1 - \lambda)^m \pi r_m^2 \rho] \\
 &\quad \times [E_{\text{elec}} + \varepsilon_{fs} r^2], \tag{10}
 \end{aligned}$$

where r_1, r_2, \dots, r_m are cluster radiuses (in M different sizes), respectively, and r is used for calculating the transmission

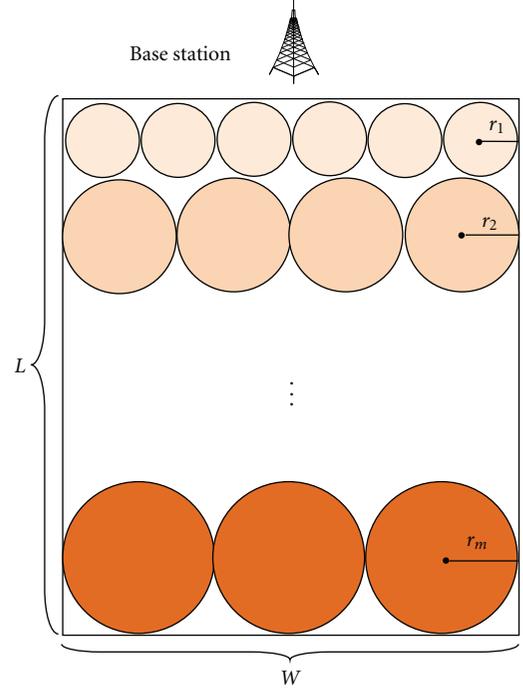


FIGURE 2: Calculation of cluster radius.

range in the 1st level, which we explain in (15). Here, E_i is the energy consumed on each cluster leader in i th level.

Because we assume that the energy consumption of cluster leaders in each level is similar, (11) is applied to calculate cluster radius in each level:

$$\begin{aligned}
 E_1 &\cong E_2 \cong \dots \cong E_m, \\
 r_1 + r_2 + \dots + r_m &= \frac{L}{2}, \tag{11}
 \end{aligned}$$

where L is the length of sensing area (see Figure 2).

4.2. A Numerical Example. It is assumed that a BS wants to construct four clusters of different sizes, and it lets $M = 4$. The ratio of $r_1, r_2, r_3,$ and r_4 can be obtained by (12a). Then we put the obtained ratio in (12b) to calculate the actual cluster radius:

$$\begin{aligned}
 \text{The 4th level: } E_4 &= \pi r_4^2 \rho E_{DA} + (1 - \lambda) \pi r_4^2 \rho \\
 &\quad \times [E_{\text{elec}} + \varepsilon_{fs}(r_4 + r_3)^2]
 \end{aligned}$$

$$\begin{aligned}
 \text{The 3rd level: } E_3 &= \pi r_3^2 \rho E_{DA} \\
 &\quad + [(1 - \lambda) \pi r_3^2 \rho + (1 - \lambda)^2 \pi r_4^2 \rho] \\
 &\quad \times [E_{\text{elec}} + \varepsilon_{fs}(r_3 + r_2)^2]
 \end{aligned}$$

$$\begin{aligned}
\text{The 2nd level: } E_2 &= \pi r_2^2 \rho E_{DA} \\
&+ \left[(1-\lambda) \pi r_2^2 \rho + (1-\lambda)^2 \pi r_3^2 \rho \right. \\
&\quad \left. + (1-\lambda)^3 \pi r_4^2 \rho \right] \\
&\times \left[E_{\text{elec}} + \varepsilon_{\text{fs}} (r_2 + r_1)^2 \right] \\
\text{The 1st level: } E_1 &= \pi r_1^2 \rho E_{DA} \\
&+ \left[(1-\lambda) \pi r_1^2 \rho + (1-\lambda)^2 \pi r_2^2 \rho \right. \\
&\quad \left. + (1-\lambda)^3 \pi r_3^2 \rho + (1-\lambda)^4 \pi r_4^2 \rho \right] \\
&\times \left[E_{\text{elec}} + \varepsilon_{\text{fs}} r^2 \right], \tag{12a}
\end{aligned}$$

$$E_1 \cong E_2 \cong E_3 \cong E_4, \tag{12b}$$

$$r_1 + r_2 + r_3 + r_4 = \frac{L}{2}.$$

5. Intracluster Energy Balancing

In this section, we describe the strategy adopted for intracluster energy balancing in details. Firstly, two different nodes, the processor and the forwarder, will be elected as cluster leaders instead of the common CH. The election of processors considers both the residual energy and the distance between the candidates and other nodes, and their locations in each cluster are regarded as ideal. Then, clusters are formed based on the radius obtained above (see Figure 3).

5.1. Processor and Forwarder Election. At the beginning of each round of rotation, each node broadcasts message E_Msg (ID , $Energy$, $L(x, y)$) with radius r_j which includes sensor node ID, residual energy, and node coordinate (we take clusters in level j as an example ($1 \leq j \leq M$)). Any other node within communication radius r_j is considered as their neighbors and updates the neighbor information table after receiving messages. Every node whose residual energy is higher than $E_{\text{res_MAX}}$ has chance to participate in the processor and forwarder competition and become a candidate. Then, each candidate will calculate the mean residual energy EM_j of all neighbors according to the updated table:

$$EM_j = \sum_{j=1}^m \frac{E_{rj}}{m}, \tag{13}$$

where EM_j is the mean residual energy of node S_j and m is the total number of S_j 's neighbors.

