Available online at www.sciencedirect.com
ScienceDirect
J. Parallel Distrib. Comput. 67 (2007) 1187 – 1200
Journal of Parallel and Distributed Computing
www.elsevier.com/locate/jpdc

Distributed clustering algorithms for data-gathering in wireless mobile sensor networks

Chuan-Ming Liu[a,*], Chuan-Hsiu Lee[a], Li-Chun Wang[b]

[a]*National Taipei University of Technology (NTUT), Taiwan*
[b]*National Chiao Tung University, Taiwan*

Received 17 October 2005; received in revised form 27 June 2007; accepted 29 June 2007

Abstract

One critical issue in wireless sensor networks is how to gather sensed information in an energy-efficient way since the energy is a scarce resource in a sensor node. Cluster-based architecture is an effective architecture for data-gathering in wireless sensor networks. However, in a mobile environment, the dynamic topology poses the challenge to design an energy-efficient data-gathering protocol. In this paper, we consider the cluster-based architecture and provide distributed clustering algorithms for mobile sensor nodes which minimize the energy dissipation for data-gathering in a wireless mobile sensor network. There are two steps in the clustering algorithm: cluster-head election step and cluster formation step. We first propose two distributed algorithms for cluster-head election. Then, by considering the impact of node mobility, we provide a mechanism to have a sensor node select a proper cluster-head to join for cluster formation. Our clustering algorithms will achieve the following three objectives: (1) there is at least one cluster-head elected, (2) the number of cluster-heads generated is uniform, and (3) all the generated clusters have the same cluster size. Last, we validate our algorithms through an extensive experimental analysis with Random Walk Mobility (RWM) model, Random Direction Mobility (RDM) model, and a Simple Mobility (SM) model as well as present our findings.

Keywords: Wireless sensor networks; Mobility; Clustering; Data-gathering; Energy efficiency

1. Introduction

Wireless sensor networks have attracted much research attention in recent years and can be used in many different applications, including battlefield surveillance, machine failure diagnosis, biological detection, inventory tracking, home security, smart spaces, environmental monitoring, and so on [1,4]. A wireless sensor network consists of a large number of tiny, low-power, cheap sensor nodes having sensing, data processing, and wireless communication components. It has not only the ability to sense some phenomena in the interested region but also the network features, thereby representing an improvement over the traditional sensor systems.

The sensor nodes in a wireless sensor network are usually deployed randomly inside the region of interest or close to it. A remote *base station* (BS) connected to the Internet is engaged to give commands to all the sensor nodes and gather information from the sensor nodes. In addition to sensing, the wireless sensor nodes can process the acquired information, transmit messages to the BS, and communicate to each others. An architecture of the wireless sensor network is depicted in Fig. 1.

Since the sensor nodes are randomly scattered in the sensor field, a wireless sensor network should have the capability of self-organizing. This indicates that the wireless sensor nodes are more autonomous than the traditional sensors. Hence, it becomes more challenging to design related protocols or algorithms in such a distributed environment. Besides, wireless sensor nodes have many limitations, including modest processing power, little storage, short communication range, and limited power source. These limitations also make designing wireless sensor network protocols difficult. Because wireless sensor nodes are low-powered, the constraint on the power consumption is an important issue when designing wireless sensor network protocols. Furthermore, the node mobility imposes more

---

[*] Corresponding author. Fax: +886 2 87732945.
E-mail address: cmliu@csie.ntut.edu.tw (C.-M. Liu).

0743-7315/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.jpdc.2007.06.010

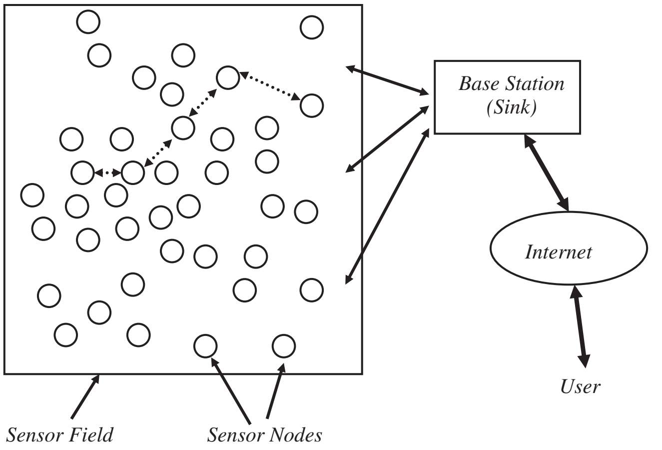


Fig. 1. The architecture of a wireless sensor network in which the sensor nodes are deployed randomly into the interested area (sensor field) and the BS (sink) connects to the Internet.

difficulties on designing the protocols since the topology of the network changes frequently.

As the architecture shown in Fig. 1, the BS connected to the Internet is usually much more powerful than sensor nodes and has power supplied. So, it is possible for the BS to send the commands by broadcasting. On the other hand, the BS needs to collect the sensed information from the sensor nodes and send it back to the user. *Data-gathering* (collecting the sensed information from the sensor nodes and routing the sensed information) hence raises an important topic in wireless sensor networks due to the power limitation of sensor node. To have an efficient data-gathering protocol in terms of the energy consumption is an on-going research work and is the core of this paper.

Many protocols have been proposed for data-gathering or communication among wireless sensor nodes [5,8,9, 12,14–17, 19,20,23,24,28,30,32,33]. Most of the proposed protocols worked on static wireless sensor networks. Nevertheless, in many applications, the sensor nodes can move either by outside force or its mobility component. For example, the sensor nodes attached to moving objects for tracking or scattered on the sea. Among the existing protocols, the cluster-based structure provides an effective architecture for data-gathering in wireless sensor networks. This paper considers such an architecture and proposes distributed clustering algorithms for the cluster-based data-gathering protocol in a wireless mobile sensor network.

There is no explicit mobility model proposed and studied in wireless mobile sensor networks so far. Since a wireless mobile sensor network is basically organized into a wireless ad hoc network, we refer to the mobility models used in mobile ad hoc networks. Furthermore, the mobility pattern of a sensor node in a wireless mobile sensor network can be programmed in advance before scattering the sensor nodes. We therefore select the following three mobility models to evaluate the proposed data-gathering protocols: Random Walk Mobility (RWM) model [7], Random Direction Mobility (RDM) model [26], and Simple Mobility (SM) model [21,22]. We will give more details about these mobility models in Section 2.

In order to measure the performance of the proposed distributed algorithms in terms of energy consumption, we consider the system lifetime of a wireless sensor network. The *system lifetime* is the duration in which the wireless sensor network lasts and relates to (1) the energy consumption and (2) the quality of service. An energy-efficient data-gathering protocol can make the wireless sensor network live longer. If the network can live longer, one does not need to reset a new wireless sensor network (re-scatter the sensor nodes) so often.

The rest of this paper is organized as follows. We give the background and related work in Section 2. In Section 3, we introduce the problems. The proposed clustering algorithms are discussed in Sections 4 and 5, respectively. We then implement these algorithms and compare the performance in terms of energy consumption in Section 6 by simulation. Then we conclude in Section 7.

2. Related work

In wireless sensor networks, sensor nodes use multi-hop communication to avoid consuming a large amount of energy for sending messages directly to the BS. Due to the multi-hop communication, sensed information may accumulate too much for end-user to process. For a sensor node, to transmit all the information it receives and its own sensed information may also consume much power. Therefore, automated methods of combining or aggregating the data into a small set of meaningful information are required [14]. Such methods depend on the applications. In this paper, we assume that the sensor nodes can do the *data-aggregation* (*data-fusing*) and focus on how the mobile sensor nodes route sensed information to the BS.

There are two categories for the existing data-gathering protocols: *hierarchy* (*cluster-based*) protocols and *non-hierarchy* protocols. The non-hierarchical protocols include Flooding [12,31], Gossiping [10], Directed Diffusion [15], SPIN [12], and REAR [9]. On the other hands, the hierarchical protocols include SMECN [19], SAR [29], Data funneling [24], LEACH [14], PEGASIS [20], HIT [6], LPT [17], and ACW [30,32]. Furthermore, the communication patterns for wireless sensor networks take one of two general forms:

- Time-driven (periodical) transmissions: periodical transmissions from all the sensor nodes, and
- Event-driven transmissions: reports from only those sensor nodes that observe a specific event.

