Ad Hoc & Sensor Wireless Networks, Vol. 1, pp. 01–21
Reprints available directly from the publisher
Photocopying permitted by license only

An Energy Efficient Clustering Scheme in Wireless Sensor Networks

MAO YE[1], CHENGFA LI[1], GUIHAI CHEN[1]*, AND JIE WU[2]

[1]*National Laboratory of Novel Software Technology*
Nanjing University, 210093, Nanjing, China
[2]*Department of Computer Science and Engineering*
Florida Atlantic University, FL 33431, USA

Received: April 12, 2005, In Final Form: September 13, 2005.

Data gathering is a common but critical operation in many applications of wireless sensor networks. Innovative techniques that improve energy efficiency to prolong the network lifetime are highly required. Clustering is an effective topology control approach in wireless sensor networks, which can increase network scalability and lifetime. In this paper, we propose a novel energy efficient clustering schema (EECS) for single-hop wireless sensor networks, which better suits the periodical data gathering applications. Our approach elects cluster heads with more residual energy in a autonomous manner through local radio communication with no iteration while achieving good cluster head distribution; further more, it introduces a novel distance-based method to balance the load among the cluster heads. Simulation results show that EECS prolongs the network lifetime significantly against the other clustering protocols such as LEACH and HEED.

Keywords: wireless sensor networks (WSN), data gathering, clustering scheme, energy efficient design, network lifetime.

1 INTRODUCTION

Continued advances of MEMS and wireless communication technologies have enabled the deployment of large scale wireless sensor networks (WSNs) [1]. The potential applications of WSNs are highly varied, such as environmental monitoring, target tracking and military surveillance [2].

---

*Corresponding author e-mail: gchen@nju.edu.cn

Sensors in such a network are equipped with sensing, data processing and radio transmission units while the power is highly limited. Due to the sensors' limited power, innovative techniques that improve energy efficiency to prolong the network lifetime are highly required. Thus energy-aware design has been a hot research area at all layers of the networking protocol stack. Data gathering is a common but critical operation in many applications of WSNs, where data aggregation and hierarchical routing mechanism are commonly used techniques. Data aggregation can eliminate data redundancy and reduce communication load [3]. Hierarchical (clustering) mechanisms are especially effective in increasing network scalability and reducing data latency, and have been extensively exploited.

In this paper, we propose and evaluate an energy efficient clustering scheme (EECS) for periodical data gathering applications in WSNs. In the *cluster head election* phase, the cluster head is elected by localized competition, which is unlike LEACH, and with no iteration, which differs from HEED. The optimal value of competition range produces a good distribution of cluster heads. Further in the *cluster formation* phase, plain nodes join clusters not only taking into account its intra-cluster communication cost, but also considering cluster heads' cost of communication to the BS. EECS is autonomous and more energy efficient, and simulation results show that it prolongs the network lifetime much more significantly than the other clustering protocols.

The remainder of this paper is organized as follows. Section 2 reviews some most related clustering protocols. Section 3 describes the assumptions and our prime goals. Section 4 exhibits the details of EECS and Section 5 analyzes several properties of EECS. Section 6 evaluates the performance of EECS via simulations and compares EECS with some other cluster protocols. Finally, Section 7 concludes the paper and gives the directions of future work.

2 RELATED WORK

LEACH [4] is the first clustering protocol proposed for periodical data gathering applications in WSNs. It assumes that sensor nodes communicate with each other by single-hop only, and they can transmit the data to the base station directly. Its operation is divided into rounds and each round is composed of two phases. In the cluster formation phase, LEACH elects some cluster heads according to the equation 1,

$$P_i(t) = \begin{cases} \frac{k}{N-k(r \mod N/k)}, & C_i(t) = 1 \\ 0, & C_i(t) = 0 \end{cases}, \tag{1}$$

where k is the desired number of cluster heads, $C_i(t)$ is the indicator function determining whether or not node i has been a cluster head in the most recent (r mod N/k) rounds, and $P_i(t)$ is the probability for node i to become a cluster head at round r. The rest of the sensor nodes choose the proper cluster to join according to the signal strength from the cluster heads. In the data transmission phase, the cluster heads aggregate the data from their cluster members, and send the aggregated data to the base station by single-hop communication. Since the cluster heads rotate in each round, the load is balanced to a certain extent.

Subsequently, the clustering technique in WSNs has been extensively exploited. The problem of clustering network organization consists of several issues, one of which is how to balance the energy consumption among nodes well to prolong the network lifetime. So far, several energy efficient clustering mechanisms have been proposed and we review some of them as follows.

In [5], to cope with the heterogenous energy circumstance, the node with higher energy has larger probability to become the cluster head. The authors propose an energy-aware $P_i(t)$ as follows:

$$P_i(t) = min\left\{\frac{E_i(t)}{E_{total}(t)}k, 1\right\}, \tag{2}$$

where $E_i(t)$ is the current energy of node i, and

$$E_{total} = \sum_{i=1}^{N} E_i(t). \tag{3}$$

Using equation 2, the nodes with higher energy are more likely to become cluster heads than nodes with less energy. Thus energy-aware LEACH is more energy efficient in the heterogenous scenario. However, to calculate the probabilities in 2, each node must have an estimate of the total energy of all nodes in the network. An underneath routing protocol should be proposed to allow each node to determine the total energy, whereas the probability in equation 1 enables each node to make a completely autonomous decision.

One primary cause of unbalanced energy depletion is the different distance from the BS. [6, 7] both provide clustering mechanisms based on distance information accordingly. Considering the nodes that are far from the BS consume much more energy for data transmitting, they adjust the system parameter k (the same meaning as in LEACH) to better the probability of electing as the cluster head for each node. Since pure distance-driven is not so direct as energy-driven for energy concern, they perform poor against the energy-driven protocols. Moreover, in order to get the proper system parameters, distance-driven protocols need to use the global distance information, such as in [6]; and that severely affects the scalability of a distributed system.

