

Mobility-assisted Distributed Sensor Clustering for Energy-efficient Wireless Sensor Networks

Kai Li, Kien A. Hua

*Department of Electrical Engineering and Computer Science, University of Central Florida
Orlando, Florida 32816*

kaili@eeecs.ucf.edu kienhua@eeecs.ucf.edu

Abstract—Clustering proves to be an effective approach to reduce energy consumption of micro-sensors and prolong the lifetime of wireless sensor networks. In this paper, we address the sensor clustering problem in the design of an autonomous MANET (*a*MANET) for energy-efficient data collection in wireless sensor networks. In the *a*MANET context, we formulate the network energy optimization as a weighted K-means clustering problem, and present a distributed technique called MADSEC (*Mobility-Assisted Distributed S*ensor *C*lustering) for sensor clustering and data collection. MADSEC allows the mobile nodes to dynamically run a distributed clustering algorithm to reposition themselves and organize energy-efficient sensor clusters. Extensive NS-2 simulation results indicate that the proposed distributed clustering strategy is as effective as a theoretical centralized K-means clustering technique. Furthermore, MADSEC can at least double the network lifetime of LEACH, a well-known static clustering method.

I. INTRODUCTION

Advances in sensing and networking technology have created enormous opportunity for developing diverse applications [1] of societal importance. In a typical Wireless Sensor Network (WSN), a large number of tiny sensors are deployed to monitor the physical environment, process sensing information, and report to the sink through wireless communications. Sensor nodes are typically resource-constrained micro-electronic devices. Once deployed, they are left unattended to operate on energy-constrained batteries. This necessitates effective solutions in various aspects of WSNs, such as routing, medium access control, duty cycle scheduling, etc.

In this paper, we consider a novel approach to WSN design, in which a separate autonomous mobile ad hoc network (*a*MANET) is used for data collection. An *a*MANET is different from a traditional MANET in that an *a*MANET node can move autonomously and cooperatively to accomplish some common task. In our environment, this task is to collect sensor data and forward them to the sink. This *a*MANET approach is motivated by wireless connected robots [14].

It has been recognized that adding mobility can improve the lifetime of WSN [3]. Existing solutions have mostly considered mobile nodes in the form of mobile sinks which normally act as gateways to a backbone network. The *a*MANET approach is different in that the mobile nodes form an autonomous MANET, in which each mobile node is responsible for collecting data from sensors in its neighbourhood. These data are eventually relayed to the sink over the *a*MANET. Since these mobile nodes are significantly less expensive than

the advanced mobile sinks, the *a*MANET approach is more cost effective for many applications.

One of the key challenges of designing the *a*MANET is that we need a clustering technique to cluster the sensors into groups. It's desirable that clustering can be performed distributedly to ensure scalability. Although a number of clustering protocols have been proposed to save sensor's energy and prolong the network lifetime [4], [5], [10], [11], most of them are based on the assumption that the entire network is composed of only static nodes. In this paper, we are targeting an environment where in addition to a large number of low cost static nodes (i.e., sensors), we can use a few resourceful mobile nodes (i.e., *a*MANET nodes). We are particularly interested in how to utilize those *a*MANET nodes for distributed sensor clustering and how much network lifetime gain can we obtain through the use of them. We formulate the energy-oriented sensor clustering in an *a*MANET context as a weighted K-means problem. Based on this novel formulation, we further present the *Mobility-Assisted Distributed S*ensor *C*lustering (MADSEC) protocol. MADSEC allows every mobile *a*MANET node to automatically and dynamically organize energy-efficient sensor clusters, with performance close to that of the K-means algorithm.

The remainder of the paper is organized as follows: Section II briefly review the related work. In Section III, we present some numerical results and then formally formulate the problem. In Section IV, we describe the MADSEC protocol. Section V shows the effectiveness of MADSEC protocol by simulating it in NS-2. Finally, we conclude the paper in Section VI.

II. RELATED WORK

Low-Energy Adaptive Clustering Hierarchy (LEACH) [4] is one of the first widely used hierarchical clustering algorithm. In LEACH, sensors make independent decisions about whether to become CHs according to a probability function. This probability function try to make sure: (1) a certain ratio of CHs will be maintained in each round (2) a sensor that become CH in one round will not be CH in a number of following rounds. Local data fusion is performed at each CH to reduce redundant packets. LEACH is also extended to a centralized LEACH-C [6] and a multiple-tier TL-LEACH [9], which utilizes two levels of CHs (primary and secondary).

