Adaptive Decentralized Re-Clustering for Wireless Sensor Networks

Chih-Yu Wen and William A. Sethares
Department of Electrical and Computer Engineering
University of Wisconsin-Madison
wen@cae.wisc.edu and sethares@ece.wisc.edu

***Abstract*— This paper discusses a dynamic decentralized algorithm for re-clustering the sensors of an ad hoc sensor network. Each sensor uses a random waiting timer and local criteria to determine whether to form a new cluster or to join a current cluster. The clusterhead reselection process is triggered when the energy reserves of the clusterhead falls below a threshold. The algorithm operates without a centralized controller, it operates asynchronously, and does not require that the location of the sensors be known *a priori*. An analysis of cluster lifetime, the energy requirements of the algorithm, and a simplified model are used to study the behaviors of the proposed algorithm. The performance of the algorithm is described analytically and via simulation.**

I. Introduction

Unlike wireless cellular systems with a robust infrastructure, sensors in an ad hoc network may be required to self-organize. Such sensor networks are self-configuring distributed systems and, for reliability, should also operate without centralized control. In addition, because of the limited energy source, energy-efficiency is a critical consideration.

There has been extensive research on the design and development of energy efficient networking techniques [1]-[11]. In [1], the Low-Energy Adaptive Clustering Hierarchy (LEACH) utilizes a randomized periodical rotation of clusterheads to balance the energy load among the sensors. LEACH-C (Centralized) [2] uses a centralized controller to select clusterheads. The main drawbacks of this algorithm are nonautomatic clusterhead selection and the requirement that the position of all sensors must be known. LEACH's stochastic algorithm is extended in [3] with a deterministic clusterhead selection. Simulation results demonstrate that an increase of network lifetime can be achieved compared with the original LEACH protocol. The Ad hoc Network Design Algorithm (ANDA) [4] maximizes the network lifetime by determining the optimal cluster size and the optimal assignment of sensors to clusterheads but requires a priori knowledge of the number of clusterheads, number of sensors in the network, and the location of all sensors. The Weighted Clustering Algorithm (WCA) [5] considers the number of neighbors, transmission power, mobility, and battery usage in choosing clusters. It limits the number of sensors in a cluster so that clusterheads can handle the load without degradation in performance. These clustering methods rely on synchronous clocking for the exchange of information among sensors which typically limits these algorithms to smaller networks [12]. The authors in [13] derived upper bounds on the lifetime of sensor networks, while in [14] an analytical model to estimate and evaluate the network lifetime is presented. [15] shows that a globally optimal solution to the problem of maximizing a static network lifetime can be provided through a graph theoretic approach.

Our previous work [6] used a decentralized algorithm for organizing an ad hoc sensor network into clusters. Each sensor operates independently, monitoring communication among its neighbors. Based on the number of neightbors and a randomized timer, each sensor either joins a nearby cluster, or else forms a new cluster with itself as clusterhead. Since clusterheads have more responsibilities than other sensors, their power may drain more quickly. Accordingly, this paper introduces a method of choosing new clusterheads for an already established cluster. In the *Adaptive Clustering Algorithm via Waiting Timer* (ACAWT), the clusterhead reselection process is triggered when the energy reserves of the clusterhead falls below a threshold. This self-configuration is energy efficient, scalable, and extendeds the lifetime of the network. An analysis of cluster lifetime and a simplified model of the algorithm are derived, and the results are compared to the behavior of the algorithm in a number of settings.

II. The Adaptive Clustering Algorithm via Waiting Timer (ACAWT)

This section describes a randomized distributed algorithm that forms clusters and reselects clusterheads efficiently. The network setup is performed in three phases: “clustering,” “reselecting a clusterhead,” and “restructuring the clusters.” The main assumptions on the network are that (a) the sensors are in fixed but unknown locations, (b) all links between sensors are bidirectional, and (c) all sensors have the same transmitting range. Observe that there is no base station or centralized control to coordinate or supervise activities among sensors.

A. Phase I: Forming Clusters

When sensors of a network are first deployed, they may apply the Clustering Algorithm via Waiting Timer (CAWT) from [6] to partition the sensors into clusters using the waiter timer

$$WT_i^{(k+1)} = \beta \cdot WT_i^{(k)}, \tag{1}$$

where $WT_i^{(k)}$ is the waiting time of sensor *i* at time step k and $0 < \beta < 1$ is inversely proportional to the number of neighbors. If the random waiting timer expires and none of the neighboring sensors are in a cluster, then sensor *i* declares itself a clusterhead. It then broadcasts a message notifying its neighbors that they are assigned to join the new cluster with ID *i*.