HEED [8] is a protocol which periodically selects cluster heads according to a hybrid of the node residual energy and a secondary parameter through constant time iterations. It uses the primary parameter, i.e. residual energy, to select an initial set of cluster heads and the secondary parameter $AMRP$,

$$AMRP = \frac{\sum_{i=1}^{M} MinPwr_i}{M}, \tag{4}$$

to "break ties" among them, where $MinPwr_i$ denote the minimum power level required by a node v_i, $1 \leq i \leq M$, to communicate with a cluster head, and M is the number of nodes within the cluster range. HEED achieves fairly uniform distribution of the cluster heads across the network.

TPC [9] is a novel two-phase clustering (TPC) scheme for energy-saving and delay-adaptive data gathering in wireless sensor networks. Each node advertises for cluster head with a random delay, and the node who overhears others' advertisement will give up its own advertisement. In such a way, the network is partitioned into clusters in the first phase. In the second phase, each member searches for a neighbor closer than the cluster head within the cluster to set up an energy-saving and delay-adaptive data relay link. With the advantages of chain topology, TPC achieves a great tradeoff between the energy and delay.

Besides the aforementioned protocols, there exists several other protocols, such as [10–13]. [10] proposes a distributed and randomized LEACH-like algorithm which provide methods to compute the optimal values of the algorithm parameters a prior and use multi-hop technique in both intra-cluster and inter-cluster communications. ACE [11] is an emergent algorithm that uses just three rounds of feedback to form an efficient cover of clusters across the network. It uses the node degree as the main parameter to elect cluster heads. [12] proposes a clustering algorithm based on ANTCLUST [13]. Using this method, the sensor nodes with more residual energy independently become cluster heads. However, it produces much control overhead during iterations.

3 PROBLEM OUTLINE

In this paper, our motivation is to study the problem in a periodical data gathering application for which LEACH is proposed, i.e. single-hop communication. Aiming to ensure fairness, we adopt the single-hop model in EECS, rather than discussing the clustering mechanism in the multi-hop model. In this typical data gathering application, sensors periodical sense the environment and transmit the data to the base station (BS), then the BS analyzes the data to draw some conclusions about the activities in the field. We make a few assumptions about the network model and introduce the energy consuming model before the problem statement.

3.1 Network Model

To simplify the network model, we adopt a few reasonable assumptions as follows:

1. N sensors are uniformly dispersed within a square field A. The BS is deployed far away from A.
2. All sensors and BS are stationary after deployment. The location of BS is known by each node. Each sensor with enough energy can communicate with BS directly.
3. Sensors can use power control to vary the amount of transmit power depending on the distance to the receiver as in [5]. For instance, Berkeley Motes [14] have about 100 different transmission power levels. For simplicity, we assume the power level is continuous.
4. Communication is symmetric and a sensor can compute the approximate distance based on the received signal strength if the transmission power is known.
5. All sensors are location-unaware, i.e. not equipped with GPS.
6. All sensors are homogeneous, i.e., they have the same capacities.

The first four assumptions are common in WSNs. Sensors sense the environment while the base station periodically gathers these sensed information. The fifth assumption is reasonable and typical in the sensor networks, since it is costly that each sensor is equipped with GPS equipment. Energy is a scarce resource for all sensors and there is no super sensor with unlimited power in the network. So the last assumption motivates the rotation of cluster heads to balance the load across the network.

In addition, there is no assumption about the initial energy. Initial energy in each sensor can be arbitrary.

3.2 Energy Consuming Model

We use a simplified model shown in [5] for the radio hardware energy dissipation. We refer readers to [5] for more details.

To transmit an l-bit data to a distance d, the radio expands energy:

$$E_{T_x}(l,d) = \begin{cases} l \times E_{elec} + l \times \epsilon_{fs} d^2, & d < d_{crossover} \\ l \times E_{elec} + l \times \epsilon_{mp} d^4, & d \geq d_{crossover} \end{cases}. \tag{5}$$

The first item presents the energy consumption of radio dissipation, while the second presents the energy consumption for amplifying radio. Depending on the distance between the transmitter and receiver, both the free space ϵ_{fs} (d^2 power loss) and the multi-path fading ϵ_{mp} (d^4 power loss) channel models are used [15].

When receiving this data, the radio expends energy:

$$E_{R_x}(l) = l \times E_{elec}. \tag{6}$$

TABLE 1
Meanings of the Notations

Notation	Meaning
A	the area of the network
N	the number of sensor nodes
μ	a random number between 0 and 1
T	a threshold between 0 and 1
id	the identity of the node
$E_{residual}$	the residual energy of the node
m_j	the sum of members in cluster j
CH	the set of cluster heads
P	the set of plain nodes
CH_i	the i^{th} node in CH
P_j	the j^{th} node in P
$d(x,y)$	the distance between node x and y
$d(x,BS)$	the distance between node x and BS
$exp(x)$	the expectation of x
$R_{compete}$	radius when the candidate nodes broadcast

Additionally, data aggregation is adopted to save energy. It is assumed that the sensed information is highly correlated, thus the cluster head can always aggregate the data of its members into a single length-fixed packet. This operation also consumes energy $E_{DA}(nJ/bit/signal)$.

3.3 Problem Statement

Once a sensor node runs out its energy, we consider the network dead because some area cannot be monitored any more. Periodical data gathering applications in large scale sensor networks appeal the design of scalable, energy efficient clustering algorithms. Thus our goals in this paper are as follows: 1) *autonomous manner*. In large scale sensor networks, centralized control manner is not practical. The large scale and limited battery power appeal an autonomous algorithm; 2) *low control overhead*. Gathering data is the ultimate task in data gathering applications. Since power is the most scarce resource of sensors, it is desirable to reduce control overhead to extend the time of data gathering; 3) *load balanced clustering mechanism*. All sensors have equal capacity and are energy constrained, so load balance is a great issue in the design. In the next section, we will describe the EECS algorithm in details.

4 EECS DETAILS

EECS is a LEACH-like clustering scheme, where the network is partitioned into a set of clusters with one cluster head in each cluster. Communication between the cluster head and BS is direct (single-hop). For easy reference, we summarize the notations in Table 1 and describe the states of nodes and control messages in Table 2.