Hybrid Energy-Efficient Distributed Clustering (HEED) [5]

is another well-known multi-hop clustering protocol. HEED selects CHs using the node residual energy as the primary parameter and intra-cluster communication cost as the secondary. HEED produces better-distributed CHs compared with LEACH and thus slightly improved network lifetime performance. In [10], the author proposed Stochastic and Equitable Distributed Energy-Efficient Clustering (SEDEEC) for energy efficient CH distribution in WSNs where the base station far from the network. SEDEEC uses an equitable and stochastic technique which uniformly distribute the energy consumption throughout the network. Energy Efficient heterogeneous clustering scheme (EEHC) [11] is designed for sensor networks where a portion of the sensor nodes are assumed to be equipped with additional energy resources. It takes advantage of node heterogeneity by assigning different weights to the probability for different type of nodes to become CH in each round.

Many other clustering protocols exists in the literature and the reader can refer to [2] for a comprehensive survey. These clustering techniques differ by the criteria they use for the choice of CHs. They save sensor energy by local data fusion, which reduce the number of data packets, and they distribute energy dissipation among sensors by proper CH role rotation strategies.

III. PROBLEM MODELLING

A. Energy Analysis

We start by deriving simple energy consumption models of two generalized clustering paradigms. In the first paradigm, CHs are chosen from normal static sensors and the Static Cluster Head (SCHs) roles are periodically rotated among sensors to distribute energy consumption. In the second paradigm, there are a few resourceful mobile aMANET nodes acting as Mobile Cluster Heads (MCHs), with the ability to reposition themselves from time to time to organize dynamic sensor clusters. We assume that communication energy consumption is not a concern of MCHs (e.g. they could be rechargeable).

Using the same radio model as in [4] and [5], the energy dissipation of transmitting and receiving one-bit data over a distance d are given as $E_{tx}(d) = E_{elec} + E_{amp} \cdot d^\lambda$ and $E_{rx} = E_{elec}$ respectively, where E_{elec} is the energy dissipation of the transceiver circuit, and E_{amp} is the that of the amplifier. Both E_{amp} and λ are dependent on the sender and receiver distance d : $E_{amp} = \epsilon_{fs}$, $\lambda = 2$ when $d \leq d_0$ and $E_{amp} = \epsilon_{amp}$, $\lambda = 4$ when $d > d_0$, where d_0 is a constant distance that depends on the environment.

Now assume each sensor transmits m -bit of data to its CH in one round. The average energy consumption for a non-CH sensor in each round would be

$$E_{SN} = m \cdot (E_{elec} + E_{amp} \cdot d_{SN}^\lambda)$$

where d_{SN} is the average distance from a sensor to its CH. For a CH, its energy consumption will be

$$E_{CH} = mN_{CH}E_{elec} + m \cdot (E_{elec} + E_{amp} \cdot d_{CH}^\lambda)$$

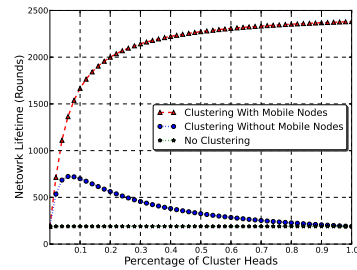


Fig. 1: Sensors' average energy consumption vs. different CH ratio

where d_{CH} is the average distance from a CH to the sink, and N_{CH} is the average number of sensors in each cluster. Note that CH sends out only m -bit data per round because CH performs data fusion (e.g. by averaging) to compress every N_{CH} data packets into a single one. Let α be the ratio of number of CHs to the total number of sensors. Then sensors' average energy consumption for the static sensor clustering paradigm would be

$$E_{SCH} = (1 - \alpha) * E_{SN} + \alpha E_{CH}$$

While for clustering with MCHs, sensors' average energy consumption is simply

$$E_{MCH} = E_{SN}$$

It's not hard to examine that $E_{SCH}/E_{MCH} > 1$ always holds as long as CH consumes energy than non-CH sensor and the ratio will be greater with larger α and increasing network size. We also note that both d_{SN} and N_{CH} are dependent on α , therefore E_{SCH} and E_{MCH} are not simple linear functions of α . In order to numerically compare the two paradigms, we use the simple approximations:

$$N_{CH} \approx \alpha N \text{ and } d_{SN} \approx \sqrt{\frac{\text{NetworkArea}}{N_{CH}}}$$