After applying the CAWT, there are three different kinds of sensors: (1) the clusterheads (2) sensors with an assigned cluster ID (3) sensors which become 2-hop sensors. These sensors will join any nearby cluster after τ seconds where τ is a constant chosen to be larger than all of the waiting times. Thus, the topology of the ad-hoc network is now represented by a hierarchical collection of clusters.

B. Phase II: Re-select A Clusterhead

This subsection presents two methods of choosing a new clusterhead for an existing cluster, the centralized model and the distributed model. If the energy E_i of clusterhead *i* is less than a threshold level η, then sensor *i* broadcasts a message to its cluster members to start the reselection process. Only those sensors with energy larger than η are eligible.

1) The Centralized Model: The current clusterhead, sensor i, determines a new clusterhead by aggregating energy and neighbor information from its cluster members and solving the optimization problem:

$$\arg\max_{l} \quad (1 - \frac{E_l^{(k)}}{E_l^{max}})^{N_l} \tag{2}$$

$$\text{subject to :} \quad E_l > \eta;\, l \in C_i, \tag{3}$$

where $E_l^{(k)}$ is the energy at time step *k*, E_l^{max} is the initial energy of sensor *l*, C_i is the index set of the cluster members of sensor *i*, and N_l is the number of neighbors of sensor l. That is, the current clusterhead picks the new clusterhead, choosing a member with large energy and many neighbors.

2) The Distributed Model and Subcluster Formation: The distributed method operates much like the CAWT in utilizing a random timer. Once the energy in the current clusterhead is below the threshold, it transmits a message to start the reselection process. Each cluster member then checks the energy constraint. As long as the cluster member satisfies the constraint, it generates a random waiting time:

$$WT_i^{(k+1)} = (1 - \frac{E_i^{(k)}}{E_i^{max}})^{N_i^c} \cdot WT_i^{(k)}, \tag{4}$$

which depends on the number of neighboring cluster members N_i^c and the remaining energy level. The motivation for forming subclusters is to provide a way to do multi-hop communication within a cluster, which may be needed because sensors are no more than 2 hops away from the initial clusterhead and sensors may be up to 4 hops away from the new clusterhead. Hence, sensors in a cluster may be further classified as: (1) subcluster member, (2) subclusterhead, or (3) clusterhead. Subclusters and subclusterheads are generated by applying the ACAWT to the cluster topology.

C. Phase III: Restructure the Clusters

For real applications, it is possible that the clusterhead may malfunction before broadcasting the reselection message. One solution is that if a certain amount of time has passed with no messages from the clusterhead, then all sensors begin their timers and apply the algorithm. As a result, restructuring the cluster formation of the network may be required when the clusterhead malfunctions or when none of the cluster members satisfy the energy constraint. In this case, it may necessary to re-initialize the network into new clusters to help balance the energy burden.

III. Analysis of the ACAWT

Since a cluster is a small network, the behavior of the algorithms may be analyzed (following our results in [6]) by modifying the Averaged Model to investigate and describe the subclustering behavior.

A. Overview of The Averaged Model

The CAWT can be modeled by a simplified averaging procedure. Assume that a single clusterhead and an average number of neighboring sensors $E^{(k)}[N_i]$ are removed during each iteration k. Assume that each sensor will be removed with probability $p_{rm}^{(k)} = r_k/m_k$, where r_k is the number of sensors to be removed and m_k is the number of sensors remaining at iteration *k*. Denote the collection of sensors at iteration *k* by V_k. Since a clusterhead and its neighboring sensors are removed at each iteration, the collection of sensors at the next iteration, V_{k+1}, is simply a new and smaller network. Lindeberg Theorem [16] can be applied to approximate the distribution of the number of clusterheads at iteration *k* by $\mathcal{N}(\mu_k, \sigma_k^2)$, where $\mu_k = \sum_{i=1}^{m_k} p_i^{(k)}$, $\sigma_k^2 = \sum_{i=1}^{m_k} p_i^{(k)}(1-p_i^{(k)})$, m_k is the number of sensors in V_k, $p_i^{(k)}$ is the updated probability distribution of sensor *i* at iteration k, which is proportional to the number of neighboring sensors, $i \in I_k$, and I_k is the index set of sensors at iteration *k*. Once the procedure terminates, the number of iterations is an estimate of the number of clusterheads formed in the network.