TABLE 2
Descriptions of the States or Messages

State	Description
`CANDIDATE`	the node is a candidate node
`HEAD`	the node is selected as the head
`PLAIN`	the node is a plain node
Message	**Description**
`COMPETE_HEAD_MSG`	tuple($id, E_{residual}$)
`HEAD_AD_MSG`	tuple($id, E_{residual}$)
`JOIN_CLUSTER_MSG`	tuple($id, d(BS, id)$)
`SCHEDULE_MSG`	tuple(*(id,time_slot_NO.)*,...)

In the network deployment phase, the BS broadcasts a "hello" message to all the nodes at a certain power level. Each node can compute its approximate distance to the BS based on the received signal strength. This helps nodes to select the proper power level when communicating with the BS. As will be shown in the *cluster formation* phase, we will use this distance to balance the load among cluster heads. In the *cluster head election* phase, well distributed cluster heads are elected with a little control overhead. And in the *cluster formation* phase , a novel weighted function is introduced to construct load balanced clusters. Detailed descriptions of the two phases are in the following subsections and Algorithms 1 and 2.

4.1 Cluster Head Election

In this phase, several cluster heads are elected. Nodes become `CANDIDATE` nodes with a probability T, and then broadcast the `COMPETE_HEAD_MSG`s within radio range $R_{compete}$ to advertise their wills. As in the pseudo-code between line 6∼16 in Algorithm 1, each `CANDIDATE` node receives the `COMPETE_HEAD_MSG`s within its $R_{compete}$ range and checks whether there is a `CANDIDATE` node with more residual energy within this radius. Once the `CANDIDATE` node finds a more powerful `CANDIDATE` node, it will give up the competition without receiving the subsequential `COMPETE_HEAD_MSG`s. Otherwise, it will be elected as `HEAD` in the end. Additionally, we use id to break the tie of $E_{residual}$ during the comparisons.

4.2 Cluster Formation

In this phase, each `HEAD` node broadcasts the `HEAD_AD_MSG` across the network, while the `PLAIN` nodes receive all the `HEAD_AD_MSG`s and decide which cluster to join. The most used metric for `PLAIN` nodes to make decisions is the distance metric. For example in [5] or [10], the `PLAIN` nodes choose the cluster head that require minimum communication according to the received signal strength. However, pursuing efficient energy consumption of the `PLAIN` nodes only may lead some `HEAD` nodes to be exhausted quickly during the *data transmission* phase.

```
/*
$\forall n_i \in N$;
$Timer_{phase1}$ guarantees that each node has enough time to complete its operation. */
state←PLAIN;
$\mu \leftarrow$ Random(0,1);
if ($\mu < T$) then
    state←CANDIDATE;
    broadcast COMPETE_HEAD_MSG;
    while ($Timer_{phase1}$ has not expired) do
        msg ← receive COMPETE_HEAD_MSG;
        $id \leftarrow$ msg.$getID()$;
        if ($E_{residual} < n_i.E_{residual}$) then
            state←PLAIN;
            break;
        else if (($E_{residual} = n_i.E_{residual}$) and ($id < n_i.id$)) then
            state←PLAIN;
            break;
        end if
    end while
end if
if (state = CANDIDATE) then
    state←HEAD;
end if
```

Algorithm 1: cluster head election

In the *data transmission* phase, the consumed energy of cluster head i, $E(CH_i)$, is as follows, assuming $d(CH_i, BS) > d_{crossover}$.

$$E(CH_i) = m_i l E_{elec} + (m_i + 1) l E_{DA} + l(E_{elec} + \epsilon_{mp} d^4). \qquad (7)$$

Observing equation 7, energy consumption of $E(CH_i)$ is composed of three parts: data receiving, data aggregation and data transmission. In the field, several cluster heads may be near the BS, while some are far away. The energy expended during data transmission for far away cluster heads is significant, especially in large scale networks. Since $d(CH_i, BS)$ has been fixed after *cluster head election* , we should justify the cluster size for each cluster head to balance their load across the network. The larger $d(CH_i, BS)$ is, the smaller member size m_i the cluster head CH_i should accommodate.

Energy consumption of the `PLAIN` node P_j during transmitting the data to CH_i is:

$$E(P_j) = \begin{cases} lE_{elec} + l\epsilon_{fs} d^2, & d < d_{crossover} \\ lE_{elec} + l\epsilon_{mp} d^4, & d \geq d_{crossover} \end{cases}, \qquad (8)$$

where $d = d(P_j, CH_i)$.

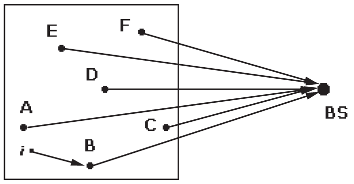


FIGURE 1
The effect of considering cluster-head's *cost*.

Observing equation 8, energy consumption of P_j is only determined by $d(P_j, CH_i)$. If P_j always chooses the cluster head CH_{best} with $\min\{E(P_j)\}$, CH_{best} may be exhausted due to long distance data transmission to the BS and immoderate cluster size, although the energy of P_j is saved. Thus, `PLAIN` node P_j in EECS chooses the cluster head by considering not only saving its own energy but also balancing the workload of cluster heads, i.e. two distance factors: $d(P_j, CH_i)$ and $d(CH_i, BS)$. The pseudo-code for cluster formation is shown in Algorithm 2.

We introduce a weighted function $cost(j, i)$ for the `PLAIN` node P_j to make a decision, which is

$$cost(j, i) = (1 - w(P_j)) \times f(P_j, CH_i) + w(P_j) \times g(CH_i), \tag{9}$$

and P_j chooses CH_i with $\min\{cost\}$ to join. As in Fig. 1, the node i computes the $cost$s of all cluster heads and finds B has $\min\{cost\}$, then it joins the cluster head B, although A is even closer.