We use typical radio parameter settings ($E_{elec} = 5nJ/bit$, $\epsilon_{fs} = 100pJ/bit/m^2$, $\epsilon_{amp} = 0.0013pJ/bit/m^4$ and $d_0 = 75m$) to conduct numerical experiments of a $100m \times 100m$ WSN with 100 sensors. Each sensor has an initial energy of 1Joule and transmits 8Kb data to its CH during each round. The network lifetime of the two schemes are shown in Fig. 1. Both clustering paradigms are reduced to the direct transmission (i.e. no clustering) case when the number of CHs is set to zero. The plot shows that for static clustering, there exists an optimal ratio of CHs, which is consistent with the simulation results obtained by [4]. While for clustering with MCHs, the performance always outperforms that of SCH and the gap drastically increases with higher CH ratios. We remark that the numerical results are only approximations of the two clustering paradigms, and performance of actual clustering protocols will differ with different designs. Motivated by these results, we will formally formulate the mobility-assisted clustering problem and give distributed solutions to achieve extended network lifetime in the following sections

B. Problem Formulation

We are considering a sensor network with N randomly deployed sensor nodes and K MCHs. Each sensor has an some initial energy that cannot be replenished. Similar to [5], we also require each node to have a fixed number of transmission power levels. This functionality is available in many sensor products such as TelosB[8]. And it's typically straightforward to set the transmission power via the standard *iotcl()* call.

During each round of length ΔT , a sensor directly sends m -bit data to its current MCH with the minimum transmission power. We use C_k to denote the cluster led by the k^{th} MCH located at \mathbf{u}_k . Then our objective is to minimize the following energy function in each round i :

$$\mathcal{Q}_i = \sum_n^N \lambda_n(i) \sum_k^K r_{nk} e_{nk}(\mathbf{u}_k)$$

where $e_{nk}(\mathbf{u}_k)$ is the energy sensor n consumed by transmitting data to MCH k , r_{nk} is a binary variable representing sensor n 's cluster membership and $\lambda_n(i)$ is a weighting factor associated with sensor n 's residual energy at the beginning of the i^{th} round. $\lambda_n(i)$ can be chosen such that lower energy nodes can have a greater influence.

If we use the first-order energy model in previous discussions: $e_{nk} = \beta + \alpha \|\mathbf{x}_n - \mathbf{u}_k\|_2^2$, where $\beta = mE_{elec}$ and $\alpha = mE_{amp}$, then the minimization of \mathcal{Q}_i could be done through an iterative weighted K-means procedure showed as Algorithm 1. We note that the above procedure has to be

Algorithm 1 Weighted K-means Algorithm

Initialization:

$\Omega_K = \{1, 2, \dots, K\}$, $\Omega_N = \{1, 2, \dots, N\}$
Randomly initialize \mathbf{u}_k for all $k \in \Omega_K$

Iteration:

Repeat

// Step 1: Assign sensor to the closest MCH (Cluster formation)

$r_{nk} \leftarrow 1$ for $k = \arg \min_j \|\mathbf{x}_n - \mathbf{u}_k\|_2^2$ for all $n \in \Omega_N$

// Step 2: Update MCHs positions (Cluster Update)

$\mathbf{u}_k \leftarrow \frac{\sum_n \lambda_n(i) r_{nk} \mathbf{x}_n}{\sum_n \lambda_n(i) r_{nk}}$ for all $k \in \Omega_K$

Until \mathcal{Q}_i converge

Output: \mathbf{u}_k and r_{nk}

executed for every round because the weighting factor $\lambda_n(i)$ changes with the sensors' residual energy. Obviously the above algorithm is a centralized solution that require the location information of the sensors and a centralized dispatch of MCHs. However, in the next section we will propose a protocol that allows MCHs to automatically organize near-optimal clusters in a distributed manner.

IV. THE MADSEC PROTOCOL

In this section, we will firstly explain the MCH initiated sensor clustering process. Then we present the protocol and the pseudo code. Finally, we describe the intra-cluster scheduling and data collection.

A. The Clustering Protocol

Based on previous discussions, we know that the weighted K-means procedure optimize the energy function by repeated execution of cluster formation and cluster updates. MADSEC approximate the optimization process in similar steps, however, in a distributed manner. In MADSEC, MCHs broadcast *invite_msg(MyID)* during cluster formation using the maximum transmission power. Sensors only join an MCH with the strongest Received Signal Strength (RSS).