It is easy to determine the mean communication distance among node S_j and its neighbor nodes:

$$d_j = \sum_{k=1, k \neq j}^m \frac{d_{jk}}{m}. \tag{14}$$

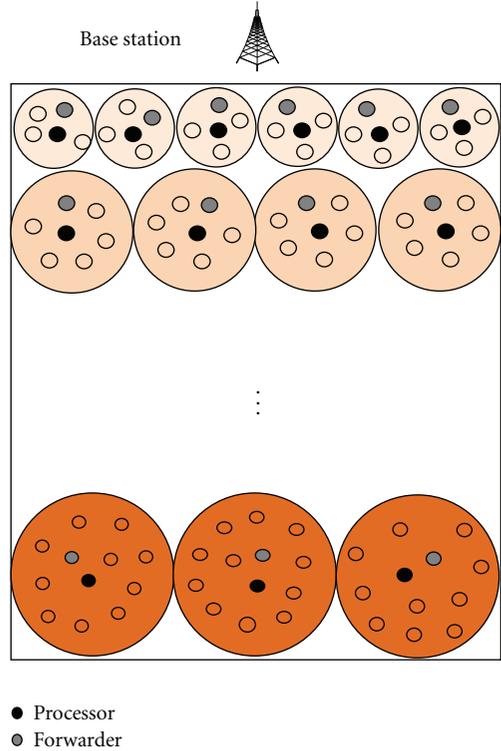


FIGURE 3: Election of cluster leaders.

d_{jk} is the distance between node S_j and S_k :

$$d_{jk} = \sqrt{(x_j - x_k)^2 + (y_j - y_k)^2}. \tag{15}$$

With consideration of both residual energy and distance, each candidate calculates the competition bids of being elected as processor and forwarder using (16) and (17), respectively:

$$CP_j = \chi \frac{E_{rj}}{EM_j} + \delta \frac{1}{d_j}, \tag{16}$$

$$CF_j = \chi \frac{E_{rj}}{EM_j} + \delta \frac{r_j + r_{j-1}}{d(j, BS)}. \tag{17}$$

The value of χ and δ is determined by the distribution of nodes within cluster and their residual energy situation. $d(j, BS)$ denotes the distance between S_j and the BS. And then the candidate broadcasts competition message Com_Pro (ID_j, E_{rj}, CP_j, CF_j) with radius r_j .

All the candidates are set in receive state and wait a time T . The length of T is determined to at least make sure that the nodes can receive the competition message from all its neighbors. Then each candidate compares competition bids of itself and all competition packet bids. The one with the largest value of CP_j will succeed in competition for processor while the one with the largest value of CF_j for forwarder. If the highest bids of CP_j or CF_j are even, the node with higher residual energy will be chosen. If a candidate possesses both the largest value of CP_j and CF_j , it will play the two roles at the same time.

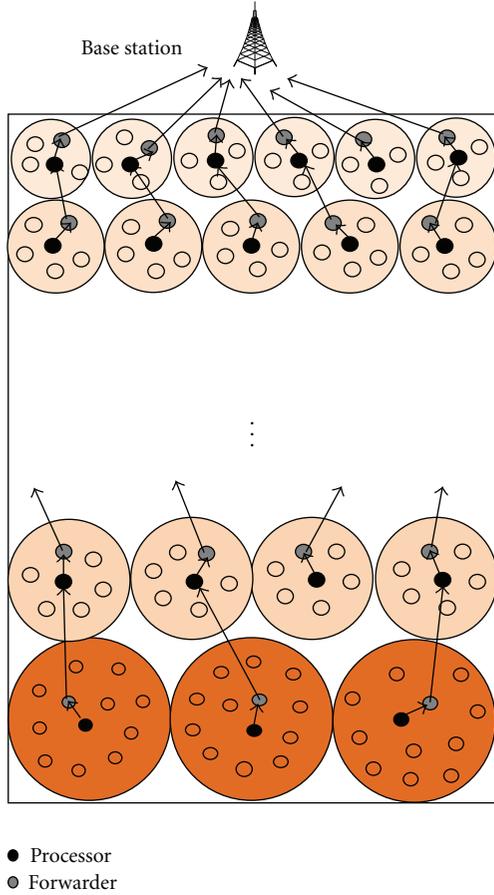


FIGURE 4: Transmission mode of cluster leaders.

5.2. Cluster Formation. According to the comparison results, the eligible candidate will broadcast processor competition success message $Suc_Pro (ID_j, Er_j, L(x_j, y_j))$ with radius r_j and, if not, wait for $Suc_Pro (ID_j, Er_j, L(x_j, y_j))$ message from neighbor nodes with the highest competition bids. Nodes give up competition as soon as they receive $Suc_Pro (ID_j, Er_j, L(x_j, y_j))$ message from neighbors. Meanwhile, they send $Join_Pro (ID_j, Er_j, L(x_j, y_j))$ message to neighbors with the highest transmit power. As forwarders are only responsible for forwarding the aggregated results, there is no need for broadcasting the success message of forwarder competition to all nodes in the cluster. It only adds the $Suc_For (ID_j, Er_j, L(x_j, y_j))$ information to the cluster-joining packet and then sends it to the processor (this step can be omitted if it is the same node). Therefore, no much overhead will be appended to the proposed algorithm compared with the existing algorithms. Then, the processor determines the TDMA slot assignment for the cluster members.

5.3. Data Aggregation Tree Construction. In this paper, the multihop algorithm considers nodes on the forwarder backbone in the forwarding direction (i.e., closer to the base station) only. Data aggregation tree generation algorithm is as follows.

We take level p and level q for example ($1 \leq p < q \leq M$). Suppose the forwarder of cluster a (CF_a) in level q needs to

choose a processor in level p as its relay node. Let clusters b and c in level p be the two nearest clusters from cluster a in level q . According to (1), CF_a can figure out the difference value between the energy consumption for sending its data packets to CP_b and CP_c :

$$\Delta E_T = \epsilon_{fs} * k_a * (d_b^2 - d_c^2). \quad (18)$$

And the difference value between the residual energy of CP_b and CP_c is

$$\Delta E = E_b - E_c. \quad (19)$$

Compare ΔE with ΔE_T ; if ΔE is bigger than ΔE_T , we will choose CP_b as the relay node; otherwise, we will choose CP_c . That means only if the residual energy of the candidate with longer distance to the current forwarder is much more than the closer one, the longer distance one can be chosen.

Each forwarder computes the value of ΔE and ΔE_T for the two nearest candidates respectively. It will choose as its relay (parent) node according to the calculation results. This strategy not only considers the residual energy of processors but also the total energy consumption of the whole network. Meanwhile, it considers the communication distance as well as the balance of energy consumption among all forwarders. Therefore, the overall situation of the network is taken better care of in our mechanism.

In the process of communication, each processor gathers the data from members except the forwarder within its cluster, aggregates them into one packet, and then transmits them to the forwarder. The forwarder will aggregate the compressive data with its own data, and then transmit them to its next forwarder (see Figure 4).