In [18], the authors pointed out that the tree-based structure is a proper architecture for collecting data in real static wireless sensor networks. A tree-based structure can be organized by many clusters with multiple layers. However, in a wireless mobile sensor network, the topology of the network changes frequently. A tree-based structure for the static environment cannot be directly applied to a mobile environment. To retain the merits of the tree-based structure, we consider a cluster-based architecture which is a tree-based architecture with two layers for data-gathering. A cluster-based (or hierarchical)

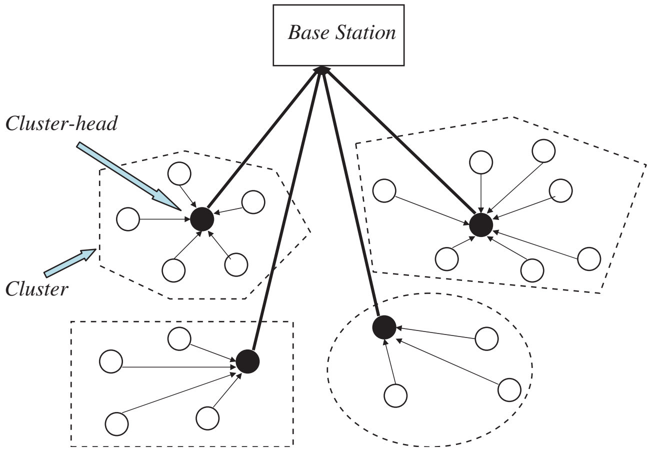


Fig. 2. A snapshot of the cluster-based architecture for wireless sensor networks where the non-cluster-head sensor nodes (unfilled circles) send the sensed information to the cluster-heads (filled circles) and the cluster-heads fuse the information received and transmit the fused message to the BS.

data-gathering protocol consists of a series of *rounds*. In each round, there are two major phases: (1) *clustering* and (2) *message transmission* phases. In Phase (1), the entire sensor network will be partitioned into different clusters. Each cluster consists of one *cluster-head* and a number of sensor nodes. There are usually two steps in Phase (1): *cluster-head election* step and *cluster formation* step. After all the clusters are organized, the cluster-head aggregates data from the sensor nodes in the cluster and then transmits information to the BS directly in Phase (2). Fig. 2 captures the operations in a round and presents a cluster-based mechanism. This paper focuses on Phase (1).

We observe that the mobility impacts on the performance during the clustering phase. In [21,22], the authors discussed the effects of different cluster formation mechanisms in mobile sensor networks using a SM model. In the SM model, each sensor node randomly selects a speed and direction among four pre-defined directions initially. Then, a sensor node moves according to the fixed speed and direction. If a sensor node reaches the boundary of the sensing area, it then reverses its direction with the same speed. In ad hoc networks, many mobility models have been presented [3]. In this paper, we also consider the RWM model [7] and RDM model [26]. In the RWM model, a sensor node can randomly choose a speed within a given range and a direction to move from its current location to a new location. After moving a constant time or a constant distance traveled, each sensor node will calculate a new speed and direction to modify its current position. In the RDM model, each sensor node has a constant speed and can randomly choose a direction to travel. A sensor node can choose a new angular direction (between 0° and 180°), if the boundary of the sensing area is reached.

Although many cluster-based routing protocols have been proposed [13,20,30], the problem about clustering the sensor nodes is realized and discussed in recent years [32,34,35]. A better clustering algorithm may average the workload on each sensor node and results in a longer system lifetime. If the number of cluster-heads generated in each round changes dramatically, some sensor nodes will use up all of their energy quickly due to the heavy workload. Therefore, the system lifetime is deteriorated. If there is no cluster-head elected in a round, each sensor node will communicate directly to the BS and consumes a large amount of energy. The other factor that impacts the energy consumption is the *cluster size*. The cluster size is the number of sensor nodes in a cluster. If the difference in the cluster size is big among all the generated clusters, the cluster-head having large cluster size will consume a larger amount of energy than the others. As a result, the system lifetime becomes short. In summary, a good clustering algorithm should achieve the following three objectives:

1. there is at least one cluster-head elected in each round,
2. the number of cluster-heads generated in each round should be uniform, and
3. all the generated clusters should have the same cluster sizes.

3. Problems

In this paper, we consider the wireless mobile sensor networks where:

- The BS is fixed and located far away from the sensor nodes.
- All the sensor nodes are mobile, homogeneous, and power limited.
- Each sensor node is equipped with a Location Finding System [2,27].
- All sensor nodes are time-synchronized [4].

Fig. 3 shows the components of a mobile sensor node considered in this paper.

Recall that there are two steps in the clustering phase: one step is to elect the cluster-heads and the following step is to form the clusters. In this paper, we first propose two efficiently distributed algorithms to elect the cluster-heads among mobile sensor nodes in terms of the energy consumption. The proposed algorithms will achieve the objectives mentioned in Section 2. We then provide a mechanism to form the clusters among the mobile sensor nodes. If a sensor node neglects its mobility when deciding which cluster-head to join, it may select a cluster-head with a longer distance to it, thereby consuming more power.

The rule for deciding the cluster-heads in LEACH provides a distributed way to elect cluster-heads and can be fully applied to a distributed environment like sensor networks. However, when applying such a rule, the difference in the number of the cluster-heads between the rounds may be big and it also happens that there is no cluster-head at all in some rounds. The cluster-head election strategy in LEACH uses the following threshold to be the probability of being a cluster-head for each sensor node:

$$T_v = \begin{cases} \dfrac{P}{1 - P\left(r \bmod \dfrac{1}{P}\right)} & \text{if } v \in G, \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where r is the number of rounds that have passed, P the desired percentage of cluster-heads, and G the set of sensor nodes not

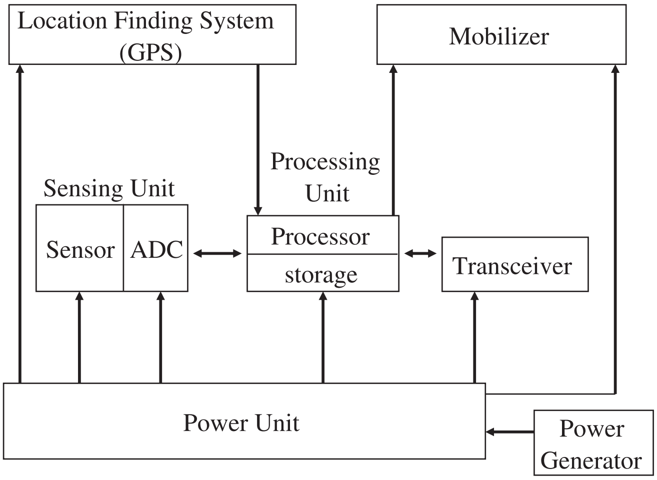


Fig. 3. The components of a wireless mobile sensor node.

being a cluster-head yet. From Eq. (1), a sensor node is elected to be a cluster-head according to the accumulative times of not being a cluster-head.

The above threshold provides a good mechanism for each sensor node to determine to be a cluster-head or not independently. In each round, the threshold is independent from the actual number of cluster-heads elected in the previous rounds. However, it is not necessary to exactly have the desired percentage of cluster-heads in every round. Hence, the above threshold is an ideal condition. For example, consider that there are 100 sensor nodes and the number of cluster-heads elected in each round is expected to be five. The optimum number of cluster-heads that should be elected in a wireless sensor network had been discussed in [11,14]. We will not discuss this here. Suppose that, in the first round, there are only three cluster-heads elected. Then, there are 97 sensor nodes left in the second round. By applying (1), the expected number of cluster-heads elected is therefore greater than 5. Besides, we can further derive the probability of no cluster-head elected in a round as

$$P_f(r) = \left(1 - \frac{P}{1 - P\left(r \bmod \dfrac{1}{P}\right)}\right)^{|G|}. \tag{2}$$

By Eq. (2), we can conclude that, using such a threshold, there can be no cluster-head elected in some rounds and the number of cluster-heads elected in a round is related to the total number of cluster-heads elected in the previous rounds. In summary, the mechanism (i.e. Eq. (1)) used in LEACH for clustering the sensor nodes fails to achieve the objectives discussed in the previous section. In Section4, we provide and discuss different distributed algorithms for cluster-head election which can achieve the mentioned objectives simultaneously.

In the experiments, we measure the energy efficiency by the system lifetime in terms of the number of rounds which a system experiences. We use the radio model used in [14] where the radio dissipates $E_{\text{elec}} = 50\,\text{nJ/bit}$ to run the transmitter or receiver circuitry and $\varepsilon_{\text{amp}} = 100\,\text{pJ/bit/m}^2$ for the transmitter amplifier. The radios have power control and can expend the minimum required energy to reach the intended recipients. We also assume an r^2 energy loss due to channel transmission [25]. Using this radio model, the transmission cost and receiving cost for a l-bit message with a distance r are

- Transmission: $E_{Tx}(l, r) = E_{\text{elec}} \times l + \varepsilon_{\text{amp}} \times l \times r^2$, and
- Receiving: $E_{Rx}(l, r) = E_{\text{elec}} \times l$.

Receiving data is also a high cost operation; therefore, the number of receptions and transmissions should be minimized to reduce the energy cost of a application. We use the data-fusion model discussed in [14] where the processor dissipates $E_{\text{D}} = 5\,\text{nJ/bit}$ to fuse the gathering data.

4. Algorithms for cluster-head election

In this section, we consider the first step in the clustering phase. Two simple but effective distributed algorithms for cluster-head election are proposed. Both of them can guarantee that there is at least one cluster-head elected and the number of cluster-heads in each round is equal. The basic idea of the first algorithm is to determine the cluster-heads by counting. The other distributed algorithm determines the cluster-heads by location. Recall that, in [11,14], the optimal number of cluster-heads has been discussed and suggested. We refer to that conclusion and will not discuss this further in this paper. The simulation in Section 6 will show that the proposed algorithms can achieve the three objectives mentioned in Section 2 and outperform than the approach used in LEACH in terms of energy efficiency.