The cluster updates in MADSEC make use of the weighted Average Minimum Reachable Power (*wAMRP*) metric to update MCH locations. *wAMRP* extends AMRP in [5] by assigning weight to each sensor, and it's defined as the weighted average minimum power for all cluster members to reach the CH. Let N be the number of sensors in a cluster. Let ϵ_i be sensor i 's residual energy, and $w_i = f(\epsilon_i)$ be the weight of this sensor. Then we have

$$wARMP = \frac{1}{N} \sum_{i=1}^N w_i \cdot MinRp_i$$

where $MinRp_i$ is the minimum transmission power for sensor i to reach the MCH, which can be estimated according to the law of radio path loss

$$MinRp_i = \begin{cases} \frac{4\pi^2 LP_r d^2}{G_t G_r \lambda^2}, & \text{if } d < d_0 \\ \frac{LP_r d^4}{G_t G_r h_t^2 h_r^2}, & \text{if } d \geq d_0 \end{cases}$$

where P_r is the receiver sensitivity, G_t and G_r are the antenna gains of the transmitter and the receiver respectively, h_t and h_r are the antenna heights, L is the system loss, and λ is the wavelength. If we use the first order path loss approximation, *wAMRP* can be written in terms of sensor and MCH locations as

$$wAMRP = \frac{C}{N} \sum_{i=1}^N w_i \|\mathbf{u} - \mathbf{x}_i\|_2^2$$

where $C = \frac{(4\pi)^2 LP_r}{\lambda^2 G_t G_r}$. It's obvious that when $\mathbf{u}^* = \frac{1}{N} \sum w_i \mathbf{x}_i$, we get the minimum *wAMRP*. Note the \mathbf{u}^* is exactly the updated MCH positions in Algorithm 1 if we normalize $\sum_n \lambda_n(i) r_{nk}$ to $\sum_n r_{nk}$. Intuitively, MCHs can use *wAMRP*

Algorithm 2 MADSEC protocol for computing *wAMRP*

```

1: ClusterInfo = {}
2: for power_level = 1 to MAX_POWER_LEVEL do
3:   Set transmission power to Power(power_level)
4:   Broadcast probe_msg(MyID)
5:   for all received ack_msg(SensorID, ResEnergy, MyID) do
6:     if SensorID  $\notin$  ClusterInfo then
7:       Add SensorID to ClusterInfo,
       Compute weight according to ResEnergy, and
       Record weight and Power(power_level)
8:     end if
9:   end for
10: end for
11: Compute wAMRP

```

as an indicator to update their locations. However we are

still too optimistic to directly apply the above observation for two reasons. Firstly, MCH could not accurately measure the $wARMP$ value at any location in its cluster, given only a finite discrete set of transmission power levels. Secondly, it's not practical for the MCH to exhaustively survey its cluster region and measure the $wARMP$ at each location to find the minimum $wARMP$. MADSEC resolve the first problem using Algorithm 2 to compute an approximated $wAMRP$. As for the second problem, we will mathematically show that, the MCH can arrive at the optimal location with only three $wAMRP$ samples at three different locations, even without any location information.

Let $wARMP = g(\mathbf{u})$ be the $wARMP$ a MCH measured at location \mathbf{u} . Assume the MCH is initially located at \mathbf{u}_0 , which is unknown. It makes two random moves to \mathbf{u}_1 and \mathbf{u}_2 . We have

$$\mathbf{u}_{i+1} = \mathbf{u}_i + vt_i \begin{bmatrix} \cos \theta_i \\ \sin \theta_i \end{bmatrix}, i = 0, 1$$

where θ_i and t_i is the random direction and random time of the i^{th} move, and v is the moving speed. The three $wAMRP$ measurements are

$$g(\mathbf{u}_i) = \frac{C}{N} \sum_{n=1}^N \|\mathbf{u}_i - \mathbf{x}_n\|_2^2, i = 0, 1, 2$$