The proposed algorithm consists of four procedures: processor and forwarder election, cluster formation, data aggregation tree construction, and data transmission. The pseudocode of SCA is shown in Pseudocode 1.

5.4. Clusters Maintenance. As the power of cluster leaders may be exhausted quickly because of the much larger loads imposed on them, the phase of cluster maintenance is very important. In SCA, the cluster maintenance phase consists of cluster-leader rotations within a cluster and cross-level data transmission to BS.

- (i) Cluster-leader rotations in a cluster: if the remaining power of any processor or forwarder is under E_{res_MAX} , a new one is elected from among other plain nodes, while a *change_msg* is broadcast to inform cluster members of the change of cluster leaders.
- (ii) Cross-level data transmission to the BS: as clusters in the 1st level are the smallest in size, the process of taking turns serving as cluster leaders for nodes in it may finish quickly. Therefore, when the BS is aware that each sensor node in the 1st level can no longer serve as a cluster leader, it will broadcast a message to allow the cluster leaders in the 2nd level to transmit data to BS directly (see Figure 5). It is the same for 3rd level, 4th level, ..., Mth level. In this way, the network lifetime can be prolonged.

Definitions:

CP_j : the competition bids of being elected as processor

CF_j : the competition bids of being elected as forwarder

$S_{i\text{-status}}$: the status of node S_i

$S_{i\text{-processor}}$: the status of node S_i is processor

$S_{i\text{-forwarder}}$: the status of node S_i is forwarder

$S_{i\text{-plain}}$: the status of node S_i is plain node

E : the difference value between the residual energy of two candidate forwarders

E_T : the difference value between the energy consumption for sending its data packets to two candidate forwarders

- (1) **Procedure** Processor and forwarder election
- (2) Each node broadcasts message packet
- (3) Neighbors update the neighbor information table
- (4) Each candidate calculates the competition bids CP_j and CF_j
- (5) Broadcast competition message Com_Pro
- (6) Receive and compare competition bids
- (7) Determine processor and forwarder
- (8) **end procedure**
- (9) **Procedure** Cluster construction
- (10) **if** $S_{i\text{-status}} \leftarrow S_{i\text{-processor}}$ **then**
- (11) S_i broadcasts competition success message with its radius
- (12) **else if** $S_{i\text{-status}} \leftarrow S_{i\text{-forwarder}}$ **then**
- (13) Add the success information to the joining packet to the processor
- (14) **else if** $S_{i\text{-status}} \leftarrow S_{i\text{-plain}}$ **then**
- (15) Send joining message to the processor
- (16) **end if**
- (17) **end procedure**
- (18) **Procedure** Data aggregation tree construction
- (19) Forwarders broadcast the cost message packet
- (20) Calculate ΔE and ΔE_T
- (21) Compare ΔE with ΔE_T and choose the eligible one as its relay node
- (22) **end procedure**
- (23) **Procedure** Data transmission
- (24) **while** $S_{i\text{-status}} = S_{i\text{-processor}}$ **do**
- (25) **if** $S_{i\text{-processor}} \leftarrow S_{i\text{-forwarder}}$ **then**
- (26) Aggregate all the data from members and transmit to its next forwarder
- (27) **else**
- (28) Receive all the data from members except the forwarder
- (29) Aggregate and transmit data to the forwarder
- (30) **end if**
- (31) **end while**
- (32) **while** $S_{i\text{-status}} = S_{i\text{-forwarder}}$ **do**
- (33) Receive data from the processor of its cluster
- (34) Aggregate and transmit to its relay forwarder
- (35) **end while**
- (36) **while** $S_{i\text{-status}} = S_{i\text{-plain}}$ **do**
- (37) Transmit data to the processor
- (38) **end while**
- (39) **end procedure**

PSEUDOCODE 1: Pseudocode of the proposed algorithm.

6. Performance Evaluation

6.1. Complexity Analysis. An analysis of the SCA algorithm is made in this section. As we can see from Figure 3, the process of processor and forwarder election is message driven; thus we first discuss its message complexity.

Lemma 1. *The message complexity of the cluster formation algorithm is $O(N)$ in the network.*

Proof. At the beginning of the processor and forwarder competition selection phase, there will be N messages $E_Msg(ID, Energy, L(x, y))$ broadcasted by all nodes (N is the total number of sensor nodes). As each node whose residual energy is higher than E_{res_MAX} has chance to become a candidate, we assume that the ratio of eligible nodes is p . Then Np candidates are produced and each of them broadcasts a competition message $Com_Pro(ID_j, Er_j, CP_j, CF_j)$.

TABLE 2: Comparison of the message complexity.

	SCA	ACT	EECA	UCR	EECS	MR-LEACH	LEACH
Message complexity	$(O)N$	$(O)N$	$(O)N$	$(O)N$	$(O)N$	$(O)N$	$(O)N$