4.1. Cluster-head election by counting

This subsection describes the distributed **a**lgorithm of **c**luster-head **e**lection by **c**ounting, **Algorithm ACE-C**. Suppose that the number of sensor nodes in a mobile sensor network is N and we number the sensor nodes from 0 to $N-1$. Each sensor node hence can use the assigned number as an unique identifier (ID) in the sensor network. We assume that there are C clusters in each round. Using the ID's, algorithm ACE-C elects the sensor nodes as the cluster-heads in a round-robin fashion. In other words, in the first round, the sensor nodes with ID's from 0 to $C-1$ are the cluster-heads. In the second round, the sensor nodes with ID's from C to $2C-1$ are the cluster-heads. The algorithm continues for the following rounds. After $\frac{N}{C}$ rounds, each sensor node has been the cluster-head once, and the whole process starts over from the sensor node with $\text{ID} = 0$.

To adapt this approach in a distributed way and allow each sensor node to decide independently whether it is a cluster-head in the current round, a sensor node will keep the total number of cluster-heads generated so far in each round and use its ID to decide. If it is the time for it to be one of the cluster-heads in the current round, it will claim that it is a cluster-head to all the other sensor nodes. Otherwise, it will be a non-cluster-head node. The algorithm ACE-C executed at each sensor node v hence can be stated as follows. Each sensor node v will copy

```
Algorithm of Cluster-head Election by Counting
Input: $C$: the number of cluster-heads in around;
       $N$: the total number of sensor nodes
(1) $N_{CH} = 0$ /* number of cluster-heads */
(2) while $N_{CH} < C$ do
      (2.1) $v_{id} = (vid+1) \bmod N$
      (2.2) if ($v_{id} = 0$) then
              (2.2.1) $v$ is a cluster-head
              (2.2.2) broadcast an advertisement message
              (2.2.3) increase $N_{CH}$ by 1
            endif
      (2.3) wait a period of time to receive the advertisement message
      (2.4) if (a message is received) then
              (2.4.1) increase $N_{CH}$ by 1
            endif
    endwhile
End
```

Fig. 4. Algorithm ACE-C used in each sensor node v for cluster-head election in each round.

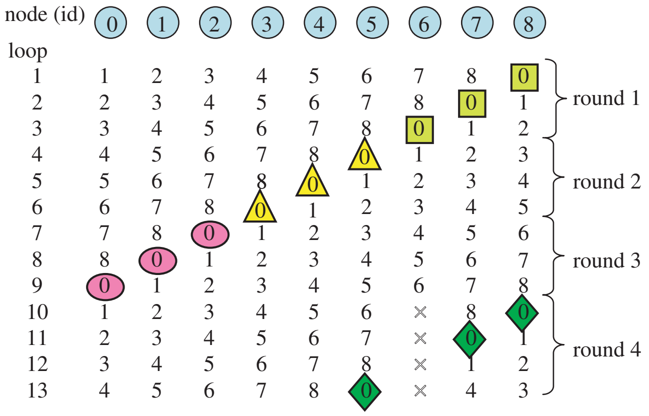
its ID to a variable v_{id}. This variable is used to determine if it is the turn for a sensor node to be a cluster-head. Initially, sensor node v sets the total number of cluster-heads, N_{CH}, to be 0. Then, it repeatedly executes the following steps. The v_{id} is first increased by one. The algorithm then considers the remainder v_{r} of v_{id} divided by N. There are two cases:

- When $v_{\mathrm{r}} = 0$, this case indicates that it is the turn for sensor node v to be a cluster-head. Sensor node v then broadcasts an advertisement message to all the other sensor nodes.
- When v_{r} is not 0, this case denotes that sensor node v is not a cluster-head. It will wait a period of time for receiving an advertisement message.

After the decision, sensor node v increases the total number of cluster-heads generated, N_{CH}, by one. The repetition stops when $N_{\mathrm{CH}} = C$ since there have been C cluster-heads elected. Algorithm ACE-C therefore will stop and the result will be used in the next step of the clustering phase.

Please note that, the value of v_{id} is kept and used continuously when algorithm ACE-C is invoked. Fig. 4 shows the high-level description of algorithm ACE-C and we present an example in Fig. 5. There are nine sensor nodes numbered from 0 to 8 as the identifiers. Suppose the number of cluster-heads in each cluster is $C = 3$. By following algorithm ACE-C, sensor nodes 6, 7, and 8 are the clusters-heads in the first round after three loops. The next three loops will decide the next three cluster-heads in the next round. The process continues for deciding the cluster-heads.

Algorithm ACE-C still works when some sensor nodes die (use up all of the energy) during the election process. Suppose sensor node u dies and it is its turn to be a cluster-head. Since sensor node u is gone, all the other sensor nodes cannot receive the advertisement message from u. After waiting a period of time (i.e. Step (2.3) in Fig. 4), all the other sensor nodes will ignore u and continue the process until there are C cluster-heads elected. In Fig. 5, sensor node 6 dies in the third round. According to the algorithm, sensor node 6 should be a cluster-head in the fourth round. However, since it has died, no message will be sent at loop 12. All the other sensor nodes wait a period of time and receive nothing. They will assume that sensor node 6 has gone and then resume the loop again. At this time, sensor node 5 will be a cluster-head at loop 13. Therefore, the cluster-heads in the fourth round are nodes 5, 7, and 8 instead.



node (id) loop	0	1	2	3	4	5	6	7	8	
1	1	2	3	4	5	6	7	8	0	round 1
2	2	3	4	5	6	7	8	0	1	
3	3	4	5	6	7	8	0	1	2	
4	4	5	6	7	8	0	1	2	3	round 2
5	5	6	7	8	0	1	2	3	4	
6	6	7	8	0	1	2	3	4	5	
7	7	8	0	1	2	3	4	5	6	round 3
8	8	0	1	2	3	4	5	6	7	
9	0	1	2	3	4	5	6	7	8	
10	1	2	3	4	5	6	×	8	0	round 4
11	2	3	4	5	6	7	×	0	1	
12	3	4	5	6	7	8	×	1	2	
13	4	5	6	7	8	0	×	4	3	

Fig. 5. An example of nine sensor nodes for Algorithm ACE-C; shaded shapes indicating the elected cluster-heads.

4.2. Cluster-head election with location

In this subsection, we present a distributed **a**lgorithm of **c**luster-head **e**lection with **l**ocation, **Algorithm ACE-L**, which is especially for mobile sensor nodes. The basic idea is to use the node mobility to have each sensor node be a cluster-head in turns. Given some fixed *reference points* in the area of the mobile sensor network, the sensor nodes closest to these reference points will be the cluster-heads, respectively, when electing the cluster-heads. To achieve this, we consider to set the distance of a sensor node to a reference point as the metric of the *delay time*, which is used when a sensor node contends a channel. The decision is hence also a product of the channel contention among sensor nodes.

Suppose that there are C clusters in each round. Algorithm ACE-L defines C fixed *reference points* in the beginning. These C reference points will affect the priority for a node to contend the radio channel and the locations of the elected cluster-heads. Consider an arbitrary sensor node v. We denote the reference point closest to v as the *main reference point* (MRP) of v. The delay time of v, which is the priority to contend a channel, is then set up according to the distance from v's current location to its MRP. All the sensor nodes having the same MRP will contend a channel and the one with the shortest delay time will success. It turns out that the sensor node closest to an MRP will obtain the channel. We hence can elect such a sensor node as a cluster-head, which is closest to an MRP. When a sensor node has been elected as a cluster-head, it will use the channel obtained to broadcast a beacon of being a cluster-head to the other nodes. On the other hands, if sensor node v receives a beacon of being a cluster-head from other sensor node u during the delay time or the channel contention and node u has the

```
Algorithm of Cluster-head Election by Location
Input: C reference points
/* chb: cluster-headbeacon */
(1) decide the MRP and set distance to be the distance to the MRP
(2) if receive chb from other node u and have the same main MRP as node u then
    (2.1) cluster-head ← true
    (2.2) exit
(3) else
    (3.1) delay time ← distance
    (3.2) while (delay time decreases one) do
            (3.2.1) if receive chb from other node u and
                    have the same MRP as node u then
                    (3.2.1.1) cluster-head ← true
                    (3.2.1.2) exit
                    endif
          endwhile
    (3.3) transmit its chb
endif
End
```

Fig. 6. Algorithm ACE-L used in each sensor node for cluster-head election.

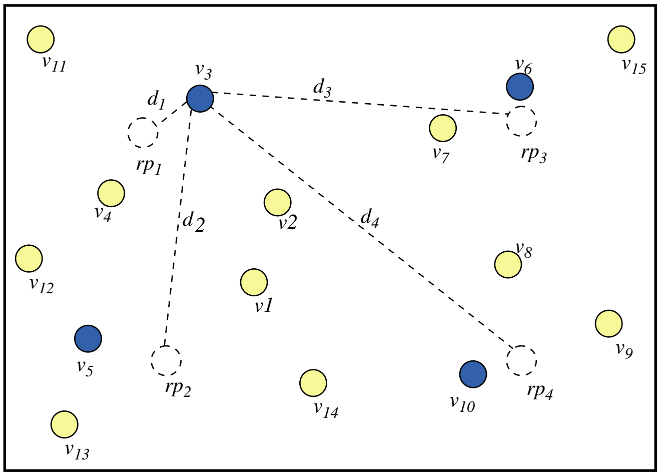


Fig. 7. An instance of execution algorithm ACE-L in a sensor network having 15 mobile sensor nodes, where the dashed circles are the reference points.

same MRP as v, sensor node v stops the competition and will be not be a cluster-head in this round.