By subtracting $g(\mathbf{u}_{i+1})$ from $g(\mathbf{u}_i)$, we get the linear equation group

$$\begin{cases} g(\mathbf{u}_0) - g(\mathbf{u}_1) = 2C\mathbf{l}_1^T [\frac{1}{N} \sum w_i \mathbf{x}_n - \frac{1}{2}(\mathbf{u}_0 + \mathbf{u}_1)] \\ g(\mathbf{u}_1) - g(\mathbf{u}_2) = 2C\mathbf{l}_2^T [\frac{1}{N} \sum w_i \mathbf{x}_n - \frac{1}{2}(\mathbf{u}_1 + \mathbf{u}_2)] \end{cases}$$

where $\mathbf{l}_1 = vt_0 \begin{bmatrix} \cos \theta_0 \\ \sin \theta_0 \end{bmatrix}$, $\mathbf{l}_2 = vt_1 \begin{bmatrix} \cos \theta_1 \\ \sin \theta_1 \end{bmatrix}$ and we use $\sum_n w_n = N$. Note that both \mathbf{x}_n and \mathbf{u}_i are unknowns since we assume location unawareness. Our goal is to arrive at $\mathbf{u}^* = \frac{1}{N} \sum w_i \mathbf{x}_n$ for cluster update.

We observe that the above computation will be carried out at \mathbf{u}_2 . Therefore the MCH only needs to calculate $\mathbf{y} = \mathbf{u}^* - \mathbf{u}_2$ to get to the desired location \mathbf{u}^* . Reorganizing the above linear equation in terms of \mathbf{y} will give us

$$\begin{bmatrix} \cos \theta_0 & \sin \theta_0 \\ \cos \theta_1 & \sin \theta_1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

where

$$b_1 = \frac{f(\mathbf{u}_0) - f(\mathbf{u}_1)}{2C} + \frac{1}{2}v^2(t_0^2 + t_1^2)$$

and

$$b_2 = \frac{f(\mathbf{u}_1) - f(\mathbf{u}_2)}{2C} + \frac{1}{2}v^2(t_0^2 + t_1^2)[1 + 2 \cos(\theta_0 - \theta_1)]$$

The only requirement for us to get a valid solution is that the 2×2 coefficient matrix be full rank, which indicates

$$\theta_1 - \theta_0 \neq k\pi, k \in \mathbb{Z}$$

The constraint dictates that the two random moves \mathbf{l}_1 and \mathbf{l}_2 should not be collinear, which can be easily satisfied by the MCH changing its direction when making the second random move. The solution to the final move \mathbf{y} means that MCH does

not need any location information for cluster updates. The only information used here is the *three wARMP samples* at three different locations and the *two random moves* that lead to these locations.

The complete clustering protocol of MADSEC is shown in Algorithm 3. The clustering process is the repeated execution of cluster formation and cluster updates, as in the centralized weighted K-means algorithm. However, in MADSEC each MCH executes the same clustering protocol independently, requiring no pre-knowledge about the network. Therefore, the termination condition could not be determined as in the centralized weighted K-means, in which the global error function is evaluated. Here we use a local decision method: each MCH makes independent decisions about whether to terminate local clustering by calculating the distance between the final locations of two consecutive iterations. The evaluation of the distance error d_E (see Algorithm 3) again does not require any location information. When local clustering is terminated by an MCH, it will wait until all the other MCHs have terminated their clustering to initiate the data collection process. The clustering synchronization is achieved through communications over the MANET formed by MCHs.

B. Data Collection

An MCH will schedule inter-cluster data aggregation after clustering is finished. Each round of data collection has a length of ΔT and is divided into M TDMA frames. Similar to [4], each MCH creates an intra-cluster TDMA schedule allocating one of the M TDMA frames to each sensor. During idle TDMA frames, an MCH fuses the data received from its member nodes into a single frame and sends it over the MCH overlay network towards the sink. Intra-cluster interference and overhearing is handled by putting sensors to sleep during other sensors data transmission frames. To reduce energy consumed by inter-cluster overhearing, an MCH further split a sensor's data transmission frame into I sub-frames and randomly allocate one of the I sub-frames for the sensor to transmit data, while putting the sensor to sleep during the rest.

Algorithm 3 The MADSEC clustering protocol

```

1: Initialize:  $d_E \leftarrow \text{Infinity}$ ,  $\text{loopCnt} \leftarrow 0$ 
2: while  $d_E > d_{th}$  and  $\text{loopCnt} < \text{MAX\_LOOP}$  do
3:   // Cluster formation
4:   MyCluster = {}
5:   Broadcast invite_msg(MyID)
6:   for all received ack_msg(SensorID, MyID) do
7:     Add SensorID to MyCluster
8:   end for
9:   // Cluster Updates
10:  Measure  $wAMRP$  (Algorithm 2)
11:  for  $i = 0, 1$  do
12:    Make a random move  $\mathbf{l}_i$ 
13:    Measure  $wAMRP$ 
14:  end for
15:  Compute  $\mathbf{y}$  and make the final move
16:   $\text{loopCnt} \leftarrow \text{loopCnt} + 1$ 
17:   $d_E \leftarrow \|\mathbf{l}_1 + \mathbf{l}_2 + \mathbf{y}\|_2^2$ 
18: end while

```

V. PERFORMANCE EVALUATION

In this section, we present the simulation results of the proposed protocol. All of the simulation results are average over least 20 runs unless otherwise specified. The general simulation parameters are presented in Table I.