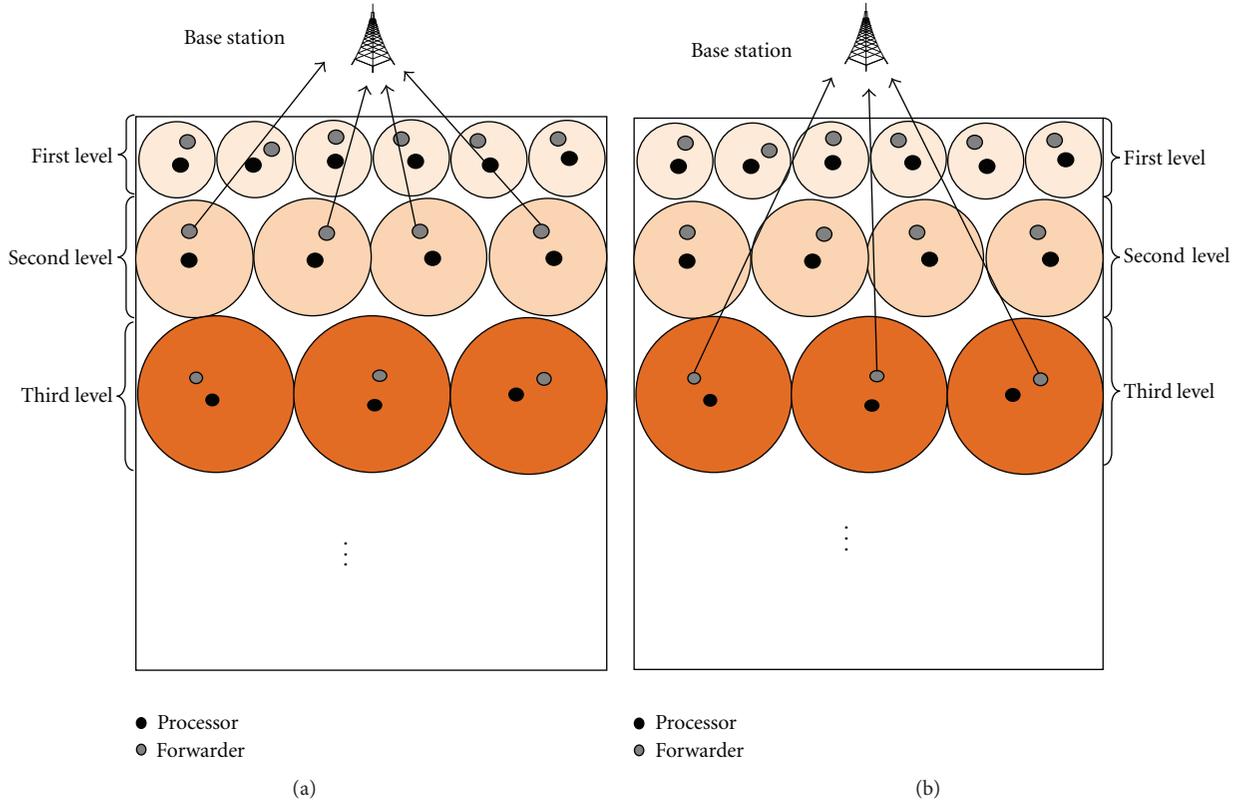


FIGURE 5: The architecture of cross-level data transmission.

Suppose k processors are selected, and they will send out k competition success message $Suc_Pro(ID_j, Er_j, L(x_j, y_j))$. Accordingly, there will be $N-k$ $Join_Pro(ID_j, Er_j, L(x_j, y_j))$ messages sent by other nonprocessors. As forwarders only add the competition success message to the cluster-joining packet, there will not be other extra messages produced. Thus the messages add up to $N + pN + k + N - k = (p + 2)N$ at the cluster formation stage per round, that is, $O(N)$.

Table 2 provides the comparison results of the message complexity for several existing protocols. Although two different nodes, the processor and the forwarder, are elected as cluster leaders instead of the common CH, the message complexity of our proposed algorithm is not added. Clearly, our approach is better than others being used for comparison. \square

6.2. Correctness Analysis

Lemma 2. *There is no chance that two nodes are both processors or forwarders if one is in the other's neighboring node set R_{CH} .*

Proof. Suppose S_u and S_v are both candidates in the cluster-leader selection phase, and S_u is in S_v 's neighboring node set

$S_v - R_{CH}$. According to our proposed algorithm, if S_u and S_v possess the even highest bids of CP_j or CF_j , the one which possesses higher residual energy will be chosen. The most special occasion is that the highest bids of CP_j or CF_j and the residual energy of two nodes are both the same. In this case, if S_u first becomes a leader node, then it will notice S_v its state, so S_v quits the competition and becomes an ordinary node, and vice versa. That is to say, cluster leaders are well distributed. \square

6.3. Discussion

6.3.1. Percentage p . As we can see from Section 5.1, the percentage p of eligible nodes determines the number of candidates of cluster leaders. On the one hand, enough candidates guarantee good cluster-leader choosing in terms of residual energy. On the other hand, too many candidates will cause a considerable message overhead. Thus a proper value of p should be chosen in order to guarantee the quality of cluster-leader selection and reduce the message overhead.

6.3.2. Synchronization. Synchronization is another important issue needed to be paid attention to for the operation

of SCA. It is assumed that all sensor nodes are synchronized and start the clustering phase at the same time. We can achieve it, for instance, by having the base station periodically broadcast synchronization pulses. Readers can obtain more details about the time synchronization issue in clustered wireless sensor networks with reference to [22].

6.3.3. Delay and throughput. The election of two nodes as cluster leaders will have some impact on delay and throughput of the whole network. Processors will transmit the processed data to their forwarders after their collection and aggregation instead of transmitting them directly to the next relay forwarder. This forwarding process takes a not long but certain time, so it would imply waiting longer at next-aggregation points and delaying the final delivery. Accordingly, the throughput will decrease. However, the adopted cross-level data transmission mode in the later phase will reduce the latency as well as increase the throughput, which will compensate for the total performance degradation.

7. Analysis of Energy Consumption

As outlined in Sections 4 and 5, the total energy consumed per round can be divided into two distinct phases which consist of the cluster set-up phase and the data transfer phase. The mathematical expressions that calculate an estimation of the energy consumed in each phase are provided, which we use to evaluate whether the loads are more balanced by adopting task separation. According to (1)–(3), we can obtain (20)–(27) as follows.

7.1. Clustering Phase. As described in Section 5, each round consists of creating a dominating set of cluster leaders chosen from a certain amount of candidates. We assume that the ratio of eligible nodes is p , then Np candidates are produced and each of them will broadcast a competition message. We take a cluster in level j as an example ($1 < j < M$). Assuming that the length of one message is l bytes and $A = L \times W$, then the energy consumed by candidates in a cluster per round is given by

$$E_c^1 = Npl \times \frac{\pi r_j^2}{A} (E_{\text{elec}} + \varepsilon_{\text{fs}} r_j^2) + ql \times \frac{\pi r_j^2}{A} \times E_{\text{elec}}, \quad (20)$$

where the first term represents the energy consumed for transmitting competition messages sent by cluster-leader candidates. The second term signifies the energy consumed in receiving the compete messages from other cluster-leader candidates within the competition radius. The number of messages received is based on the estimate that q cluster-leader candidates will fall within the competition radius.