Fig. 6 presents algorithm ACE-L step by step. In Step (1), a sensor node v calculates the distances to all the reference points and uses the reference point having the shortest distance as the MRP. Step (2) then examines whether there is an elected cluster-head which uses the same MRP as v. If there is one, the algorithm stops. Otherwise, the algorithm moves to Step (3). In Step (3), a sensor node v sets up the delay time according to the distance to its MRP. After setting the delay time, node v again examines whether there is an elected cluster-head which uses the same MRP as in Step (2). If the delay time passes, sensor node v will successfully obtain the channel. This indicates that node v is the sensor node closest to its MRP and will be a cluster-head. It then transmits the beacon of being a cluster-head to the other sensor nodes.

An example of electing $C = 4$ cluster-heads among 15 mobile sensor nodes, $v_1, v_2, \ldots,$ and v_{15}, using algorithm ACE-L is shown in Fig. 7. Let the four reference points are rp_1, rp_2, rp_3, and rp_4, respectively. Consider sensor node v_3. The distances from v_3 to all the reference points rp_1, rp_2, rp_3, and rp_4 are d_1, d_2, d_3, and d_4, respectively, and d_1 is the shortest distance. Reference point rp_1 is the MRP of v_3. Recall that we set the delay time according to the distance from a sensor node to its MRP. Hence v_3 has the smallest delay time among all the sensor nodes which have rp_1 as the MRP. After the channel contention, sensor node v_3 successes and is the cluster-head by referring to rp_1. Similarly, sensor nodes v_5, v_6, and v_{10} are the cluster-heads by referring to rp_2, rp_3, and rp_4, respectively.

The correctness comes from the property of transmission among sensor nodes in a wireless environment. Under a fixed frequency, only one sensor node can use that frequency to transmit data at an arbitrary time instance. Therefore, at each time slot, there is only one sensor node can transmit the cluster-head beacon. Suppose there are C reference points, $rp_1, \ldots, rp_C$. By the property mentioned above, if a sensor node uses reference point rp_i as the MRP for some $i \leqslant C$ and obtains the channel, all the other sensor nodes use reference point rp_i as the MRP will not send any beacon and the cluster-head is thus elected. For the other reference points, the cluster-heads can be elected similarly. Therefore, the algorithm will generate C cluster-heads exactly unless there are fewer than C sensor nodes left.

4.3. Comparison between cluster-head election algorithms

We now compare algorithm ACE-C and algorithm ACE-L in three aspects. First of all, we consider the mobility of sensor nodes. Since algorithm ACE-C does not consider the locations of the sensor nodes, it fits for both mobile and static sensor networks. In contrast with algorithm ACE-C, algorithm ACE-L is designed especially for mobile sensor networks. When applying algorithm ACE-L to a static environment, some sensor nodes will use up all the energy quickly because the static reference points will make a sensor node being a cluster-head in a consecutive rounds until it dies. Hence, algorithm ACE-L does not fit for a static sensor network. To adopt algorithm ACE-L to a static environment, one can use dynamic reference points instead. However, every sensor node needs to know the moving pattern of the reference points.

Secondly, the mobility models on the sensor network will affect the performance of algorithm ACE-L since the algorithm refers to the fixed reference points. In general, the reference points are given uniformly in the area of the sensor network. In this paper, we consider that the sensor nodes are randomly scattered and each sensor node moves randomly under the three mobility models considered. The impact of the mobility models on the cluster-head election using algorithm ACE-L becomes less. For a mobility model where the sensor nodes are distributed non-uniformly, using algorithm ACE-L to elect cluster-heads may not be a good choice. However, since algorithm ACE-C does not consider the location of a sensor node, the mobility models will not affect algorithm ACE-C at all.

Last, we discuss these two proposed algorithms from the viewpoint of a good clustering algorithms mentioned in Section 2.

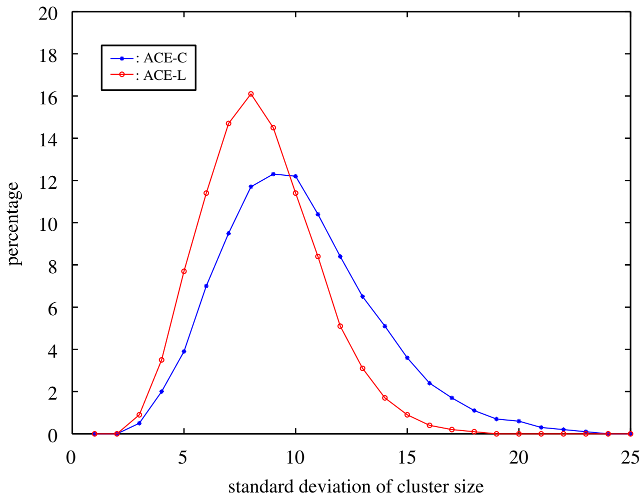


Fig. 8. The distribution of the standard deviation of the cluster size generated by algorithms ACE-C and ACE-L for 200,000 rounds.

Recall the three objectives for a good clustering algorithm. It is not difficult to see that both algorithms can generate at least one cluster-head in each round and the number of cluster-heads elected in each round is uniform for these two algorithms. As for the cluster size, things are different for these two algorithms.

Since algorithm ACE-C only uses the ID to decide the cluster-heads without considering the locations of sensor nodes, the elected cluster-heads may be close to each other. As a result, the cluster sizes (i.e. the number of sensor nodes in a cluster) may be different among all the generated clusters. It turns out that some sensor nodes (being cluster-heads with a large number of sensor nodes in the corresponding clusters) may consume a large amount of energy and the system lifetime hence becomes short. Algorithm ACE-L uses the locations to elect the cluster-heads; hence, can avoid such a situation. Consider that there are 100 sensor nodes in a sensor network and the expected number of clusters in each round for the best energy consumption is 5 [11,14]. The expected cluster size thus is 20. We apply algorithms ACE-C and ACE-L to generate the cluster-heads, respectively, and both algorithms use the clustering mechanism, CM, discussed in next section to form the clusters. Fig. 8 shows the distribution of the standard deviation of the cluster size generated by algorithms ACE-C and ACE-L for 200,000 rounds. The results indicate that the cluster size produced by algorithm ACE-L is more uniform than the one generated by algorithm ACE-C. Indeed, in our experiments, algorithm ACE-L generates more percentage of clusters of size between 18 and 22 than algorithm ACE-C does. Therefore, algorithm ACE-L leads to a better performance in terms of energy consumption and the experimental results shown in Section 6 demonstrate this trend.

5. Clustering with mobility (CM)

After the cluster-heads have been elected, the next step in the clustering phase in a round is the cluster formation step. We observe that the node mobility plays an important role when forming the clusters. If a sensor node neglects its mobility when deciding which cluster-head to join, it may select a cluster-head with a longer distance to it; therefore, consumes more power. Recall that each sensor node is equipped with a Location Finding System. Having the location information from the attached location finding device, each mobile sensor node can calculate its speed and direction. The proposed mechanism, CM (**C**lustering with **M**obility), uses the location information on each sensor node to decide which cluster this sensor node belongs to properly.

When the cluster-heads in the current round have been decided, the elected cluster-heads then broadcast an advertisement message to all the sensor nodes to recruit the cluster members. This advertisement message includes the cluster-head's position, direction, and speed. Every non-cluster-head node must listen to such a broadcast after the cluster-head election has been done. After receiving the advertisement message, each sensor node needs to select a proper cluster-head to join.

To determine which cluster-head to join, each non-cluster-head node considers the relative position to each cluster-head and will select the nearest cluster-head to join. However, since all the sensor nodes are mobile, the currently nearest cluster-head may be the farthest one for the sensor node when the message transmission phase starts in the current round. Hence, the proposed mechanism, CM, uses the location information to predict the relative position when the message transmission phase starts. Nevertheless, it is not easy to predict when the message transmission phase will start in a round. We hence consider the time instance in the time period from the time, t_1, when the cluster-head election ends, to the time, t_2, when the next round starts. Suppose sensor nodes $h_1, h_2, \ldots, h_n$ have been elected as the cluster-heads at time t_1. For a non-cluster-head sensor node v, since the advertisement messages include the location information of the cluster-head nodes', sensor node v can calculate all the distances to all the cluster-heads at time t_1 and t_2, respectively. Let d_i^1 and d_i^2 be the distances from sensor node v to the cluster-head node h_i at time t_1 and t_2, respectively, for $i = 1, 2, \ldots, n$. At a given time instance $t \in [t_1, t_2]$, the distance from sensor node v to each cluster-head h_i can be estimated with

$$DH_i(t) = \frac{d_i^2 - d_i^1}{t_2 - t_1}(t - t_1) + d_i^1, \quad i = 1, 2, \ldots, n. \tag{3}$$

Fig. 9 gives a picture of the method we have described above. Then, suppose the time instance t is given, sensor node v will select a cluster-head h_j to join during the cluster formation step, where

$$DH_j(t) = \min_{i=1}^{n}\{DH_i(t)\}.$$

Please note that Eq. (3) is an estimator for the distance from a sensor node v to a cluster-head at time t. It is not the exact distance. One, of course, can use another way to estimate.