We first compare the performance of the following cluster-

TABLE I: Simulation Parameters

Type	Parameter	Value
Network	Network size	$100m \times 100m$
	Number of sensors (N)	100
	Sensor distribution	Random
	Sink Location	(50m, 0)
Application	Sensor initial energy(E_0)	1 Joules
	ME speed (v)	2m/s
	Data packet size (k)	1000 bytes
	Data rate (r)	100 kb/s
Radio transceiver	Radio frequency	2.4GHz
	Maximum transmission power	-1.58dBm
	Receiver Sensitivity	-24dBm
	E_{elec}	5 nJ/bit
	ϵ_{fs}	10 pJ/bit/m ²
	ϵ_{mp}	0.0013 pJ/bit/m ²
	Threshold distance (d_0)	75m
MADSEC	Round length (ΔT)	50 s
	Clustering Frequency (T_{NO})	5 rounds
	Number of TDMA frames (M)	100
	Number of Sub-frame (I)	2
	Number of power levels (L)	20

ing schemes:

- Centralized LEACH (C-LEACH): The basic operations of our implementation of C-LEACH are similar to that of the original LEACH, except that we assume there exists a centralized server holding information of the whole network, and the CH election and rotation are totally controlled by the server. Therefore, our C-LEACH implementation should be an upper bound of the original LEACH protocol.
- Random Mobility (RM) clustering: For RM clustering, each MCH makes a random move to a new location when clustering is triggered. Sensor join an MCH with the minimum RSS.
- MADSEC: We considered MADSEC with equal weights and unequal weights. For unequal weights, MADSEC uses the exponential function $w_i = \exp(\bar{\epsilon}/\epsilon_i)$, where $\bar{\epsilon}$ is the average residual energy of sensors in a cluster.

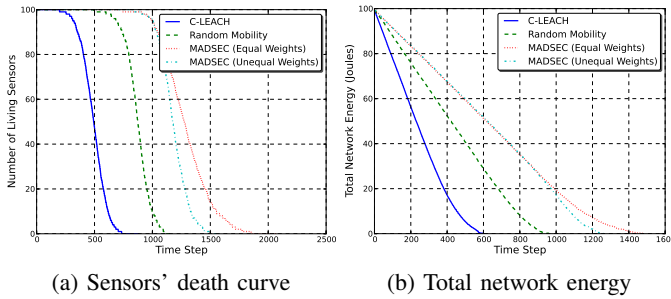


Fig. 2: Network lifetime comparison of different clustering schemes

CH ratio is set to 5% for all schemes, which should be the range where static clustering has the best performance

TABLE II: Network lifetime of different clustering schemes

Initial Energy (Joules)	Protocols	Round first node dies	Round last node dies
0.5	C-LEACH	101	442
	RMC	275	583
	MADSEC(equal weights)	422	952
	MADSEC(unequal weights)	467	782
1.0	C-LEACH	242	783
	RMC	584	1125
	MADSEC(equal weights)	906	1836
	MADSEC(unequal weights)	983	1536

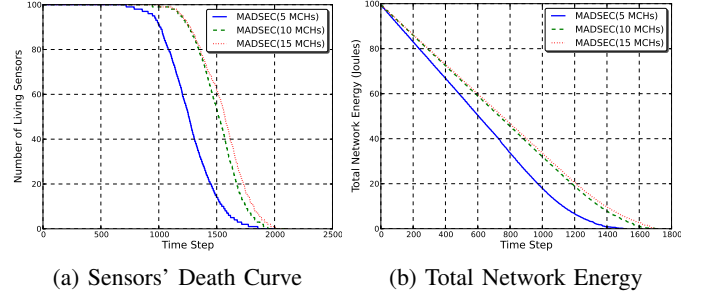
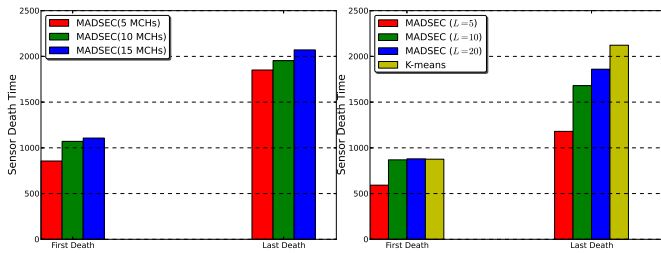