In formula 9, f and g are two normalized functions for the distance $d(P_j, CH_i)$ and $d(CH_i, BS)$, respectively:

$$\begin{aligned} f(P_j, CH_i) &= \frac{d(P_j, CH_i)}{d_{f\_max}}, \\ g(CH_i) &= \frac{d(CH_i, BS) - d_{g\_min}}{d_{g\_max} - d_{g\_min}}, \end{aligned} \tag{10}$$

where $d_{f\_max} = exp(\max\{d(P_j, CH_i)\})$, $d_{g\_max} = \max\{d(CH_i, BS)\}$ and $d_{g\_min} = \min\{d(CH_i, BS)\}$. w is the function of P_j as follows:

$$w(P_j) = c + (1 - c)\sqrt{\frac{d(P_j, BS)}{(d_{g\_max} - d_{g\_min})}}. \tag{11}$$

f subfunction in $cost$ guarantees that members choose the closest cluster head in order to minimize the intra-cluster communication cost, while g subfunction makes the nodes join the cluster head with small $d(CH_i, BS)$ to alleviate the workload of the cluster heads farther from the BS. w subfunction is the weighted factor for the tradeoff between f and g. Furthermore, the optimal value of weighted factor c in the subfunction

w depends on the specific network scale and we will make a deeper study in the later work.

```
if ($state$ = HEAD) then
    broadcast HEAD_AD_MSG;
    wait for JOIN_CLUSTER_MSGs;
end if
if ($state$ = PLAIN) then
    receive all HEAD_AD_MSGs;
    compute the $cost$ for each cluster head;
    choose the cluster head with $\min\{cost\}$ and broadcast the JOIN_CLUSTER_MSG;
end if
```

Algorithm 2: cluster formation

4.3 Synchronization Issues

Synchronization between each phase should be guaranteed so that each node has enough time to complete the procedure. For the first phase, we could choose a proper time interval $T1$ according to the system parameters and wireless channel quality; and in the second phase, each cluster head broadcasts a TDMA schedule within its cluster. Then the members process in the corresponding time slots and turn off the radio in the idle time to save energy further. Additionally, we make BS periodically synchronize the nodes over the network against the time drift.

5 EECS ANALYSIS

In this section, we analyze the performance of EECS in detail and explain how to set the parameters T and $R_{compete}$.

Lemma 1. *The overhead*[†] *complexity in transmitting and receiving the control messages across the network is* $O(N)$.

Proof: Observing EECS, every node sends small constant-length control messages each round without iterations. Each `HEAD` node sends three messages which are `COMPETE_HEAD_MSG`, `HEAD_AD_MSG` and `SCHEDULE_MSG`; each `CANDIDATE` node sends two messages which are `COMPETE_HEAD_MSG` and `JOIN_CLUSTER_MSG`; while the others send `JOIN_CLUSTER_MSG`s

---

[†] The overhead we will mention later refers to this “overhead” here.

only. Clearly, the total overhead is as follows, where $k_{exp} = exp(|CH|)$.

$$\begin{aligned} \text{total overhead} &= 3k_{exp} + 2(NT - k_{exp}) + N(1 - T) \\ &= k_{exp} + NT + N \end{aligned} \tag{12}$$

So, the total overhead is $O(N)$ □

Good quality `HEAD` nodes should be guaranteed by enough competition of the `CANDIDATE` nodes. Since T is the only crucial factor which affects the sum of `CANDIDATE` nodes, it must be large enough to guarantee enough `CANDIDATE` nodes. On the other hand, the larger T is, the more overhead is produced in the *cluster head election* phase. So, we must decrease T to reduce the overhead with guaranteed `HEAD` quality. In practice, it is better to slightly overestimate T to guarantee the quality of the cluster heads.

In LEACH, there is no interaction during the *cluster head election*. So the overhead is near optimal, which is $2NP + N(1 - P) = NP + N$, where P is the probability of one node to be `HEAD`. Thus the overhead of EECS is only $(1 + T)/(1 + P)$ times of LEACH when N is large enough. In HEED, `HEAD` nodes are elected with iterations. Although the communication is localized and the algorithm terminates in $O(1)$ iterations, HEED still produces much more overhead with the upper bound $N_{iter} \times N$. Clearly, our approach is better than HEED. The above property shows that the control overhead of EECS is significantly lower and meets the primary goal proposed in Section 3.

Lemma 2. *There is at most one cluster head in every $R_{compete}$ radio covered range.*

Proof: Let S be the set of all sensor nodes. And for $\forall x \in CH$, let $C_x = \{y | d(y, x) \le R_{compete}, y \in S\}$. For contradiction, we assume that there is a node $y \in C_x$ which is also a cluster head. According to the competition metric in *cluster head election*, $x.E_{residual} > z.E_{residual}, \forall z \in C_x$. Since $y \in C_x$, then $x.E_{residual} > y.E_{residual}$. The communication is symmetric in the network model of EECS. If y is the cluster head, $y.E_{residual} > x.E_{residual}$ as x is within the distance $R_{compete}$, which is a contradiction.

So, for $\forall x \in CH$, $\forall y \in C_x$, there is $y \notin CH$. □

Observing the above Lemma, it claims that the distance between any two cluster heads is no less than $R_{compete}$. However, it is worthy to notice that sometimes there may exist a "gradient phenomenon" as in Figure 2, where $S_1.E_{residul} > S_2.E_{residual} > ... > S_5.E_{residual}$. Consequently, S_2, S_3, S_4 and S_5 give up the competition and S_1 is the only winner. We will explain the effect of the gradient phenomenon later in Section 6.4.

In [5], the author proves that there is an optimal number of cluster heads k_{opt} in a given scene. Since EECS is a LEACH-like protocol, we want

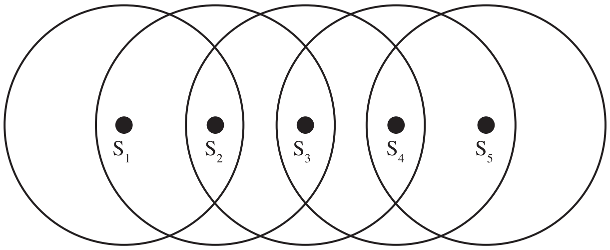


FIGURE 2
The gradient phenomenon in the network.

to elect k_{opt} cluster heads every round. According to *Lemma* 2, $R_{compete}$ affects the cluster heads directly. So we compute the optimized value of $R_{compete}$, denoted by R_{opt} in the following lemma.