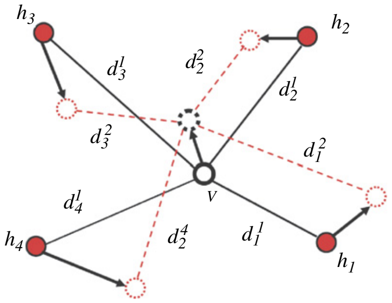


Fig. 9. A portion of a wireless mobile sensor network where, in some round, sensor node v is a non-cluster-head and h_1, h_2, h_3, and h_4 are cluster-heads; arrows represent sensor nodes' directions and speeds; solid (dashed) lines represent the distances from v to all the cluster-heads at time t_1 (t_2).

From the estimation equation, we observe that the value of t impacts the cluster formation. We further define

$$T_c(t) = \frac{t - t_1}{t_2 - t_1} \tag{4}$$

as the *clustering factor*. Then, when $t = t_1$, $T_c = 0$ and the way to form the clusters is the same as the way in LEACH. Note that, when the time instance t is given to determine the cluster-head to join for the sensor nodes, the value of T_c is also set and vice versa. Recall that, in each round, the message transmission phase is the last phase and will consume most of the power in a round. Our experimental results indicate that the value of T_c between 0.75 and 0.9 can lead to a better performance in terms of energy consumption.

After T_c has been set, a sensor node v can use Eq. (3) to decide which cluster-head to join. If there is a tie, some strategies can be used to break the tie:

- randomly select one among the cluster-heads having the same distance to the sensor node;
- using a different clustering factor to decide again; and
- consider the ratio of $\frac{d_i^2 - d_i^1}{t_2 - t_1}$ and select the one with smallest ratio since it presents that the change on the distance is small.

6. Experiments

In this section, we present the experimental results for the proposed distributed algorithms, including ACE-C and ACE-L, and CM. Many data-gathering routing protocols have been proposed on static sensor networks, such as PEGASIS and HIT. These protocols use a large amount of messages passing between sensor nodes to find better routing paths (in the form of chains) to gather data; therefore, result in a better performance than LEACH. However, when the network topology changes, the routing paths also change. In a mobile sensor network where the topology changes frequently, these protocols may not fit since a huge amount of messages will be passed between sensor nodes. In contrast with LEACH, LEACH uses a fewer amount of messages to construct the cluster-based architecture and thus can be fully applied to a mobile sensor network. We therefore consider to compare our algorithms with LEACH.

Recall that a cluster-based data-gathering protocol consists of a series of rounds and there are two major phases: clustering and message transmission phases in each round. The proposed algorithms ACE-C and ACE-L are used for cluster-head election and the mechanism CM is used for cluster formation in clustering phase. Using different algorithms in a step makes different protocols and different results. Hence, in our experiments, we consider the cluster-based protocols which employ the proposed distributed algorithms. We first implement the protocol **CM** using the mechanism, discussed in Section 5, which forms the clusters by considering the node mobility. The step for electing cluster-heads in protocol CM is the same as the one used in LEACH. Then, based on CM, we further implement two other protocols which differ in the way to elect cluster-heads as discussed in Section 4. Protocol **ACE-C** uses the cluster-head election with counting and protocol **ACE-L** uses the cluster-head election with location. The flow chart in Fig. 10 gives a picture about the process of a cluster-based protocol at each sensor node in our simulation. From now on, we will use the protocols instead of the algorithms when we discuss the performance.

The experimental results show that all these three protocols, CM, ACE-C, and ACE-L, make the whole system live longer in comparison with LEACH. For the experimental results, we first discuss the system lifetime and make a comparison between these protocols. Then, we demonstrate that the number of cluster-heads generated by the cluster-head election algorithms is uniform. Last, we show the impact of the clustering factor T_c used in CM mechanism.

We consider three mobility models in the experiments, including SM model, RWM model, and RDM model. The simulation environment is set up as follows. The sensor network area considered in the simulation is 200 m × 200 m. The BS is located at position (100, 300). There are 100 mobile sensor nodes in the sensing area. The speed of each sensor node is from 0 to 1 m/s and the initial energy in a sensor node is 1.0 J. We have each sensor node send a 2000-bits data packet to the BS in each round. The period of one round is 5 s. Initially, all the sensor nodes are assumed to be scattered randomly in the area. For the cluster-head election, the initial probability P of a sensor node to be a cluster-head is 0.05 when we consider LEACH protocol. The expected number of cluster-heads in each round is therefore $100 \times 0.05 = 5$.

We list all the parameters used in the simulation in Table 1 for reference.

6.1. System lifetime

In the experiments, we use the system lifetime as the measure of energy consumption. The system lifetime is the total number of rounds which a wireless mobile sensor network experiences. We first consider the round at which the first dead sensor node occurs. Recall that, if the number of cluster-heads in each round is equal, the energy consumption will be similar among all the sensor nodes and hence the system lifetime can last longer. In particular, an equal number of cluster-heads in each round makes the first sensor node died in a latter round. Table 2 shows

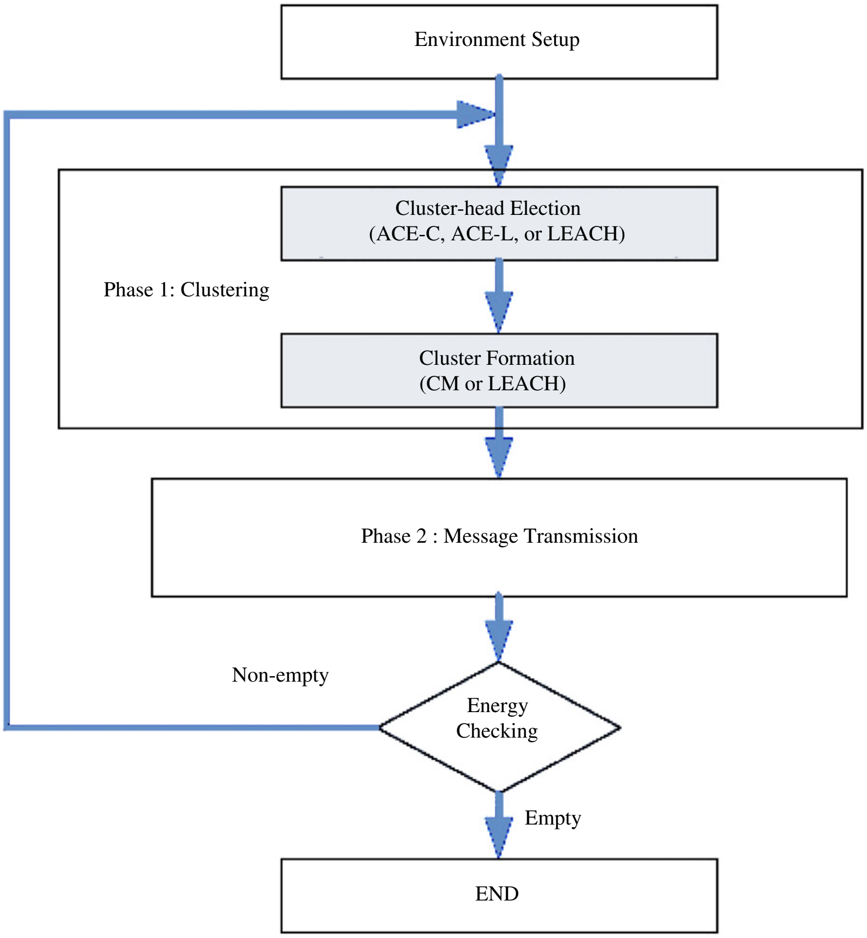


Fig. 10. The process of a cluster-based protocol at each sensor node in our simulation; using different approaches at each step making different protocols and results.

Table 1
Parameters used in the simulation

Items	Values
Network area	200 m × 200 m
Base-station location	(100,300)
Mobility models	SM, RWM, RDM
Number of sensor nodes	100
Speed of a sensor node	between 0 and 1 m/s
Size of the packet (in a round)	2000 bits
Period of each round	5 s
Transmission distance	> 300 m
Transmission rate	> 1 M/s
Reference points	(50,50), (150,150), (150,50), (50,150)
ID's of sensor nodes	0–99
Expected number of cluster-heads	5 (or 4 for MRP)

the average system lifetimes with different mobility models in terms of the number of rounds.

In this experiment, we run each protocol 40 times and measure the round at which the first sensor node dies and the round at which the last sensor node dies (i.e. the total number of rounds the sensor network experiences). All the three proposed protocols perform better than LEACH does and can make the system live longer. Protocol CM can improve LEACH protocol under the mobile environments. Based on the CM mechanism, protocols using our cluster-head election algorithms, ACE-C and ACE-L, can further make the system live much longer due to an equal number of cluster-heads elected in each round. In general, the proposed protocols can also make the first dead sensor node occur in a later round. However, when using algorithm ACE-L on the SM model, because there are only four directions to select initially for a sensor node and a sensor node bounces back when it meets the boundary, a sensor node may be a cluster-head more often than the others. Therefore, a mobile sensor network with SM model using protocol ACE-L may have the first death node occur earlier as shown in Table 2.