B. Modified Averaged Model

The operation of the ACAWT with the distributed model is partitioned into rounds, where each round initializes, subclusters are formed, clusterheads are reselected, and finally a reorganization phase may be needed. This modified Averaged Model is given in Table I.

To obtain the mean and variance of the number of clusterheads of each iteration, the probability distribution of these random variables must be updated. However, it is not simple to calculate $p_i^{(j,k)}$ at iteration *k* at round *j* since the process of selecting a clusterhead at each iteration is complex. The following simplified analysis restructures the connectivity of the network so that each sensor has the same average neighboring density at each iteration. Therefore, we have

$$E^{(j,k+1)}[N_i] = \frac{N_b^{(j,k)} - r_k^{(j)} \cdot E^{(j,k)}[N_i]}{m_{k+1}^{(j)}}. \tag{5}$$

TABLE I

MODIFIED AVERAGED MODEL: PROCEDURE FOR ANALYZING THE ADAPTIVE CLUSTERING ALGORITHM VIA WAITING TIMER.

a) Let $m_k^{(j)}$ be the number of possible clusterhead candidates at iteration k at round *j*.
b) Let $N_b^{(j,k)}$ be the sum of neighboring sensors at iteration k at round *j*, $N_b^{(j,k)} = \sum_{i=1}^{m_k^{(j)}} N_i^{(j,k)}$, where $i \in I_{j,k}$; $I_{j,k}$ is the index set of likely candidates at iteration *k* at round *j*.
c) Let $E^{(j,k)}[N_i]$ be the average number of neighbors at iteration k at round *j*, $E^{(j,0)}[N_i] = \frac{N_b^{(j,0)}}{m_0^{(j)}}$.
d) Assign the prior probability $p_i^{(j,k)}$ to sensor *i*, proportional to the number of neighboring sensors $N_i^{(j,k)}$ and energy level $E_i^{(j)}/E_i^{max}$. That is, $p_i^{(j,k)} \propto \frac{N_i^{(j,k)}}{N_b^{(j,k)}} \cdot \frac{E_i^{(j)}}{E_i^{max}}$.
e) Let $E_{ch}^{(j)}$ be the remaining energy of a clusterhead at round *j*.
f) Let the initial number of sensors that will be removed $r_0^{(j)}$ be 0 $\forall$ *j*.

assign $j = 0$
while $E_{ch}^{(j)} < \eta$
  (1) subclustering formation
    assign $k = 0$
    while $(m_k^{(j)} - r_k^{(j)}) > 0$
      $m_{k+1}^{(j)} = m_k^{(j)} - r_k^{(j)}$,
      $E^{(j,k+1)}[N_i] = \frac{N_b^{(j,k)} - r_k^{(j)} \cdot E^{(j,k)}[N_i]}{m_{k+1}^{(j)}}$,
      $r_{k+1}^{(j)} = \lceil E^{(j,k+1)}[N_i] \rceil^* + 1$,
      $k = k + 1$.
    end
    $j = j + 1$.
  (2) reselecting a new clusterhead
    if $k > 0$
      assign the subclusterhead with the lowest sensor ID to be the new clusterhead with energy $E_{ch}^{(j)}$.
    else
      stop (restructure the cluster formation).
    end
end
$^*\lceil \cdot \rceil$ is the ceiling function.

Thus, the distribution of the number of subclusterheads can be approximated by $\mathcal{N}(\mu_{sch_j}, \sigma^2_{sch_j})$, where

$$\mu_{sch_j} = \sum_{k=1}^{N_{it}^{(j)}} \mu_{k_j} = \sum_{k=1}^{N_{it}^{(j)}} \sum_{i=1}^{m_k^{(j)}} p_i^{(j,k)}, \tag{6}$$

$$\sigma^2_{sch_j} = \sum_{k=1}^{N_{it}^{(j)}} \sigma^2_{k_j} = \sum_{k=1}^{N_{it}^{(j)}} \sum_{i=1}^{m_k^{(j)}} p_i^{(j,k)}(1 - p_i^{(j,k)}), \tag{7}$$

where $N_{it}^{(j)}$ is the number of iterations at round *j*.