Lemma 3. *There is an optimal range R_{opt} for $R_{compete}$, which is $\sqrt{\frac{A}{\pi k_{opt}}}$, where k_{opt} is the optimal range of $|CH|$.*

Proof: Let $P(CANDIDATE)$ be the probability of one node being the `CANDIDATE` node, so the sum of `CANDIDATE` nodes

$$n = P(CANDIDATE) \times N. \tag{13}$$

In the $R_{compete}$ radius range, there are m nodes in `CANDIDATE` state (boundary cases are ignored), where

$$m = n \times \frac{\pi R_{compete}^2}{A}. \tag{14}$$

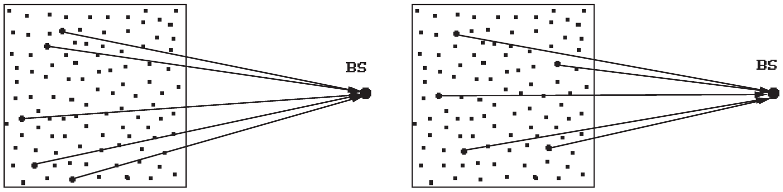
Since all nodes have the same capacity, these m nodes have equal probability to be `HEAD`, then the probability of one node being `HEAD` node is:

$$\begin{aligned} P(HEAD) &= P(HEAD|CANDIDATE) \times P(CANDIDATE) \\ &= \frac{1}{m} \times P(CANDIDATE) \\ &= \frac{A}{\pi R_{compete}^2 N}. \end{aligned} \tag{15}$$

So, the expectation of the sum of cluster heads is

$$exp(|CH|) = N \times P(HEAD) = \frac{A}{\pi R_{compete}^2}. \tag{16}$$

In order to optimize energy consumption, we want to let $exp(|CH|)$ equal to $k_{opt} = \frac{\sqrt{N}}{\sqrt{2\pi}}\sqrt{\frac{\epsilon_{fs}}{\epsilon_{mp}}\frac{M}{d_{toBS}^2}}$ in [5]. Combining the induction in [5] and



(a) non-uniform distribution (b) uniform distribution

FIGURE 3
Distribution characteristics of cluster heads (a) non-uniform distribution (b) uniform distribution.

equation (16), we can find the optimal radius R_{opt} is

$$R_{opt} = \sqrt{\frac{A}{\pi k_{opt}}} = \sqrt{\frac{M\sqrt{2\epsilon_{mp}}}{\sqrt{N\pi\epsilon_{fs}}}d_{toBS}}. \tag{17}$$

□

In LEACH, cluster heads are elected simply at random. As a result, the distribution of the cluster heads is not insured and may be non-uniform as in Figure 2(a). Some members have to expend much more energy to communicate with the corresponding cluster heads far away. The last two lemmas show that there is one and only one cluster head within any $R_{compete}$ with high probability. Thus, the cluster heads in EECS are distributed evenly as in Figure 2(b).

Lemma 4. *The complexity of node information stored in each node is* $O(\sqrt{N})$.

Proof: Here, we prove this lemma according to the role of each node respectively.

Case 1: If the node is a `HEAD` node, it stores no more than $\frac{N}{k_{opt}}T$ competing nodes' information and about $\frac{N}{k_{opt}}$ members' information as well. Thus the complexity of node information stored in a `HEAD` node is $O(\sqrt{N})$.

Case 2: If the node is a `CANDIDATE` node[‡], it stores at most $\frac{N}{k_{opt}}T$ competing nodes' information during the competing step and about k_{opt} `HEAD` nodes' information in the cluster formation step. Therefore, the complexity of node information stored in a `CANDIDATE` node is $O(\sqrt{N})$.

Case 3: If the node is a `PLAIN` node, it only stores about k_{opt} `HEAD` nodes' information during the entire procedure. So the complexity of nodes' information stored in a `PLAIN` node is $O(\sqrt{N})$.

[‡] The CANDIDATE node here refers to the node that fails in the competition.

TABLE 3
Parameters of Simulations

Parameter	**Value(scene 1)**	**Value(scene 2)**
Area	(0, 0) $\sim$ (100, 100)	(0, 0) $\sim$ (200, 200)
Location of BS	(50, 200)	(100, 350)
N	400(600)	1000(1500)
Initial energy	$0.5J$	$1.0\ J$
E_{elec}	$50\ nJ/bit$	
ϵ_{fs}	$10\ pJ/bit/m^2$	
ϵ_{mp}	$0.0013\ pJ/bit/m^4$	
$d_{crossover}$	$87\ m$	
E_{DA}	$5\ nJ/bit/signal$	
Packet size	$4000\ bits$	

Therefore, we conclude that the complexity of node information stored in each node is $O(\sqrt{N})$. □

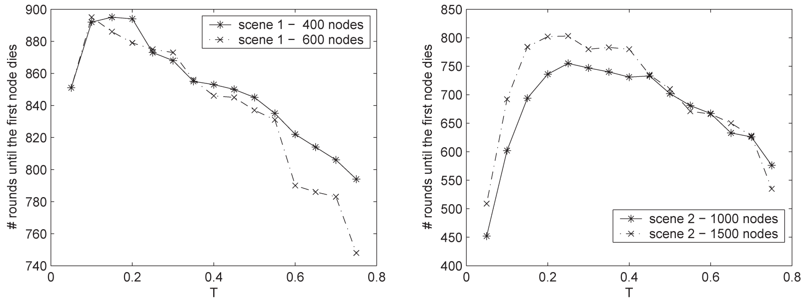
Observing Lemma 4, EECS utilizes only $O(\sqrt{N})$ node information for each node to form the cluster autonomously, which meets the requirement proposed in Section 3.3.

6 SIMULATION

In this section, we evaluate the performance of the EECS protocol implemented with MATLAB. In the simulation, we adopt the same MAC protocols in EECS as in LEACH. For simplicity, we assume the probability of signal collision and interference in the wireless channel is ignorable. In order to explain the relation between the network scale and the parameters in EECS, we run each kind of simulations in two different scenes, i.e., a normal scale scene (scene 1) and a large scale scene (scene 2) respectively.