Fig. 3: MADSEC Behaviors with varying number of MCHs

based on our numerical experiments. In ideal conditions, clustering should be triggered in every round (i.e. $T_{NO} = 1$) to maximally increase the network lifetime. However, such small T_{NO} will incur too much overhead and system instability in practice. Therefore we set $T_{NO} = 5$ in the simulations. The results are shown in Fig. 2. We note that even RM clustering can significantly increase the network lifetime compared with C-LEACH. The two variants of MADSEC further improve on RM clustering with a large margin. Another view of the results is shown in Table II. The weighted MADSEC scheme has the best performance in terms of first node death time, because it gives more weight to sensors of less energy thus more adaptive in distributing sensor energy consumption compared with equal weights MADSEC. However, the energy function equal weights MADSEC aims to minimize is exactly the average energy consumption of the entire network, which is why it achieves the longest last sensor death time and also performs the best in the total network energy consumption benchmark.

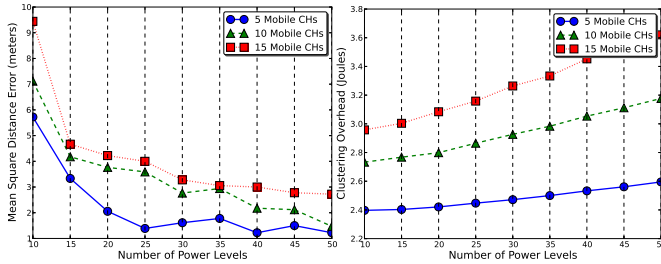
In the second experiment, we are interested in how the performance of MADSEC scales with varying number of MCHs. The results are shown in Fig. 3 and Fig. 4a. As we can see, MADSEC does benefit from more CHs, which is consistent with our numerical analysis. However, we remark that more MCHs incur more cost in real applications, therefore a trade off has to be made during deployment.

We also studied the behaviours of MADSEC under different number of power levels. We compare the actual cluster formed using the equal weights MADSEC with that of the centralized K-means algorithm. In this experiment, MADSEC is initialized by randomly placing five MEs in the sensor network. The five coordinates are also used to initialize the K-means algorithm. We then measure the mean square distance error between final positions of MCH and that computed by K-means. The results are shown in Fig. 5. As we see, the more



(a) Network lifetime under vary-number of MCHs (b) Network lifetime under varying Number of power levels

Fig. 4: Network lifetime of different MADSEC parameter settings



(a) Mean square distance error (b) Clustering overhead

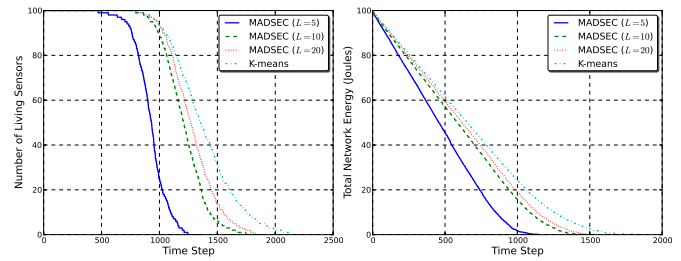
Fig. 5: MADSEC behaviors with varying number of power levels

power levels we have, the more accurately MCHs calculate target location in the cluster updates step. However, we also note that, more power levels incur more clustering overhead (see Fig. 5b). This is because in the cluster formation step, the number of *probe_msg* messages broadcast by MCHs is proportional to the number of power levels. The overhead is also proportional to the number of MCHs. Generally, more MCHs will result in smaller cluster sizes, therefore larger location errors because *wAMRP* is an averaging metric.

Finally we compare the network lifetime of MADSEC with different power levels to K-means. In this experiment, K-means directly calculates the optimal cluster configuration using sensors' location information, while MADSEC executes the distributed clustering. The results are shown in Fig. 6 and Fig. 4b. We note the performance of MADSEC is very close to that of the K-means algorithm in general, which is a result of its near-optimal cluster formation. And the performance gap between MADSEC and K-means decreases with more power levels.