Suppose that k processors are selected and each of them will send out a competition success message within radius r_j . Thus the energy consumed in each cluster for the processor advertisement message will be

$$E_c^2 = k \times \frac{\pi r_j^2}{A} \times l \times (E_{\text{elec}} + \varepsilon_{\text{fs}} r_j^2). \quad (21)$$

As there are $N - k$ nonprocessor nodes, each of which will receive this message and then sends a Join_Pro message to its processor; energy consumed during this process will be

$$E_c^3 = (N - k) \times \frac{\pi r_j^2}{A} \times l \times E_{\text{elec}} + (N - k) \times \frac{\pi r_j^2}{A} \times l \times (E_{\text{elec}} + \varepsilon_{\text{fs}} r_j^2). \quad (22)$$

Finally, each processor will receive these Join_Pro messages and the amount of energy consumed will be

$$E_c^4 = (N - k) \times \frac{\pi r_j^2}{A} \times l \times E_{\text{elec}}. \quad (23)$$

As forwarders only add the competition success message to the cluster-joining packet, there will not be other extra messages produced and then more energy consumed.

7.2. Data Transmission Phase. In the data transmission phase, each plain node sends a single data message of t bytes to the processor, and the energy consumed is

$$E_c^5 = (N - k) \times \frac{\pi r_j^2}{A} \times t \times (E_{\text{elec}} + \varepsilon_{\text{fs}} r_j^2). \quad (24)$$

Then each processor will receive these data messages:

$$E_c^6 = (N - k) \times \frac{\pi r_j^2}{A} \times t \times E_{\text{elec}}. \quad (25)$$

Next, each processor will aggregate the messages of its own cluster and relayed from its above level:

$$E_c^7 = \left[(N - k) \times \frac{\pi r_j^2}{A} \times t + D_{\text{relay}}^j \right] E_{\text{DA}}, \quad (26)$$

where D_{relay}^j represents the amount of data relayed from level $j + 1$ to level j . Since processors and forwarders in the same cluster are very close to each other, energy consumed can be considered negligible in the local forwarding process. Finally, forwarders will transmit these data to their next relay nodes:

$$E_c^8 = (1 - \lambda) \left[(N - k) \times \frac{\pi r_j^2}{A} \times t + D_{\text{relay}}^j \right] \times \left[E_{\text{elec}} + \varepsilon_{\text{fs}} (r_j + r_{j-1})^2 \right]. \quad (27)$$

From the equations given above, we can summarize the total energy consumed in each round by each processor and

forwarder in level j , respectively: $E_{\text{pro}} = E_c^2 + E_c^4 + E_c^6 + E_c^7$, $E_{\text{for}} = E_c^3 + E_c^8$,

$$\begin{aligned}
E_{\text{pro}} &= l \times (E_{\text{elec}} + \varepsilon_{\text{fs}} r_j^2) + (N - k) \times \frac{\pi r_j^2}{A} \times l \times E_{\text{elec}} \\
&\quad + (N - k) \times \frac{\pi r_j^2}{A} \times t \times E_{\text{elec}} \\
&\quad + \left[(N - k) \times \frac{\pi r_j^2}{A} \times t + D_{\text{relay}}^j \right] E_{\text{DA}}, \quad (28) \\
E_{\text{for}} &= l \times E_{\text{elec}} + l \times (E_{\text{elec}} + \varepsilon_{\text{fs}} r_j^2) + (1 - \lambda) \\
&\quad \times \left[(N - k) \times \frac{\pi r_j^2}{A} \times t + D_{\text{relay}}^j \right] \\
&\quad \times \left[E_{\text{elec}} + \varepsilon_{\text{fs}} (r_j + r_{j-1})^2 \right].
\end{aligned}$$

Let $Y = (N - k) \times (\pi r_j^2 / A)$, then

$$\begin{aligned}
E_{\text{pro}} &= l \times (E_{\text{elec}} + \varepsilon_{\text{fs}} r_j^2) + Y \times l \times E_{\text{elec}} + Y \times t \\
&\quad \times E_{\text{elec}} + (Y \times t + D_{\text{relay}}^j) E_{\text{DA}}, \quad (29) \\
E_{\text{for}} &= l \times (E_{\text{elec}} + \varepsilon_{\text{fs}} r_j^2) + l \times E_{\text{elec}} + (1 - \lambda) \\
&\quad \times \left[Y \times t + D_{\text{relay}}^j \right] \left[E_{\text{elec}} + \varepsilon_{\text{fs}} (r_j + r_{j-1})^2 \right].
\end{aligned}$$

In order to estimate the energy consumption of each processor and forwarder in one round, we consider the difference of their consumed energy. Our original goal is to lighten the load of CHs by task separation. If energy consumption of a couple of processor and forwarder in the same cluster in each round is nearly equal, the energy consumption of CHs is slowed down to half its common values, thus prolonging network lifetime to the maximum extent. Then, we have

$$\begin{aligned}
E_{\text{for}} - E_{\text{pro}} &= (Yt + D_{\text{relay}}^j) \\
&\quad \times \left\{ (1 - \lambda) \left[E_{\text{elec}} + \varepsilon_{\text{fs}} (r_j + r_{j-1})^2 \right] - E_{\text{DA}} \right\} \\
&\quad - (Y - 1) l E_{\text{elec}} - Y t E_{\text{elec}}. \quad (30)
\end{aligned}$$

As it is an equation whose highest order is quartic (r_j^4), it is not easy to observe their difference intuitively. So we randomly take a distribution whose total level $M = 5$ as an example. Assuming $j = 3$, we calculate the value of $E_{\text{for}} - E_{\text{pro}}$.

TABLE 3: Calculation parameters.

Parameter	Value	Unit
N	98	
A	9800	m^2
K	20	
λ	0.15	
r_2	7.12	m
r_3	11.36	m
r_4	18.12	m
r_5	28.91	m
E_{elec}	50	nJ/bit
ε_{fs}	10	pJ/bit/m^2
E_{DA}	5	nJ/bit/signal
$l(t)$	1	byte

Combining with Tables 5 and 6, the parameters used are listed in Table 3:

$$Y = (N - k) \times \frac{\pi r_j^2}{A} = 3.19,$$

$$D_{\text{relay}}^j = D_{\text{relay}}^3 = (1 - \lambda)^2 \pi r_5^2 \rho + (1 - \lambda) \pi r_4^2 \rho = 27.74,$$

$$\begin{aligned}
E_{\text{for}} - E_{\text{pro}} &= (8Y + 27.74) \left\{ 0.85 \left[50 + 0.01 \times (7.12 + 11.36)^2 \right] - 5 \right\} \\
&\quad - 8(Y - 1) \times 50 - 8Y \times 50 \\
&= (8Y + 27.74) \times 40.4 - 800Y + 400 \\
&= -0.3 (nJ). \quad (31)
\end{aligned}$$

From the above calculations we can see that, for a network which has 5 levels, the energy consumption difference between a couple of processor and forwarder in the middle level is only 0.3 nJ. That is to say, the consumed energy of each processor and its corresponding forwarder is nearly equal, and the processor really works for spreading the load. So we obtain the expected results.