The radio transmitter, radio amplifier and data fusion unit are the main energy consumers of a sensor node, so we calculate the energy consumption of these three components in the simulation. The parameters of simulations are listed in TABLE 3, where the parameters of the energy consumption model are the same as LEACH [5]. Unless otherwise specified, every simulation result shown below is the average of 100 independent experiments where each experiment uses a different randomly-generated uniform topology of sensor nodes.

Lifetime is the criterion for evaluating the performance of sensor networks. In the simulation, we measure the lifetime in terms of the round when the first node dies, because in data gathering applications a certain area cannot be monitored any more once a node dies. We use the energy utilization rate



(a) normal scene (b) large scene

FIGURE 4
The impact of T on the network lifetime: (a) normal scene, (b) large scene.

to evaluate the efficiency of energy consumption, which is defined as the ratio of the total energy consumed when the first node dies to the initial total energy. A high energy utilization rate implies that energy consumption is distributed well across the network and any node will not deplete its energy very quickly.

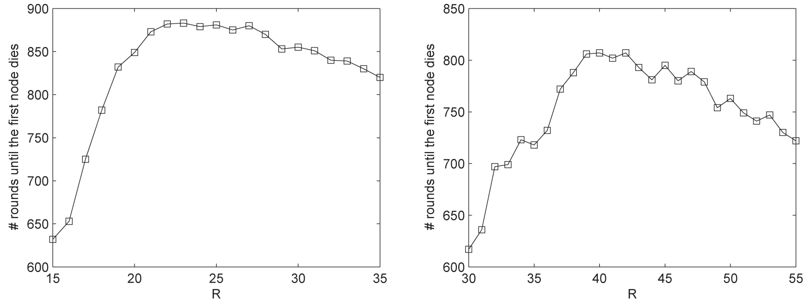
According to equation 10 and the network scale, we set $f(d) = \frac{d}{100}$, $g(d) = \frac{d-100}{200-100}$ in scene 1, and $f(d) = \frac{d}{100}$, $g(d) = \frac{d-150}{350-150}$ in scene 2.

6.1 Experiment of T

We first examine the impact of T on the network lifetime, as the scales are different. We have done two independent experiments in different scales. In normal scale, $N = 400, 600$, $R_{compete} = 26, 22$, $c = 0.8$; in large scale, $N = 1000, 1500$, $R_{compete} = 40, 35$, $c = 0.6$. As T varies from 0.05 to 0.75, Figure 4 shows the relation between T and the network lifetime. There is an optimal range for the value of T, which is about $0.1 \sim 0.3$ in the given scene. According to the explanation about T in Section 5, T must be properly set with guaranteed `HEAD` quality and low overhead. Another point needed to be mentioned that the optimal value T_{opt} decreases when the network density increases. It can be explained that there is an optimal sum of `CANDIDATE` nodes in a given network coverage size.

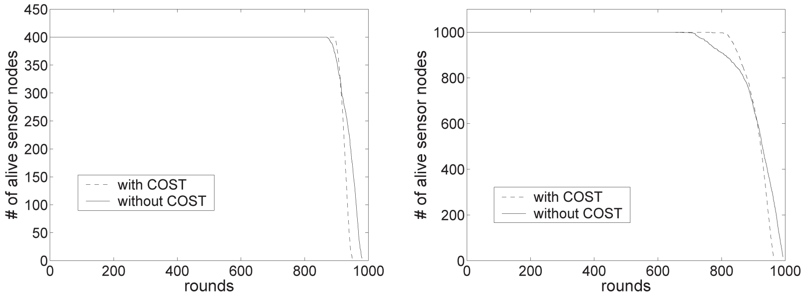
6.2 Experiment of $R_{compete}$

In the experiment shown in Figure 5, we demonstrate *Lemma* 3 by observing the relation between $R_{compete}$ and the network lifetime. In scene 1, $N = 400$ and $k_{opt} = 4 \sim 7$, so the optimal value R_{opt} is between $21 \sim 28$; In scene 2, $N = 1000$ and $k_{opt} = 6 \sim 10$, so R_{opt} is between $36 \sim 46$. Observing the impact on network lifetime when $R_{compete}$ varies, Figure 5 suggests that the optimal value of $R_{compete}$ is about 25 in scene 1 and about 40 in scene 2. Both results fall into the optimal range computed prior.



(a) normal scene (b) large scene

FIGURE 5
The impact of $R_{compete}$ on the network lifetime: (a) normal scene, (b) large scene.



(a) normal scene (b) large scene

FIGURE 6
The impact of *cost* on the network lifetime: (a) normal scene, (b) large scene.

6.3 Experiment of the Weighted Function *cost*

In Figure 6, the experiment shows the efficiency of *cost* introduced to balance the load among the cluster heads, where the dash line denoted as the method without considering the cluster heads' load balance issue. We set c at 0.8 in the scene 1 and 0.6 in the scene 2 respectively. Comparing the $without - cost$ method ($c = 1$) with the $with - cost$ method, we find that the *cost* indeed extends the network lifetime. The value of c is determined by the specific scene. While the network grows larger, the difference among $d(CH_i, BS)$s impacts the load balance among the cluster heads more and more distinctly. So c should be decreased and the `PLANE` node will consider more about the load of cluster head when joining the cluster. That's why the value of c is bigger in scene 1 than in scene 2. What's more, as shown in Figure 6, the *cost* performs more effectively in scene 2 than in scene 1. The simulation shows that the unequal clustering mechanism with *cost*