Moreover, suppose that the expectation of the number of neighboring sensors of each sensor is used to approximate the number of neighboring sensors that will be removed at each iteration (i.e. the sensors which will eventually join the new subcluster). Thus,

$$E^{(j,k)}[N_i] = E^{(j)}[N_i] = \frac{1}{m_0^{(j)}} \sum_{i=1}^{m_0^{(j)}} N_i, \text{ for all } k.$$

Then

$$r_k^{(j)} = \lceil E^{(j)}[N_i] \rceil + 1,$$

and a simple formula for predicting the number of subclusterheads at round *j* is

$$N_{sch}^{(j)} = \frac{m_0^{(j)}}{\lceil E^{(j)}[N_i] \rceil + 1}. \tag{8}$$

The close relationship between the behavior of the ACAWT and that of the modified Averaged Model is shown experimentally in Section VI.

IV. Energy Consumption Analysis

This section considers energy consumption of the ACAWT using both the centralized and distributed models. The total power requirements include both the power required to transmit messages and the power required to receive (or process) messages. The 1-hop and 2-hop cluster members depend on the initial hierarchy of clusters. A *n*-hop cluster member is a sensor which is *n* hops away from its initial clusterhead. Let N_i be the number of neighboring sensors of sensor *i*, N_i^{n-hop} be the number of *n*-hop cluster members of clusterhead sensor *i*, and I_s be the index set of the subclusterheads.

A. The Centralized Model

For the present clusterhead to select a new clusterhead, it must gather information from the sensors in the cluster. Thus the clusterhead requests data by sending the interest message using 2 rounds of local flooding propagation to its 1-hop and 2-hop cluster members. The number of transmissions $N_{T_1}^c$ and receptions $N_{R_1}^c$ of this design choice are approximately given by

$$N_{T_1}^c \approx 1 + N_i^{1-hop}, \tag{9}$$

$$N_{R_1}^c \approx N_i^{1-hop} + \sum_{j \in C_i} N_j, \tag{10}$$

where C_i is the index set of the cluster members of sensor *i*.

Data from the cluster members is then sent towards the clusterhead. The number of transmissions $N_{T_2}^c$ and receptions $N_{R_2}^c$ are

$$N_{T_2}^c \approx N_i^{1-hop} + N_i^{2-hop}, \tag{11}$$

$$N_{R_2}^c \approx \sum_{j \in C_i} N_j. \tag{12}$$

When the clusterhead receives the desired information for solving the optimization problem of (2) and (3), it determines the new clusterhead and notifies all members. The number of transmissions $N_{T_3}^c$ and receptions $N_{R_3}^c$ are thus $N_{T_3}^c = N_{T_1}^c$ and $N_{R_3}^c = N_{R_1}^c$.

B. The Distributed Model

This subsection examines the energy consumption of the distributed model in three phases. Phase I of this model is to broadcast a message and group cluster members into subclusters. In this phase, the cluster is considered as a small network where the energy consumption analysis of the CAWT [6] can be applied. Therefore, if the current clusterhead is sensor i, the number of transmissions $N^d_{T_1}$ and receptions $N^d_{R_1}$ in an error-free channel are approximately given by

$$N^d_{T_1} \approx 2 \cdot (N^{1-hop}_i + N^{2-hop}_i), \tag{13}$$

$$N^d_{R_1} \approx 2 \cdot \sum_{j \in C_i} N_j. \tag{14}$$

The mission of Phase II is to collect sufficient information from subcluster members. The subclusterhead first broadcasts an interest message to inform its members about what kind of data it requires. Based on this message, the subcluster members propogate the desired data back to the subclusterhead. Thus, the number of transmissions $N^d_{T_2}$ and receptions $N^d_{R_2}$ are approximately

$$N^d_{T_2} \approx \sum_{j \in I_s} (1 + 2 \cdot N^{1-hop}_j + N^{2-hop}_j), \tag{15}$$

$$N^d_{R_2} \approx \sum_{j \in I_s} (N^{1-hop}_j + 2 \cdot \sum_{k \in C_j} N_k). \tag{16}$$

In the final phase, subclusterheads exchange ID information in order to determine the new clusterhead. The energy consumed in this phase may depend on the number of subclusterheads, the related positions among subclusterheads, and how they communicate with each other. Assume that there exists n_{sch} subclusterheads in a cluster. In this case, each subclusterhead broadcasts an interest message including its sensor ID to the whole cluster, which allows subclusterheads to figure out which subclusterhead is the new clusterhead immediately as they receive the ID information and thereby complete the reselection process. Therefore, we may approximate the number of transmissions $N^d_{T_3}$ and receptions $N^d_{R_3}$ by

$$N^d_{T_3} \approx n_{sch} \cdot (N^{1-hop}_i + N^{2-hop}_i), \tag{17}$$

$$N^d_{R_3} \approx n_{sch} \cdot \sum_{j \in C_i} N_j, \tag{18}$$

The analysis suggests that, compared with the overall energy consumption of the distributed model, the centralized model consumes less energy for reselecting a clusterhead while the reselection process may fail due to the malfunction of the current clusterhead and the corrupted information collection.