8. Simulations

We conduct simulations to study the performance of our proposed energy balancing algorithm. First of all, we describe the simulation settings. Secondly, simulation results are presented showing the performance results under different performance metrics. Finally, we discuss and analyze the simulation results. Table 4 provides a comparison of the related work with respect to different clustering attributes, from which we choose LEACH, EECA, MR-LEACH, and ACT for comparison.

8.1. Simulation Environment. We analyze the performance of SCA algorithm by Omnet++ which allows efficient and realistic modeling of sensor nodes by using an integrated technical

TABLE 4: Comparison of related works with respect to clustering attributes.

Clustering protocol	Cluster-leader selection	Cluster formation	Cluster size	Data aggregation tree construction	Energy dissipation
LEACH	Random	Closest CH	Equal	No	Unbalanced
MR-LEACH	Nodes with the largest residual energy	Closest CH	Equal	BS helps to choose relay CHs for lower layer CHs	Somewhat balanced
EECS	Random with election	Closest based using three parameters	Unequal	No	Somewhat balanced
UCR	Random with election	Closest CH	Unequal	Greedy geographic forwarding algorithm based on relay path cost	Somewhat balanced
EECA	Based on two parameters	Closest CH	Equal	Based on a weight function	Relatively balanced
ACT	Ideal location	Closest CH	Unequal	Equal allocation of the relay loads	Relatively Balanced
SCA	Based on two parameters	Closest CH	Unequal	Based on the comparison of transmission and residual energy	Balanced

TABLE 5: Simulation parameters.

Parameter	Value
Number of nodes	98
Network scale (m ²)	70 × 140
Location of BS (m)	(40, 150)
Initial energy of each node (J)	2
The ratio of candidates p	0.3
E_{elec} (nJ/bit)	50
ϵ_{fs} (pJ/bit/m ²)	10
d_0 (m)	100
E_{DA} (nJ/bit/signal)	5
Data packet size (bit)	500

computing environment. Because this paper focuses on energy-efficient and balanced routing in the network layer, an ideal MAC layer and error-free communication links are assumed for simplicity. We perform the simulation study under steady state, and the other parameters of the simulation are listed in Table 5.

8.2. Results and Analysis

8.2.1. Network Lifetime. First of all, we measure the lifetime of network. Figure 6 gives the number of living nodes over time. As evident from the figure, SCA has a longer network lifetime than LEACH, EECA, MR-LEACH, and ACT. As for LEACH, each sensor node elects itself as a CH with some probability with no regard to the residual energy. Moreover, all CHs communicate with the BS directly in LEACH which leads to high energy consumption in communication and thus shorting the network lifetime. Even though the CHs

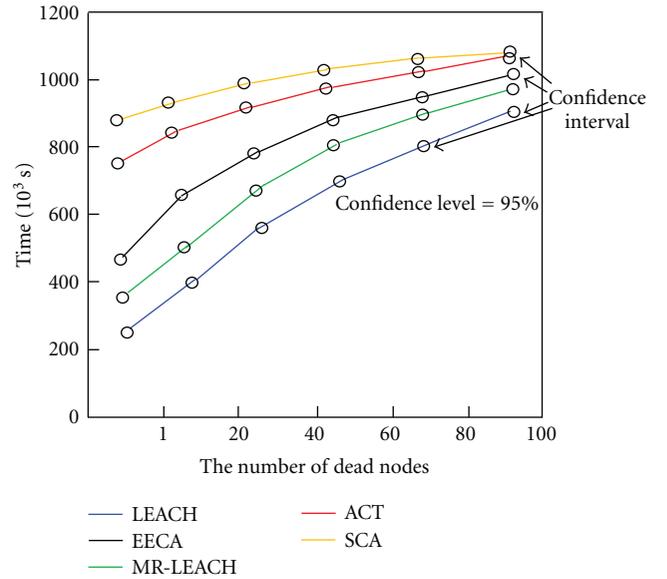


FIGURE 6: Network lifetime.

in EECA and MR-LEACH are selected from sensor nodes with sufficient power, their use of multihop communications increases the burden of cluster heads near the BS. As the CHs close to the BS share higher relaying loads, their energy would be used up faster and die earlier. ACT considers the adjustment of cluster sizes during data relay, but the same disadvantages with EECA and MR-LEACH still exist in it, so that it performs better than the two but worse than SCA. When the data is relayed among clusters, the cluster sizes are adjusted and the task of a CH is allocated to two nodes in SCA, which reduces the energy consumption of critical nodes; as a result, SCA achieves the longest network lifetime.

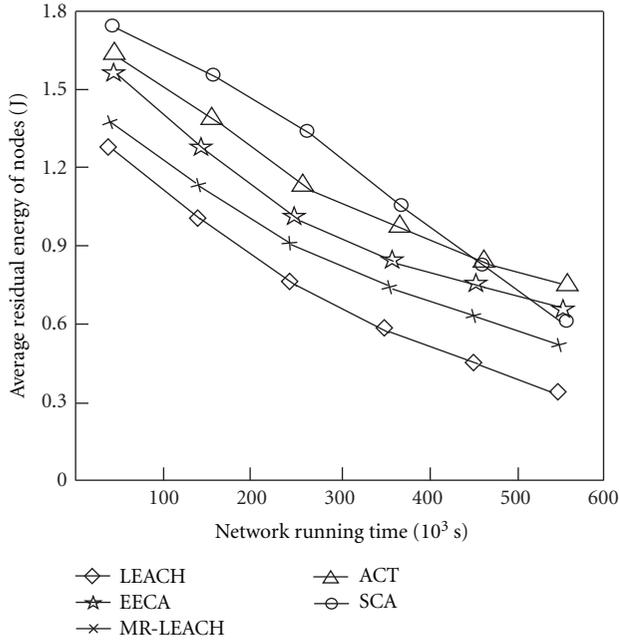


FIGURE 7: Average residual energy.