VI. CONCLUSION

In this paper, we addressed the sensor clustering problem in the design of an autonomous Mobile Ad Hoc Network (*aMANET*). We formulate the mobility-assisted sensor clustering problem that result in a weighted K-means solution. Based on the formulation, we propose the Mobility-assisted Distributed Sensor Clustering for energy-efficient data collection in WSNs. MADSEC allows mobile *aMANET* nodes to intelligently organize energy-efficient sensor clusters in a distributed manner, with performance comparable to that of a centralized algorithm. We finally showed the effectiveness of our approach by NS2 simulation.



(a) Sensors' death curve (b) Total network energy

Fig. 6: Comparing K-means and MADSEC(different number of power levels)

ACKNOWLEDGEMENT

This material is based upon work partially supported by the National Science Foundation under Grant Number 0917082. Any opinions, findings, and conclusion or recommendations expressed in this materials are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

- [1] Puccinelli, D., Haenggi, M., "Wireless sensor networks: applications and challenges of ubiquitous sensing," *Circuits and Systems Magazine*, IEEE , vol.5, no.3, pp. 19- 31, 2005
- [2] Boyinbode, O., Hanh Le, Mbogho, A., Takizawa, M., Poliah, R., "A Survey on Clustering Algorithms for Wireless Sensor Networks," *Network-Based Information Systems (NBIS)*, 2010 13th International Conference on , vol., no., pp.358,364, 14-16 Sept. 2010
- [3] Yi Shi, Hou, Y.T., "Theoretical Results on Base Station Movement Problem for Sensor Network," *INFOCOM 2008. The 27th Conference on Computer Communications*, IEEE , vol., no., pp.1,5, 13-18 April 2008
- [4] Heinzelman, Wendi Rabiner, Anantha Chandrakasan, and Hari Balakrishnan. "Energy-efficient communication protocol for wireless microsensor networks." *System Sciences*, 2000. Proceedings of the 33rd Annual Hawaii International Conference on. IEEE, 2000.
- [5] O. Younis and S. Fahmy, "Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Trans. Mobile Comput.*, vol.3, no. 4, Oct.-Dec. 2004.
- [6] Muruganathan, S.D., Ma, D.C.F., Bhasin, R.I., Fapojuwo, A., "A centralized energy-efficient routing protocol for wireless sensor networks," *Communications Magazine*, IEEE , vol.43, no.3, pp.S8,13, March 2005
- [7] R. Rajagopalan and P. K. Varshney, "Data aggregation techniques in sensor networks: A survey," *IEEE Communications Surveys and Tutorials*, vol. 8, no. 4, pp. 48-63, 4th Quarter 2006.
- [8] Polastre, J., Szewczyk, R., Culler, D., "Telos: enabling ultra-low power wireless research," *Information Processing in Sensor Networks*, 2005. IPSN 2005. Fourth International Symposium on , vol., no., pp. 364- 369, 15 April 2005
- [9] Loscri V. , Marano S. , Morabito G. , " A Two-Levels Hierarchy for Low-Energy Adaptative Clustering Hierarchy (TL-LEACH).". Proceedings "VTC2005", Dallas (USA), pp. 1809-1813, Sept, 2005
- [10] B. Elbhiri, R. Saadane, D. Aboutajdine, "Stochastic and Equitable Distributed Energy-Efficient Clustering (SEDEEC) for heterogeneous wireless sensor networks", *International Journal of Ad Hoc and Ubiquitous Computing*, Vol. 7, no. 1, pp. 4-11, Jan. 2011.
- [11] D. Kumar, T. C. Aseri, and R. Patel, "EEHC: Energy efficient heterogeneous clustered scheme for wireless sensor networks," *Computer Communications*, vol. 32, no. 4, pp. 662-667, 2009.
- [12] Wei Wang, Srinivasan, V., Kee-Chaing Chua, "Extending the Lifetime of Wireless Sensor Networks Through Mobile Relays," *Networking, IEEE/ACM Transactions on* , vol.16, no.5, pp.1108-1120, Oct. 2008
- [13] Hung Quoc Ngo, Young-Koo Lee, Sungyoung Lee, "MEPA: A New Protocol for Energy-Efficient, Distributed Clustering in Wireless Sensor Networks," *Wireless Communication Systems*, 2007. ISWCS 2007. 4th International Symposium on , vol., no., pp.40-44, 17-19 Oct. 2007
- [14] http://www.mobilerobots.com/Mobile_Robots.aspx