V. Sensor Lifetime and Cluster Lifetime

The main objective of the ACAWT is to extend the lifetime of the clusters so that the network may remain functional longer. Say that the cluster lifetime occurs when the first sensor in the cluster fails. Since the clusterhead requires large amounts of energy for communication, it is likely that the first sensor failure occurs at the clusterhead. Therefore, it is worthwhile to understand the lifetime of individual sensors.

Depending on the traffic model of the network, the expected sensor lifetime may be different. Suppose that the sensors measure periodically and transmit the data back to the clusterhead for further processing with a steady traffic. We also assume that the clusterhead collects the information from cluster members and communicates with the base station with a steady traffic flow. Denote P^{ch}_i as the power dissipation of sensor *i* for being a clusterhead and P_i as the power dissipation of sensor *i* for being a cluster member. From the analysis of the Modified Averaged Model in Section III, sensor *i* is chosen to be a clusterhead at round *j* with probability $p^{(j)}_i$. Note that if sensor *i* has not been chosen as a clusterhead before, then $p^{(j)}_i$ is proportional to the number of neighboring sensors and its energy level; otherwise, $p^{(j)}_i = 0$. Thus, the expected lifetime $E[T^{(j)}_i]$ of sensor *i* at round *j* is

$$E[T^{(j)}_i] = p^{(j)}_i \cdot \frac{E^{(j)}_i - E^{(j+1)}_i}{P^{ch}_i} + (1 - p^{(j)}_i) \cdot \frac{E^{(j)}_i - E^{(j+1)}_i}{P_i},$$

where $E^{(j)}_i - E^{(j+1)}_i$ is the energy consumption at round *j*. Hence, the operation time of performing the phase of reselecting a clusterhead is

$$E[T_i]_{ch} = \sum_j E[T^{(j)}_i].$$

Since this process is a part of the network operation, the expected lifetime of sensor *i* is given by

$$\begin{aligned} E[T_i] &\geq E[T_i]_{ch} \\ &= \sum_j E[T^{(j)}_i] \end{aligned}$$

Based upon the definition of the cluster lifetime, the cluster lifetime is equal to the minimum of the expected lifetime of sensors. That is,

$$L_{ch} \equiv \min_i \{E[T_i]\} \geq \min_i \{E[T_i]_{ch}\}. \tag{19}$$

However, for a cluster with a fixed clusterhead, the expected lifetime of sensor *i* with the prior probability p_i for being a clusterhead is

$$\tilde{E}[T_i] = p_i \cdot \frac{E_i}{P^{ch}_i} + (1 - p_i) \cdot \frac{E_i}{P_i}. \tag{20}$$

Similarly, the expected lifetime of a cluster with a fixed clusterhead is

$$\tilde{L}_{ch} \equiv \min_i \{\tilde{E}[T_i]\}. \tag{21}$$

To quantitatively measure how well the cluster lifetime are extended, we introduce a parameter, cluster lifetime factor (CLF). The CLF is defined as the ratio of the cluster lifetime of a changing clusterhead to the cluster lifetime of a fixed clusterhead. Thus, the CLF is

$$CLF \equiv \frac{L_{ch}}{\tilde{L}_{ch}} = \frac{\min_i\{E[T_i]\}}{\min_i\{\tilde{E}[T_i]\}}. \quad (22)$$

Now we provide an example on how the cluster lifetime can be extended by applying the ACAWT. Assume that sensors of the network have identical initial energy levels and power dissipation, which means $E_i^{max} = E_0$, $P_i^{ch} = P_{ch}$, and $P_i = P$ for all *i*. Note that $P_i^{ch} = P_{ch} = u \cdot P$, where $u > 1$. In a cluster with the strategy of fixing clusterhead, the cluster lifetime is

$$\tilde{L}_{ch} = \min_i\{\frac{E_i^{max}}{P_i^{ch}}\} = \frac{E_0}{u \cdot P}.$$

On the other hand, assuming that sensor *i* is the clusterhead at round *j*, the cluster lifetime with a changing clusterhead is

$$\begin{aligned} L_{ch} &\geq \min_i\{\frac{E_i^{(j)} - E_i^{(j+1)}}{P_i^{ch}} + \frac{E_i^{max} - (E_i^{(j)} - E_i^{(j+1)})}{P_i}\} \\ &= \min_i\{\frac{\triangle E_i}{u \cdot P} + \frac{E_0 - \triangle E_i}{P}\}, \end{aligned}$$

where $\triangle E_i = E_i^{(j)} - E_i^{(j+1)}$. Therefore, the cluster lifetime factor (CLF) is

$$CLF \geq u - (u-1) \cdot \min_i\{\frac{\triangle E_i}{E_0}\}, \quad (23)$$

which shows that $CLF > 1$ since $u > 1$ and $E_0 > \triangle E_i$. That means the cluster can last longer by using different clusterheads at different times, further extending the lifetime of the network.