8.2.2. *Average Residual Energy.* Figure 7 compares the average residual energy of nodes of the five algorithms. We can observe that average residual energy of nodes under SCA algorithm is greater than that of the other four algorithms. LEACH adopts single-hop communications with the CH sending its data directly to the BS leading to its lower average residual energy. EECA, MR-LEACH, ACT, and SCA utilize multihop communications that require less energy consumption from each sensor node. With consideration of cluster size and cluster-leader production, SCA balances the load on each cluster and alleviates the burden of those critical nodes. In addition, $(\Delta E - \Delta E_T)$ is defined as the metric to choose relay nodes during data aggregation tree construction, which considers the total energy consumption of the whole network. In this way, energy spent by sensor nodes close to the BS is less than in EECA, MR-LEACH, and ACT, so the average energy dissipation in SCA is lower than that in the other four. But as time goes on, more and more clusters need to transmit their data to the BS directly in SCA. Forwarders bear most of the tasks at the later phase, and the effect of processors is not outstanding now. However, selecting two nodes for cluster leaders will consume more extra energy. Therefore, it increases the average energy dissipation in SCA and leads to more energy consumption than ACT and EECA after running for approximately $460/10^3$ s.

8.2.3. *The Standard Deviation of Energy Consumption of Cluster Leaders.* Figure 8 compares the standard deviation of energy consumption of cluster leaders in LEACH, EECA, MR-LEACH, ACT, and SCA. The CHs in LEACH are picked out randomly, providing each sensor node a chance to serve as a CH. Accordingly, the standard deviations of energy consumption of CHs in LEACH show substantial variations. MR-LEACH considers only the residual energy of nodes,

TABLE 6: Variations of cluster radius with level M .

Level M	r_1	r_2	r_3	r_4	r_5
$M = 3$	13.81	21.72	34.46	×	×
$M = 4$	7.69	12.16	19.34	30.81	×
$M = 5$	4.49	7.12	11.36	18.12	28.91

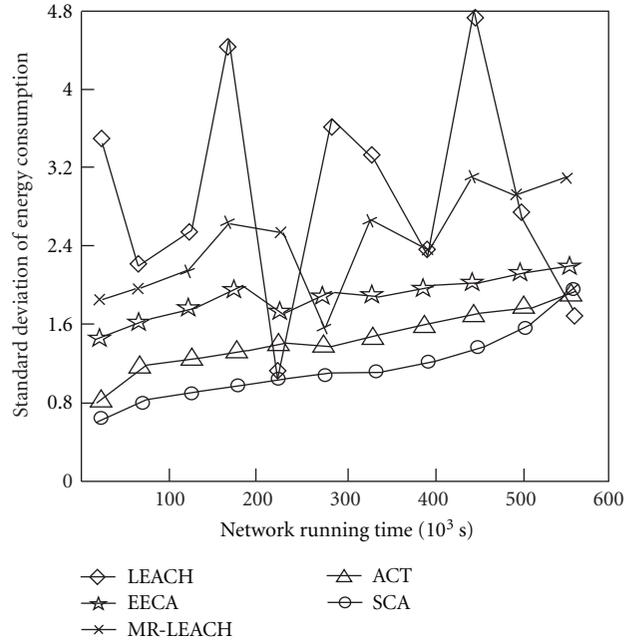


FIGURE 8: The standard deviation of energy consumption of cluster leaders.

and EECA chooses the CHs based on residual energy and distance. Thus their curves display irregular oscillation in each round. The CHs in ACT are chosen according to the ideal locations calculated depending on the load balance, and its curve of standard deviation for energy consumption is relatively steady. SCA calculates cluster sizes according to the loads on CHs to balance the given loads on each CH. Meanwhile, it separates tasks of one single cluster head to two nodes, which further balances the energy consumption of cluster leaders. As a result, the value of standard deviation of energy consumption in SCA is minimized.

Table 6 shows the variation of cluster radius with level M within the same scenario, which is consistent with our design idea in Section 4.1.

8.2.4. *Variance of Average Residual Energy.* Figure 9 shows the experiment result of variance of average residual energy of nodes in network and reflects the proportionality of network energy consumption. As to LEACH algorithm, most energy loads concentrate on CHs and the excessive energy consumption leads to early death of them, thus causing much more uneven distribution of node energy in network than all others being compared. In EECA, MR-LEACH, and ACT, the shorter the distance between cluster leaders and the BS, is the much heavier burden the leaders will have. In addition, even though ACT considers arranging cluster sizes based on

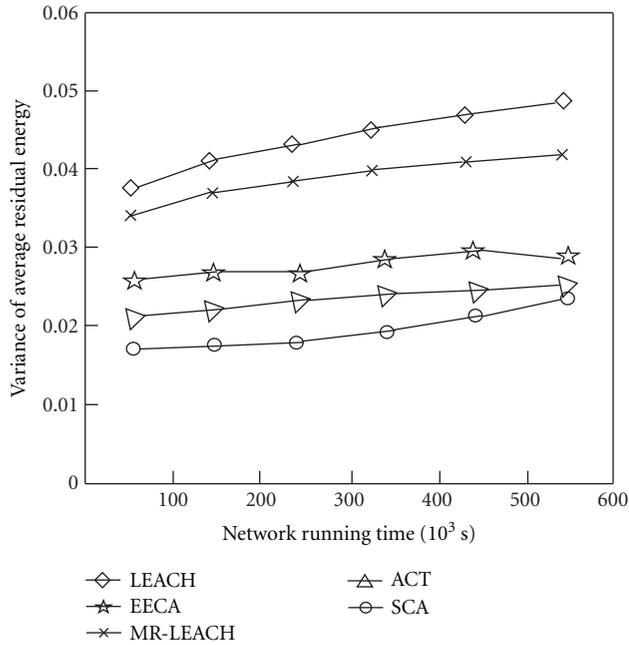


FIGURE 9: Variance of average residual energy.

the energy consumption, the selection of a new CH makes the locations of CHs deviate from the original ideal ones. All these make the scatterings of energy consumption oscillating. In comparison, SCA has a better and more stable value in the early phase, but in the later phase cross-level data transmission directly to the BS and the election of two new cluster leaders leads to the rapid reduction of energy.