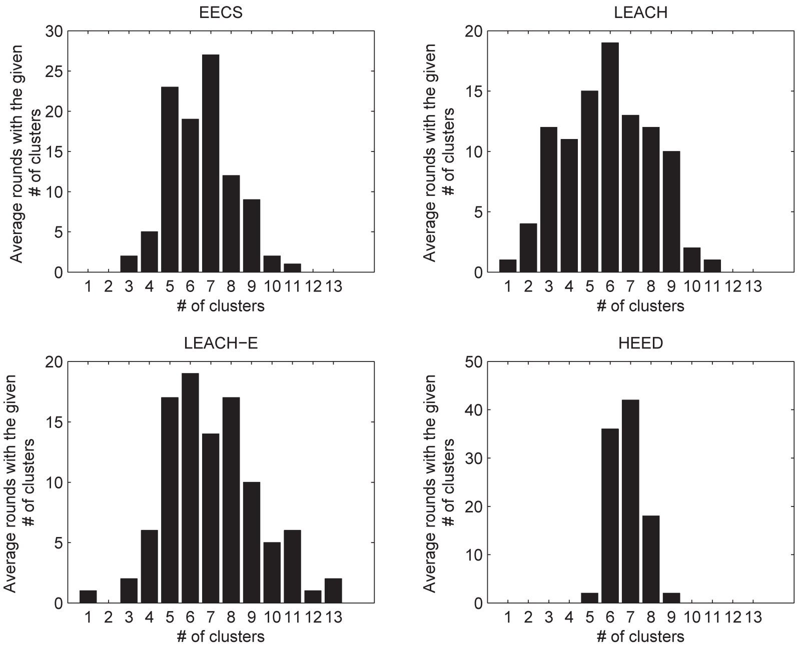


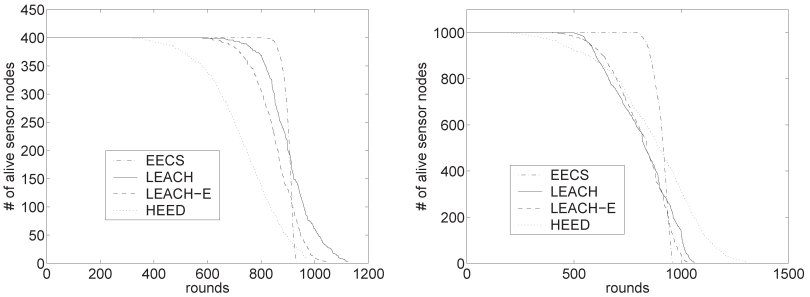
FIGURE 7
The number of clusters in each round in EECS, LEACH, LEACH-E and HEED (scene 1).

indeed further prolongs the network lifetime especially in the large scale scenario. In this paper, the *cost* function is simple, and we will optimize the *cost* function in the next work.

6.4 Performance Comparison

Finally, we make the performance comparison in two categories of scenario, i.e. homogeneous (the initial energy of each node is identical) and heterogenous (the initial energy of each node is arbitrary). Comparing with LEACH, energy-aware LEACH (LEACH-E) and HEED, the simulation results show that EECS performs best in both homogeneous and heterogeneous scenarios and prolongs the network lifetime significantly.

In the homogeneous scenario, we set $k_{opt} = 6$, $T = 0.2$, $R_{compete} = 26$ and $c = 0.8$ in the normal scene and $k_{opt} = 9$, $T = 0.15$, $R_{compete} = 40$ and $c = 0.6$ in the large scene respectively. In Figure 7, it exhibits the distribution of the number of clusters in randomly selected 100 rounds in EECS, LEACH, LEACH-E and HEED respectively. Shown in the figure, the number of clusters varies widely in each simulation run in LEACH and LEACH-E; on the other hand, the cluster number varies narrowly at the k_{opt} range in EECS and HEED. In LEACH and LEACH-E, the clusters in each

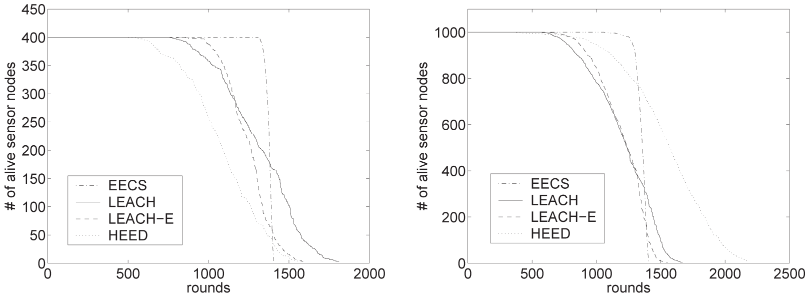


(a) normal scene (b) large scene

FIGURE 8
Performance comparison in homogeneous scenario: (a) normal scene, (b) large scene.

round is not controlled although the expectation is aware; while in EECS and HEED, localized competition achieves well distributed cluster heads over the network in all probability in each round. Through the experiment of EECS, we find that the node close to the BS plays `HEAD` role even more times than the node far from the BS. Because of the different $d(x, BS)$, the node close to the BS expends smaller energy than the node far from the BS when it becomes `HEAD` and has even more residual energy which results in the "gradient phenomenon". When considering the residual energy, the gradient phenomenon is reasonable and balances the energy consumption even better.

Figure 8 shows the variation of total number of sensors still alive in the homogeneous scenario when the simulation time elapses. In [8], the author claims that HEED performs better than LEACH with adding two more assumptions: 1) residual energy is aware through the network, and 2) a node can only select a cluster head in its cluster range proximity. Due to the proximity constraint, uncovered nodes have to elect themselves as cluster heads and LEACH produces much more cluster heads than expected. However, in this paper, the assumptions in EECS are identical with LEACH [5]. In the figure, we find LEACH and LEACH-E perform better than HEED, and EECS performs the best. In HEED, the distribution of cluster heads is fairly well spatially each round. The node far from the BS exhausts much more quickly than the node close to the BS. Thus well distribution spatially can not always achieve load balance over the entire network. And it is also the reason why we use "gradient phenomenon" to balance the energy consumption in EECS. Since LEACH-E is proposed to cope with the heterogenous scenario, and the overhead for propagating the energy information over the network some what affects the system performance, LEACH-E performs no better (even worse) than LEACH in homogeneous scenario. In scene 1, EECS prolongs the lifetime by more than 35% against LEACH. The energy utilization rate is about 93% in



(a) normal scene (b) large scene

FIGURE 9
Performance comparison in heterogenous scenario: (a) normal scene, (b) large scene.

EECS, while only 53% in LEACH. The reason for this is that EECS always achieves well distributed cluster heads according to the residual energy; further, we consider to balance the load among the cluster heads with weighted function. In Figure 8b, the efficiency of EECS is more distinct when the network scale grows.