VI. Experimental Results

This section explores the dynamic distribution of clusterheads by applying the centralized or the distributed model for reselecting clusterheads. In the experiments, the CAWT is carried out first to form the initial hierarchy of clusters, which provides the backbone for the clusterhead-reselection process. Afterwards, each clusterhead examines its energy level to determine whether to start the reselection process.

The following experiments investigate how subclusters are formed in a cluster and how the role of the clusterhead changes among the cluster members after carrying out the reselection operation. Given a randomized energy level to each sensor, figure 1 shows how the subclusters are formed in a specific cluster.

Figure 2 illustrates the relationship between the average number of subclusterheads and R/l ratio over 200 runs, where R/l is the ratio of transmitting range *R* to the side length *l* of the square. It shows that for a larger R/l ratio (i.e. a larger cluster), it is less likely to have a new clusterhead to include all sensors in the original cluster. This may be because a sensor can not be a clusterhead twice.

In order to validate the analysis of the procedure of the proposed algorithm, figure 3 shows the standard deviation of the mean number of subclusterheads when applying the ACAWT, the prediction formula, and the Modified Averaged Model, respectively. The plots vary the number of sensors

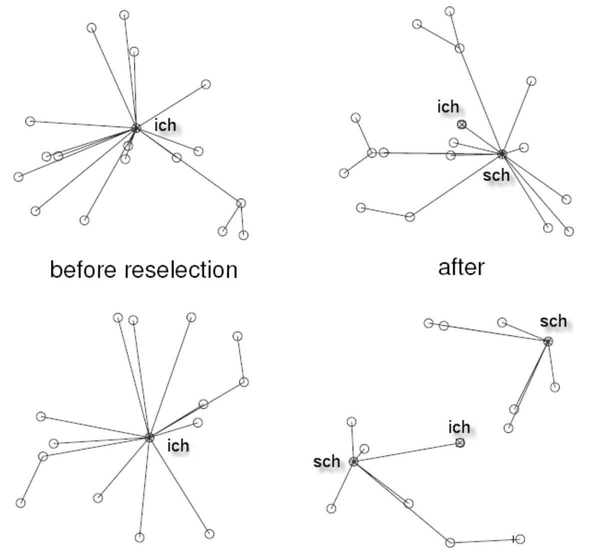

Fig. 1. A subcluster is formed in a specific cluster which requires reselection of a new clusterhead. In the subclustering formation "×" represents the initial clusterhead (ich); "∗" represents a subclusterhead (sch) using the ACAWT.

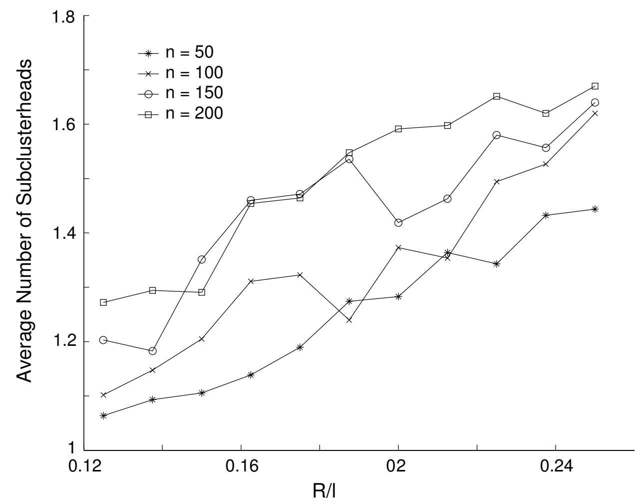

Fig. 2. Average number of subclusterheads as a function of the ratio R/l.

n and the transmission power R/l. Observe that compared with the algorithm, though the standard deviation of the the mean number of subclusterheads of the analysis is large, these results suggest that the modified Averaged Model is a way to approximately predict the performance of the ACAWT. The graphs also show that in most cases the number of subclusters in a cluster is either one or two at the first round.