9. Conclusions

In this paper, we focus on the problem of unbalanced energy dissipation when employing the multihop routing in a cluster-based WSN. We propose an approach that balances the energy consumption among clusters and slows the energy consumption of CHs. For intercluster energy balancing, a cluster radius is calculated with consideration of the relaying load undertaken by each cluster, thus balancing the energy dissipation among clusters. As for intracluster energy dissipation, a separating mode is adopted to alleviate the burden of critical nodes in each cluster and prolong the network time. Simulation results show that our method outperforms LEACH, MR-LEACH, EECA, and ACT in aspects of network lifetime, energy efficiency, and balanced extent of energy dissipation.

Acknowledgments

The authors would like to thank the anonymous reviewers and the editor for their constructive comments on this article. This work is supported by Program for Changjiang Scholars and Innovative Research Team in University (no. IRT1078), the Key Program of NSFC-Guangdong Union Foundation (U1135002), Major national S&T program (2011ZX03005-002), the National Natural Science Foundation of China

(no. 61272541, 61202389), and the Fundamental Research Funds for the Central Universities (no. JY10000903001).

References

- [1] P. Medagliani, J. Leguay, G. Ferrari, V. Gay, and M. Lopez-Ramos, "Energy-efficient mobile target detection in wireless sensor networks with random node deployment and partial coverage," *Pervasive and Mobile Computing*, vol. 8, no. 3, pp. 429–447, 2012.
- [2] N. Aslam, W. Phillips, W. Robertson, and S. Sivakumar, "A multi-criterion optimization technique for energy efficient cluster formation in wireless sensor networks," *Information Fusion*, vol. 12, no. 3, pp. 202–212, 2011.
- [3] A. F. Liu, P. H. Zhang, and Z. G. Chen, "Theoretical analysis of the lifetime and energy hole in cluster based wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 71, no. 10, pp. 1327–1355, 2011.
- [4] R. Zhu, Y. Qin, and J. Wang, "energy-aware distributed intelligent data gathering algorithm in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2011, Article ID 235724, 13 pages, 2011.
- [5] K. Iwanicki and M. Van Steen, "Gossip-based self-management of a recursive area hierarchy for large wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 4, pp. 562–576, 2010.
- [6] T. Kim, Y. Lee, J. Sung, and D. Kim, "Hierarchical network protocol for large scale wireless sensor networks," in *Proceedings of the 7th IEEE Consumer Communications and Networking Conference (CCNC '10)*, pp. 1–2, Las Vegas, Nev, USA, January 2010.
- [7] D. Kumar and R. B. Patel, "Multi-hop data communication algorithm for clustered wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2011, Article ID 984795, 10 pages, 2011.
- [8] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.
- [9] X. Fan and Y. Song, "Improvement on LEACH protocol of wireless sensor network," in *Proceedings of the International Conference on Sensor Technologies and Applications, (SENSORCOMM '07)*, pp. 260–264, Valencia, Spain, October 2007.
- [10] V. Loscri, G. Morabito, and S. Marano, "A two-levels hierarchy for low-energy adaptive clustering hierarchy (TL-LEACH)," in *Proceedings of the 62nd IEEE Vehicular Technology Conference (VTC '05)*, pp. 1809–1813, September 2005.
- [11] M. B. Yassein, A. Al-zou'bi, Y. Khamayseh, and W. Mardini, "Improvement on leach protocol of wireless sensor network (vleach)," *International Journal of Digital Content Technology and Its Applications*, vol. 3, no. 2, pp. 132–136, 2009.
- [12] M. O. Farooq, A. B. Dogar, and G. A. Shah, "MR-LEACH: multi-hop routing with low energy adaptive clustering hierarchy," in *Proceedings of the 4th International Conference on Sensor Technologies and Applications, (SENSORCOMM '10)*, pp. 262–268, Venice, Italy, July 2010.
- [13] M. Ye, C. F. Li, G. H. Chen, and J. Wu, "EECS: an energy efficient clustering scheme in wireless sensor networks," in *Proceedings of the 24th IEEE International Performance Computing and Communications Conference (IPCCC '05)*, pp. 535–540, 2005.

- [14] G. Chen, C. F. Li, M. Ye, and J. Wu, "An unequal cluster based routing protocol in wireless sensor networks," *Wireless Network*, vol. 15, no. 2, pp. 193–207, 2009.
- [15] F. Bouabdallah, N. Bouabdallah, and R. Boutaba, "On balancing energy consumption in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 6, pp. 2909–2924, 2009.
- [16] J. S. Li, H. C. Kao, and J. D. Ke, "Voronoi-based relay placement scheme for wireless sensor networks," *IET Communications*, vol. 3, no. 4, pp. 530–538, 2009.
- [17] W. K. Lai, C. F. Fan, and L. Y. Lin, "Arranging cluster sizes and transmission ranges for wireless sensor networks," *Information Sciences*, vol. 183, no. 1, pp. 117–131, 2012.
- [18] U. J. Jang, S. G. Lee, and S. J. Yoo, "Optimal wake-up scheduling of data gathering trees for wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 72, no. 4, pp. 536–546, 2012.
- [19] C. Sha, R. C. Wang, H. P. Huang, and L. J. Sun, "Energy efficient clustering algorithm for data aggregation in wireless sensor networks," *Journal of China Universities of Posts and Telecommunications*, vol. 17, no. 2, pp. 104–122, 2010.
- [20] Z. Liu, Q. Zheng, L. Xue, and X. Guan, "A distributed energy-efficient clustering algorithm with improved coverage in wireless sensor networks," *Future Generation Computer Systems*, 2011.
- [21] E. Zeydan, D. Kivanc, C. Comaniciu, and U. Tureli, "Energy-efficient routing for correlated data in wireless sensor networks," *Ad Hoc Networks*, vol. 10, no. 6, pp. 962–975, 2012.
- [22] S. el Khediri, A. Kachouri, and N. Nasri, "Diverse synchronization issues in wireless sensor networks," in *Proceedings of the International Conference on Microelectronics (ICM'11)*, pp. 1–6, Hammamet, Tunisia, 2011.