In the heterogenous scenario, the initial energy varies from $0.5J \sim 1J$ in the small scale scene; while in the large scale scenario, it varies from $1J \sim 2J$. As shown in Figure 9, HEED improves a little against itself in the homogenous scenario, but performs no better than LEACH. That's because, in HEED, energy affects the tentative cluster head set only; and it is not the deterministic factor to the policy of cluster head election. With considering the residual energy of each node, both EECS and LEACH-E extend the network lifetime over the original LEACH. Moreover, EECS performs greatly better than LEACH-E, since the cluster heads in EECS is distributed much better than LEACH-E (Actually, there is no consideration about the distribution of cluster heads in LEACH-E). The results of simulation show that EECS performs the best and prolongs the network lifetime significantly over LEACH, LEACH-E and HEED.

In order to save energy further, multi-hop communication among the cluster heads is adopted during the inter-cluster communications in the data transmission phase. Notice that we focus on the cluster set-up algorithm but not the data transmission approach in our current work. Future work will consider the multi-hop technique in inter-cluster communication. Readers should refer to [16] for details about multi-hop routing in clustered networks. What's more, broadcast can be considered as a reverse function of data gathering, where a data item from a source is sent to all nodes in the network. Energy efficient broadcasting has also been extensively studied in the literature, such as [17] provides a good survey of energy-efficient broadcasting and energy models.

7 CONCLUSION AND FUTURE WORK

In this paper, we present a novel, autonomous, energy efficient and load balanced clustering scheme applied for periodical data gathering applications. EECS produces a near uniform distribution of cluster heads across the network through localized communication with little overhead. What's more, a novel approach has been introduced to distribute the energy consumption among the cluster heads in the *cluster formation* phase. Simulation results show that EECS prolongs the network lifetime significantly and the total energy is more efficiently consumed.

All of our contributions here are focused on the cluster set-up stage. There is still much room to improve the performance of data transmission. In large scale sensor networks, multi-hop communication is a mainstream technique for energy saving. We remove the assumption of single-hop and design an energy efficient protocol for both intra-cluster and inter-cluster data transmission in EEUC [18]. Interested readers are encouraged to refer to our another work [18] for more details.

ACKNOWLEDGMENTS

This work is supported by the China NSF grant, the China Jiangsu Provincial NSF grant (BK2005208), the China 973 project (2002CB312002) and TRAPOYT award of China Ministry of Education, US NSF grants ANI 0073736, CCR0329741, CNS 0422762, CNS 0434533, and EIA 0130806.

REFERENCES

[1] Estrin, D., Girod, L., Pottie, G. and Srivastava, M. (2001). Instrumenting the world with wireless sensor networks. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP).*

[2] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y. and Cayirci, E., (2002). A Survey on Sensor Networks. *IEEE Communications Magazine*, August, 102–114.

[3] Krishnamachari, B., Estrin, D. and Wicker, S. (2002). The impact of data aggregation in wireless sensor networks. *Proceedings of the 22nd International Conference on Distributed Computing Systems Workshops*.

[4] Heinzelman, W., Chandrakasan, A. and Balakrishnan, H. (2000). Energy-Efficient Communication Protocol for Wireless Microsensor Networks. *Proceedings of the 33rd International Conference on System Sciences (HICSS'00).*

[5] Heinzelman, W., Chandrakasan, A. and Balakrishnan, H. (2002). An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications*, 1(4), 660–669.

[6] Lee, S. H., Yoo, J. J. and Chung, T. C. (2004). Distance-based Energy Efficient Clustering for Wireless Sensor Networks. *Proceedings of the 29th Annual IEEE international Conference on Local Computer Networks(LCN'04).*

[7] Depedri, A., Zanella, A. and Verdone, R. (2003). An Energy Efficient Protocol for Wireless Sensor Networks. 2005, from http://www.ee.oulu.fi/carlos/WSNPapers/DE03.pdf.

[8] Younis, O. and Fahmy, S. (2004). HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks. *IEEE Transactions on Mobile Computing*, 2004, 3(4), 660–669.

[9] Choi, W., Shah, P. and Das, S. (2004). A Framework for Energy-Saving Data Gathering Using Two-Phase Clustering in Wireless Sensor Networks. *Proceedings of the First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous'04).*

[10] Bandyopadhyay, S. and Coyle, E. (2003). An Energy-Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks. *Proceedings of IEEE INFOCOM'03*.

[11] Chan, H. and Perrig, A. (2004). ACE: An Emergent Algorithm for Highly Uniform Cluster Formation. *Proceedings of the First European Workshop on Sensor Networks (EWSN).*

[12] Kamimura, J., Wakamiya, N. and Masayuki, M. (2004). Energy-Efficient Clustering Method for Data Gathering in Sensor Networks. *Proceedings of the Annual International Conference on Broadband Networks*.

[13] Labroche, N., Monmarché, N. and Venturini, G. (2002). A new clustering algorithm based on the chemical recognition system of ants. *Proceedings of ECAI 2002*.

[14] Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D. E. and Pister, K. S. J. (2000). System Architecture Directions for Networked Sensor. *Architectural support for Programming Languages and Operating Systems*, 93–104.

[15] Rappaport, T. (1996). Wireless Communication: Principles & Practice. Prentice-Hall.

[16] Mhatre, V. and Rosenberg, C. (2004). Design guidelines for wireless sensor networks: communication, clustering and aggregation. *Ad Hoc Networks Journal, Elsevier Science*, 2(1), 45–63.

[17] Simplot-Ryl, D., Stojmenović, I. and Wu, J. (2005). Energy Efficient Broadcasting, Activity Scheduling and Area Coverage in Sensor Networks. *Handbook of Sensor Networks: Algorithms and Architectures*, I. Stojmenovic (ed.), Wiley, 343–379.

[18] Li, C., Ye, M., Chen, G. and Wu, J. (2005). EEUC: An energy efficient unequal clustering mechanism for wireless sensor networks. *accepted to appear in IEEE MASS'05*.