For comparison, the same network topology and sensor energy level are used to study the performance of the two models during the first round. Let the threshold level η be $E_{max}/2$. Samples from the distributions, $E_{max} \cdot U(0,1)$ and $E_{max}/2 \cdot (1+U(0,1))$ are assigned to clusterheads and cluster members as the remaining energy, respectively. Figures 4 and 5 demonstrate typical runs of the ACAWT. It shows that this kind of local dynamic distribution of clusterheads allow each cluster to adjust its energy load among cluster members, which alleviates the problem that the battery of fixed clusterheads

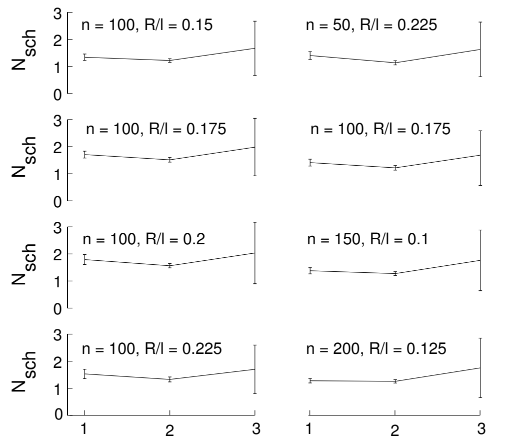

Fig. 3. The average number of subclusterheads formed in a cluster using (1) the Adaptive Clustering Algorithm, (2) the prediction formula, and (3) the Modified Averaged Model, respectively, with varying R/l ratio; the standard deviations are taken 200 runs.

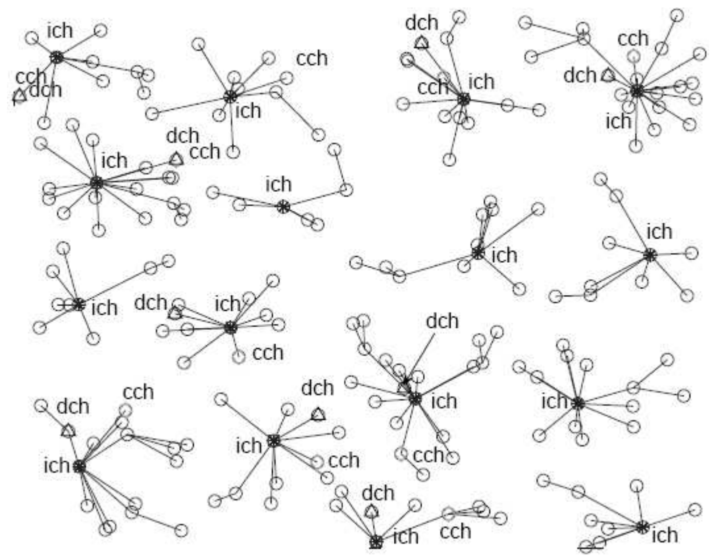

Fig. 5. Clusters are formed and clusterheads are reselected in a random network of 200 sensors with $R/l = 0.125$; “□” represents the initial clusterhead (ich); “◇” represents a new clusterhead using the centralized protocol (cch); “ △ ” represents a new clusterhead using the decentralized protocol (dch).

will drain quickly. Therefore, when the reselection operation is completed, the energy usage is spread among the network and thereby the lifetime of the network is extended.

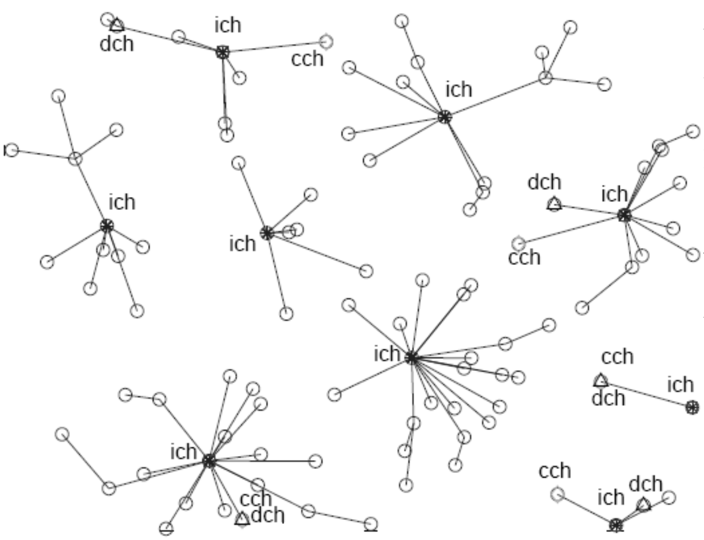

Fig. 4. Clusters are formed and clusterheads are reselected in a random network of 100 sensors with $R/l = 0.175$; “□” represents the initial clusterhead (ich); “◇” represents a new clusterhead using the centralized protocol (cch); “ △ ” represents a new clusterhead using the decentralized protocol (dch).