We further plot the system lifetimes when each sensor node has initial energy 1.0 J with RDM model and RWM model in Figs. 11 and 12, respectively. Again, as shown in the plots, our proposed protocols make the system live longer than LEACH does, thereby result in a better performance than LEACH in terms of energy consumption. Protocols ACE-C and ACE-L use algorithms ACE-C and ACE-L to elect the cluster-heads in each round, respectively. As mentioned in Section 4.3, these two cluster-head election algorithms can generate at least one

Table 2
The performance about the system lifetimes of all the protocols with different mobility models

	First node dead round				Last node dead round			
	LEACH	CM	ACE-C	ACE-L	LEACH	CM	ACE-C	ACE-L
SM	304	309	413	253	590	602	832	984
RWM	384	426	628	646	530	555	780	907
RDM	458	501	712	663	530	560	767	882

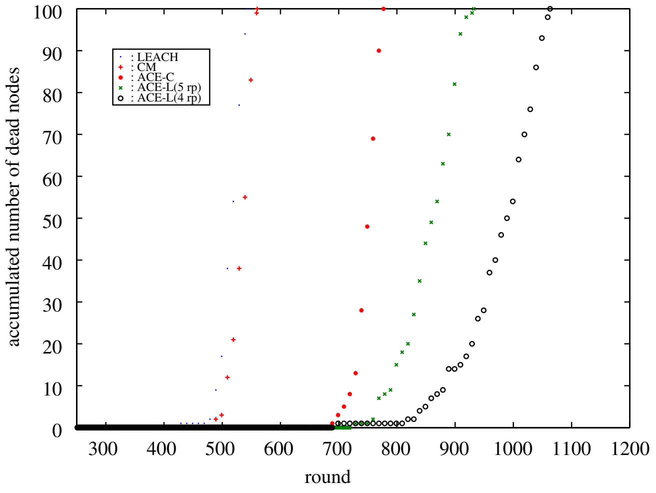


Fig. 11. The system lifetime of the wireless mobile sensor network which has 100 sensor nodes in the range of 200 m × 200 m and uses the Random Direction Mobility model on the sensor nodes.

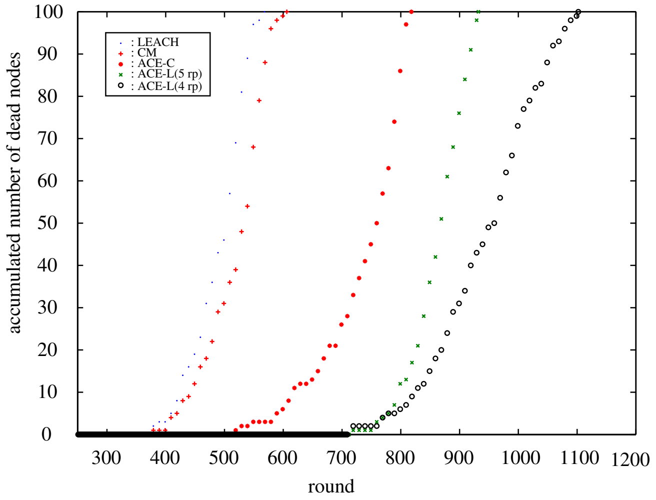


Fig. 12. The system lifetime of the wireless mobile sensor network which has 100 sensor nodes in the range of 200 m × 200 m and uses the Random Walk Mobility model on the sensor nodes.

cluster-head in each round and the number of cluster-heads elected in each round is uniform. So, the case that all the sensor nodes send their sensed information directly to the BS can be avoided. Hence, protocols ACE-C and ACE-L can improve further in the system lifetime.

In general, protocol ACE-L is better than protocol ACE-C since the positions of the cluster-heads in each round are more uniformly distributed in the interested area. In some rounds, there could be five cluster-heads close to each other when applying protocol ACE-C. In such a case, some cluster-heads will consume the energy dramatically due to a large cluster size as mentioned in Section 4 and the cluster members also need to consume more energy due to a long distance to the cluster-heads. In our experiments, we further set the number of the reference points to be four and the reference points are located at the centers of the four quadrants of the sensing region. For five reference points, we add one reference at the center of the sensing region. In our experiments, the result shows that the algorithm using four reference points (ACE-L(4pt)) outperforms the one using five reference points (ACE-L(5pt)) in terms of the energy consumption. Such a result relates to the deployment of the reference points and we will not address the problem about the deployment of the reference points in this paper. However, we provide an analysis on these two cases in Appendix A. Such an analysis explains why the energy consumption of using four reference points is less than the energy consumption of using five reference points. Besides, since the CM mechanism predicts the sensor node location more precisely in the RDM model, the performance of using the RDM model is better than the one of using the RWM model.

6.2. *Number of generated cluster-heads*

As mentioned in Section 2, by using the cluster-head election strategy of LEACH protocol, the difference in the number of cluster-heads between the rounds may be big and it happens that there is no cluster-head at all in some rounds. Our experiment results show that the range of the number of the cluster-heads in a round can be from 0 to 10 when applying the rule in LEACH and the percentage of the round in which there is no cluster-head elected is about 12–15%. Such a cluster-head election will deteriorate the performance in terms of the energy consumption. Table 3 shows the distribution of the number of the cluster-heads generated in each round for protocols ACE-C, ACE-L, and LEACH. The results also indicate that if one expects C cluster-heads in each round, our distributed algorithms will elect C cluster-heads exactly unless there are fewer than C sensor nodes left in the system. Since algorithm ACE-C uses counting, it will generate more rounds having C cluster-heads than algorithm ACE-L does.

Table 3
The percentage of different number of the cluster-heads elected in a round during a system lifetime using different cluster-head election algorithms

# of cluster-heads	0	1	2	3	4	5	6	7	8	Over 9
LEACH (%)	12	3	6	15	14	15	12	10	9	4
ACE-C (%)	0	0	0.2	0	0.1	99.7	0	0	0	0
ACE-L (%)	0	0.6	0.4	0.4	0.7	97.9	0	0	0	0

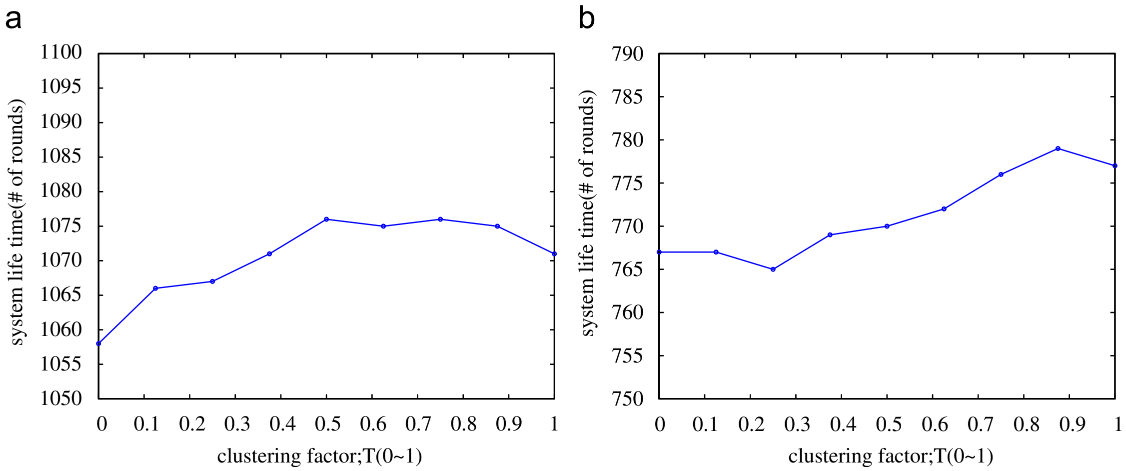


Fig. 13. The effects of different T_c values on the system lifetime in a wireless mobile sensor network with Random Walk Mobility model. (a) ACE-L protocol, (b) ACE-C protocol.

6.3. *Impact of clustering factor and the sensor node speed*

To study the impact of clustering factor T_c, we consider the time period of a round is about 5 s. The value of T_c ranges from 0 to 1. For each fixed T_c, we run 100 different cases and take the average system lifetime (total number of rounds) among these 100 cases. For a wireless mobile sensor network where each sensor node has an initial energy 1.0 J in an area of 200 m × 200 m with the RWM model, Figs. 13(a) and (b) show the effects of different T_c values on the system lifetime for protocol ACE-L and ACE-C, respectively. In general, as the value of T_c increases, the system lifetime increases. Recall that the message transmission phase is the second phase in a round and consumes most of the energy. As T_c is larger than 0.5, T_c is closer to the time when the message transmission phase starts and protocols using CM can have better prediction on the relative location to find the nearest cluster-heads. Hence, setting $T_c > 0.5$ leads to less energy consumption for transmission in the system.

We now further discuss the CM mechanism. Since mechanism CM works well when the location of a sensor node can be predicted well, we consider the SM model where one can fully predict sensor node's location and speed. Fig. 14 shows the effects of different T_c values on the system lifetime in a wireless mobile sensor network where each sensor node has an initial energy 1.0 J in a sensing area of 200 m × 200 m with the SM model. As shown in Fig. 14, as the clustering factor T_c increases, the system lifetime increases and the best value of T_c is between 0.75 and 0.9.