VII. Conclusion

This paper has presented a randomized, decentralized algorithm for re-clustering the sensors of an ad hoc network. A random waiting timer and a neighbor-based criterium are used to form clusters automatically. The centralized model and distributed model may be applied to execute the clusterhead-reselection process. The Modified Averaged Model is introduced for the purpose of understanding the performance of the clustering algorithm. Simulation results indicate that the analysis of the ACAWT agrees well with the behavior of the algorithm. Under the assumption of energy levels and power dissipation of the sensors, the analysis of cluster lifetime suggests that the proposed algorithm may be a solution to spread the energy usage over the network and achieve a better load balancing among clusterheads.

In this paper, we have assumed that the sensors function well and the communication environment is error free. In future, we plan to investigate certain failure scenarios in the network, such as the event of clusterhead failure, and consider fault-tolerance in the network operation.

References

[1] W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” in *Proceedings of IEEE HICSS*, January 2000.

[2] W. R. Heinzelman, A. Chandrakasan, H. Balakrishnan, “An application specific protocol architecture for wireless microsensor network,” in press: *IEEE Transaction on Wireless Networking*.

[3] M.J. Handy, M. Haase, D. Timmermann, “Low energy adaptive clustering hierarchy with deterministic cluster-head selection,” 4th *International Workshop on Mobile and Wireless Communications Network*, pp. 9-11, September 2002.

[4] C.F. Chiasserini, I. Chlamtac, P. Monti and A. Nucci, “Energy efficient design of wireless ad hoc networks,” in *Proceedings of European Wireless*, February 2002.

[5] M. Chatterjee, S. K. Das, and D. Turgut, “WCA: A weighted clustering algorithm for mobile ad hoc networks,” *Journal of Cluster Computing, Special issue on Mobile Ad hoc Networking*, No. 5, pp. 193-204, 2002.

[6] C.-Y. Wen and W. A. Sethares, “Automatic decentralized clustering for wireless sensor networks,” in *EURASIP Journal on Wireless Communications and Networking*, Volume 2005, Issue 5, pp. 686-697.

[7] A.D. Amis, and R. Prakash, “Load-balancing clusters in wireless ad hoc networks,” in *Proceedings of ASSET* 2000 , Richardson, Texas, March 2000.

[8] S. Basagni, “Distributed clustering for ad hoc networks,” in *Proceedings of International Symposium on Parallel Architectures, Algorithms and Networks*, pp. 310-315, June 1999.

[9] M.N. Halgamuge, S. M. Guru, and A. Jennings, “Energy efficient cluster formation in wireless sensor networks,” 10th *International Conference on Telecommunications*, vol.2, pp. 1571-1576, 2003.

[10] C. R. Lin and M. Gerla, “Adaptive clustering for mobile wireless networks,” *IEEE Journal on Selected Areas in Communication*, Vol. 15 pp. 1265-1275, September 1997.
[11] A. B. McDonald, and T. Znati, “A mobility based framework for adaptive clustering in wireless ad-hoc networks,” in *IEEE Journal on Selected Areas in Communications*, Vol. 17, No. 8, pp. 1466-1487, Aug. 1999.
[12] J. Lundelius and N. Lynch. “An upper and lower bound for clock synchronization.” *Information and Control*, Vol. 62 1984.
[13] M. Bhardwaj and A. P. Chandrakasan, “Bounding the lifetime of sensor networks via optimal role assignments,” in *IEEE INFOCOM* 2002, vol. 3, 2002, pp. 1587-1596.
[14] J. Zhu and S. Papavassiliou, “On the energy-efficient organization and the lifetime of multi-hop sensor networks,” in *IEEE Communications Letters*, vol. 7, no. 11, November 2003, pp. 537-539.
[15] I. Kang and R. Poovendran, “Maximizing static network lifetime of wireless broadcast ad hoc networks,” in *IEEE International Conference on Communications (ICC)* 2003, Anchorage, Alaska.
[16] P. Billingsley, *Probability and Measurement*, John-Wiley & Sons, Inc 1979.