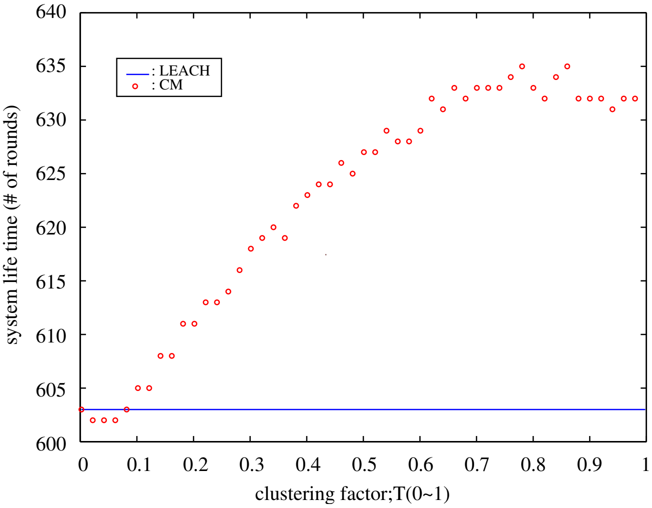


Fig. 14. The effect of different T_c values on the system lifetime in a wireless mobile sensor network with simple mobility model.

We observe that the sensor node speed impacts the performance of CM mechanism as well; therefore, impacts the system lifetime. Fig. 15 shows the effect of different maximum speeds ranged from 0 to 2.0 m/s on the system lifetime in a wireless mobile sensor network where each sensor node has an initial energy 1.0 J in an area of 200 m × 200 m on the RWM model. The cluster factor is set to $T_c = 0.875$ by following the conclusion in Fig. 14. When the maximum speed in a mobile sensor

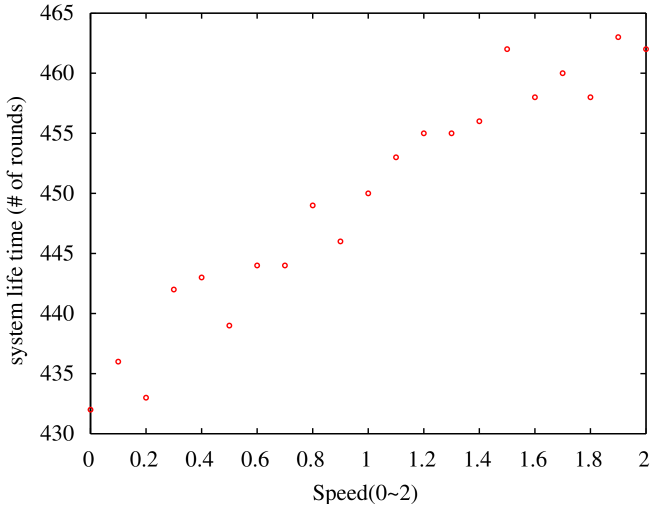


Fig. 15. The impact of different maximum speeds on the system lifetime in a wireless mobile sensor network with Random Walk Mobility model.

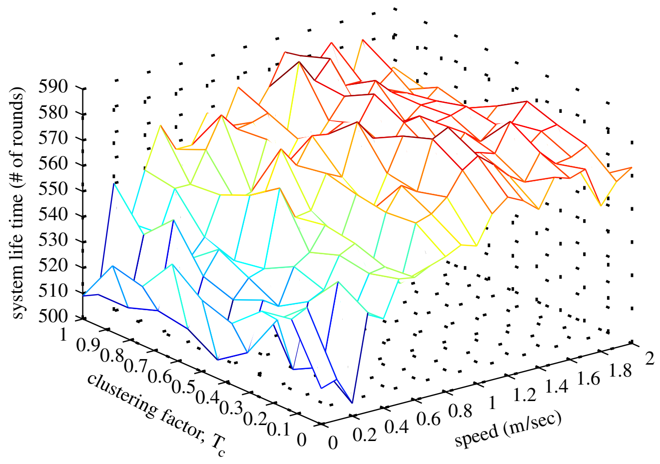


Fig. 16. The total impact of both clustering factor and maximum speeds on the system lifetime in a wireless mobile sensor network which has 100 sensor nodes using the Random Walk Mobility model.

network increases, the change on the distance between a sensor node and the cluster-head becomes dramatic. Proper prediction on the location when forming the clusters will improve the performance. On the other hand, if the average speed is slow, using prediction may not help too much since the communication range between sensor nodes may not change too much. Fig. 16 shows the trend of the impact of both the clustering factor and the sensor node speed on the performance of CM mechanism. We can conclude that, in general, as the clustering factor and the maximum speed increase, the system lifetime increases for CM mechanism.

7. Conclusions

In this paper, we provide different distributed clustering algorithms which lead less energy dissipation for data-gathering in a cluster-based mobile sensor network. There are two steps when clustering the mobile sensor nodes: cluster-head election and cluster formation steps. For cluster formation, our CM (Cluster with Mobility) mechanism can achieve a better performance in terms of the energy consumption and system lifetime when the sensor nodes are capable of mobility. Two distributed cluster-head election algorithms, ACE-C and ACE-L, are proposed for cluster-head election in a wireless mobile sensor network. Based on the CM mechanism, using the two proposed cluster-head election algorithms makes different clustering algorithms. Our clustering algorithms achieve the three objectives: there is at least one cluster-head elected in each round, the number of cluster-heads generated in each round is always the same (except the final rounds), and all the generated clusters should have the same cluster sizes.

We consider the Random Walk Mobility and Random Direction Mobility models as well as the simple mobility model in our experimental analysis. The experimental results show that our algorithms yield a better performance in terms of energy consumption; therefore, lead a longer system lifetime. When applying CM mechanism, we conclude that as the maximum speed of a sensor node or the clustering factor increases, the system lifetime increases. The experimental results also indicate a better clustering factor which is from 0.75 to 0.9. Our future work tailors toward providing distributed clustering algorithms which fit different mobility models.

Appendix A. Discussion on the number of reference points

First, we define the following notations:

- k: the number of the reference points,
- N: the total number of sensor nodes,
- l: the packet size,
- d_{toBS}: the distance between cluster-head and BS, and
- d_{toCH}: the distance between a sensor node and cluster-head.

Suppose all the sensor nodes are uniformly distributed in the network area. Using the radio model and data-fusion model in [14], the energy consumption of a cluster-head in each round is

$$\begin{aligned} E_{\mathrm{CH}} &= E_{TX} + E_{RX} + E_{\text{data-fusion}} \\ &= l \times E_{\mathrm{elec}} + l \times \varepsilon_{\mathrm{amp}} \times (d_{\mathrm{toBS}})^2 + \left(\frac{N}{k} - 1\right) \\ &\quad \times l \times E_{\mathrm{elec}} + \left(\frac{N}{k} - 1\right) \times E_{\mathrm{D}} \\ &= l \times \Bigg[E_{\mathrm{elec}} + \varepsilon_{\mathrm{amp}} \times (d_{\mathrm{toBS}})^2 \\ &\quad + \left(\frac{N}{k} - 1\right)(E_{\mathrm{elec}} + E_{\mathrm{D}}) \Bigg]. \end{aligned} \tag{5}$$

On the other hand, the energy consumption of a non-cluster-head in one round can be derived as

$$\begin{aligned} E_{\text{non-CH}} &= E_{TX} \\ &= l \times E_{\mathrm{elec}} + l \times \varepsilon_{\mathrm{amp}} \times (d_{\mathrm{toCH}})^2 \\ &= l \times (E_{\mathrm{elec}} + \varepsilon_{\mathrm{amp}} \times (d_{\mathrm{toCH}})^2). \end{aligned} \tag{6}$$

By combining Eqs. (5) and (6), the total energy consumption of all the sensor nodes in a cluster is

$$\begin{aligned}E_{\text{cluster}} &= E_{\text{CH}} + \left(\frac{N}{k} - 1\right) E_{\text{non-CH}} \\ &= l \times \Bigg\{ E_{\text{elec}} + \varepsilon_{\text{amp}} \times (d_{\text{toBS}})^2 \\ &\quad + \left(\frac{N}{k} - 1\right) \left(2E_{\text{elec}} + E_{\text{D}} + \varepsilon_{\text{amp}} \times (d_{\text{toCH}})^2\right) \Bigg\}. \end{aligned} \tag{7}$$

Using Eq. (7), we can obtain the total energy consumption of all the sensor nodes in a round. We first consider the case where four reference points are used. Recall that the four reference points are located at (50,50), (50,150), (150,50), and (150,150), respectively, and the BS is located at (100,300) in the experiments. According to [11], let $(d_{\text{toCH}})^2 = \frac{100^2}{2\pi}$. Then, by Eq. (7), the total energy consumption of all the sensor nodes in a cluster for each round is

$$E_{\text{cluster}_4} = 0.0127 + 0.0000002 \times (d_{\text{toBS}})^2. \tag{8}$$

By applying the distances between the cluster-heads and the BS to Eq. (8), we can conclude that, when there are four reference points located as above, the total energy consumption of all the mobile sensor nodes is a round is 0.0868 J. Similarly, when there are five reference pointed located at (50,50), (50,150), (150,50), (150,150), and (100,100), respectively, we can derive the total energy consumption of all the mobile sensor nodes in a round as 0.0883 J. Therefore, using four reference points outperforms using five reference points in terms of energy consumption.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Comput. Networks 38 (2002) 393–422.

[2] N. Bulusu, J. Heidemann, D. Estrin, Gps-less low cost outdoor localization for very small devices, IEEE Personal Comm. Magazine 7 (5) (2000) 28–34.

[3] T. Camp, J. Boleng, V. Davies, A survey of mobility models for ad hoc network research, Wireless Comm. Mobile Comput. 2 (5) (2002) 483–502.

[4] A. Cerpa, J. Elson, D. Estrin, L. Girad, M. Hamilton, J. Zhao, Habitat monitoring: application driver for wireless communication technology, in: Proceedings of ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean, 2001, pp. 3–5.

[5] Z. Cheng, M. Perillo, B. Tavli, W. Heinzelman, S. Tilak, N. Abu-Ghazaleh, Protocols for local data delivery in wireless microsensor networks, in: Proceedings of the 45th IEEE Midwest Symposium on Circuits and Systems (MWSCAS '02), 2002, pp. 623–626.

[6] B.J. Culpepper, L. Dung, M. Moh, Design and analysis of hybrid indirect transmissions (hit) for data gathering in wireless micro sensor networks, ACM SIGMOBILE Mobile Comput. Comm. Rev. 8 (1) (2004) 61–83.

[7] D. Davies, Evaluating mobility models within an ad hoc network, Master's Thesis, Colorado School of Mines, 2000.

[8] E.J. Duarte-Melo, M. Liu, Data-gathering wireless sensor networks: organization and capacity, Comput. Networks 43 (2003) 519–537.

[9] H. Hassanein, J. Luo, Reliable energy aware routing in wireless sensor networks, in: Proceedings of the 2nd IEEE Workshop on Dependability and Security in Sensor Networks and Systems, 2006, pp. 54–64.

[10] S. Hedetniemi, A. Liestman, A survey of gossiping and broadcasting in communication networks, Networks 18 (4) (1988) 319–349.

[11] W. Heinzelman, Application-specific protocol architectures for wireless networks, Ph.D. Thesis, Massachusetts Institute of Technology, 2000.

[12] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, Adaptive protocols for information dissemination in wireless sensor networks, in: Proceedings of ACM International Conference on Mobile Computing and Networking (MobiCom '99), 1999, pp. 174–185.

[13] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy-efficient communication protocol for wireless microsensor networks, in: Proceedings of the 33rd Hawaii International Conference on System Sciences, 2000, pp. 1–10.

[14] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, An application-specific protocol architecture for wireless microsensor networks, IEEE Trans. Wireless Comm. 1 (4) (2002) 660–670.

[15] C. Intanagonwiwat, R. Govindan, D. Estrin, Directed diffusion: a scalable and robust communication paradigm for sensor networks, in: Proceedings of ACM International Conference on Mobile Computing and Networking (MobiCom '00), 2000, pp. 56–67.

[16] K. Kalpakis, K. Dasgupta, P. Namjoshi, Maximum lifetime data gathering and aggregation in wireless sensor networks, in: Proceedings of the 2002 IEEE International Conference on Networking (ICN'02), 2002, pp. 685–696.

[17] M. Lee, V.W.S. Wong, Lpt for data aggregation in wireless sensor networks, in: Proceedings of IEEE Global Telecommunications Conference, 2005.

[18] P. Levis, S. Madden, D. Gay, J. Polastre, R. Szewczyk, A. Woo, E. Brewer, D. Culler, The emergence of networking abstractions and techniques in tinyos, in: Proceedings of the First Symposium on Networked System Design and Implementation (NSDI04), 2004, pp. 1–14.

[19] L. Li, J.Y. Halpern, Minimum-energy mobile wireless networks revisited, in: Proceedings of the IEEE International Conference on Communications (ICC '01), 2001.

[20] S. Lindesy, C. Raghavendra, K.M. Sivalingam, Data gathering algorithms in sensor networks using energy metrics, IEEE Trans. Parallel Distributed Systems 13 (9) (2002) 924–935.

[21] C.-M. Liu, C.-H. Lee, Power efficient communication protocols for data gathering on mobile sensor networks, in: Proceedings of the 60th IEEE Vehicular Technology Conference, Fall, 2004.

[22] C.-M. Liu, C.-H. Lee, L.-C. Wang, Power-efficient communication algorithms for wireless mobile sensor networks, in: Proceedings of the 1st ACM International Workshop on Performance Evaluation of Wireless Ad hoc, Sensor, and Ubiquitous Networks, 2004, pp. 121–122.

[23] J. Park, S. Sahni, An online heuristic for maximum lifetime routing in wireless sensor networks, Proc. IEEE Trans. Comput. 55 (8) (2006) 1048–1056.

[24] D. Petrovic, R.C. Shah, K. Ramchandran, J. Rabaey, Data funneling: routing with aggregation and compression for wireless sensor networks, in: Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Application, 2003, pp. 156–162.

[25] T.S. Rappaport, Wireless Communications, Prentice-Hall, Englewood Cliffs, NJ, 1996.

[26] E. Royer, P.M. Melliar-Smith, L. Moser, An analysis of the optimum node density for ad hoc mobile networks, in: Proceedings of the IEEE International Conference on Communications (ICC '01), 2001.

[27] A. Savvides, C. Han, M.B. Srivastava, Dynamic fine-grained localization in ad hoc network of sensors, in: Proceedings of ACM International Conference on Mobile Computing and Networking (MOBICOM '01), 2001.

[28] R.C. Shah, J.M. Rabaey, Energy aware routing for low energy ad hoc sensor networks, in: Proceedings of IEEE Wireless Communications and Networking Conference (WCNC), 2002, pp. 17–21.

[29] K. Sohrabi, J. Gao, V. Aliawadhi, G.J. Pottie, Protocols for self-organization of a wireless sensor network, IEEE Personal Comm. (2000) 16–27.

[30] D. Tian, N.D. Georganas, A node scheduling scheme for energy conservation in large wireless sensor networks, Wireless Comm. Mobile Comput. 3 (2003) 271–290.

[31] Y.-C. Tseng, S.-Y. Ni, E.-Y. Shih, Adaptive approaches to relieving broadcast storms in a wireless multi-hop mobile ad hoc network, IEEE Trans. Comput. 52 (5) (2003) 545–557.

[32] L.-C. Wang, C.-W. Wang, C.-M. Liu, An adaptive contention window-based cluster head election mechanism for wireless sensor networks, in: Proceedings of the IEEE Vehicular Technology Conference 2005-Fall (VTC 2005-Fall), 2005.

[33] M. Younis, M. Youssef, K. Arisha, Energy-aware management for cluster-based sensor networks, Comput. Networks 43 (5) (2003) 649–668.

[34] O. Younis, S. Fahmy, Heed: a hybrid, energy-efficient, distributed clustering approach for ad-hoc sensor networks, IEEE Trans. Mobile Comput. 3 (4) (2004) 366–379.

[35] L. Zhao, X Hong, Q Liang, Energy-efficient self-organization for wireless sensor networks: a fully distributed approach, in: Proceedings of IEEE Global Telecommunications Conference, 2004, pp. 2728–2732.

Chuan-Ming Liu is an assistant professor in the Department of Computer Science and Information Engineering, National Taipei University of Technology (NTUT), Taiwan. He received his Ph.D. in Computer Sciences from Purdue University in 2002 and B.S. and M.S. degrees both in Applied Mathematics from National Chung-Hsing University, Taiwan, in 1992 and 1994, respectively. He is a member of Upsilon Pi Epsilon Honor Society in Computer Science since 1998. His research interests include parallel and distributed computation, data management and data dissemination in wireless environments, ad hoc and sensor networks, and analysis and design of algorithms.

Chuan-Hsiu Lee received his B.S. and M.S. degrees in Electrical Engineering and Computer Science from National Taipei University of Technology (NTUT), Taiwan in 2003 and 2005, respectively. His research interests are in the areas of mobile sensor networks, ad hoc networks, and mobile computing. Currently, he is an engineer in ALi Co., Taiwan.

Li-Chun Wang received his B.S. degree from National Chiao Tung University, Taiwan, ROC in 1986, the M.S. degree from National Taiwan University in 1988, and the Ms. Sci. and Ph.D. degrees from the Georgia Institute of Technology, Atlanta, in 1995, and 1996, respectively, all in electrical engineering. From 1990 to 1992, he was with the Telecommunications Laboratories of the Ministry of Transportations and Communications in Taiwan (currently the Telecom Labs of Chunghwa Telecom Co.). In 1995, he was affiliated with Bell Northern Research of Northern Telecom, Inc., Richardson, TX. From 1996 to 2000, he was a Senior Technical Staff Member in the Wireless Communications Research Department of AT&T Labs-Research. Since August 2000, he has been an Associate Professor in the Department of Communication Engineering of National Chiao Tung University in Taiwan and became a full processor in August 2005. His current research interests are in the areas of adaptive/cognitive wireless networks, radio network resource management, cross-layer optimization, and cooperative wireless communications networks. Dr. Wang was a co-recipient (with Gordon L. Stuer and Chin-Tau Lea) of the Jack Neubauer Best Paper Award of the IEEE Vehicular Technology Society in 1997. He has published over 30 journals and 70 international conference papers and is holding three US patents. He served as an Associate Editor for the IEEE Transactions on Wireless Communications from 2001 to 2005, the guest editor of the special issue on “Mobile Computing and Networking” for IEEE Journal on Selected Areas in Communications in 2005 and the special issue on ”Radio Resource Management and Protocol Engineering in Future IEEE Broadband Networks” for IEEE Wireless Communications Magazine in 2006.