Department of Computer Science Technical Reports

Department of Computer Science

2003

Distributed Clustering for Scalable, Long-Lived Sensor Networks

Ossama Younis

Sonia Fahmy
Purdue University, fahmy@cs.purdue.edu

Report Number:
03-026

DISTRIBUTED CLUSTERING FOR SCALABLE, LONG-LIVED SENSOR NETWORKS

Ossama Younis
Sonia Fahmy

Department of Computer Sciences
Purdue University
West Lafayette, IN 47907

CSD TR #03-026
July 2003

Distributed Clustering for Scalable, Long-Lived Sensor Networks

Ossama Younis and Sonia Fahmy
Department of Computer Sciences, Purdue University
250 N. University Street, West Lafayette, IN 47907–2066, USA
e-mail: {oyounis,fahmy}@cs.purdue.edu

***Abstract*—Prolonged network lifetime, scalability, and load balancing are important requirements for many ad-hoc sensor network applications. Clustering sensor nodes is an effective technique for achieving these goals. In this work, we propose a new hybrid energy-efficient approach for clustering nodes in ad-hoc sensor networks. Based on this approach, we present a protocol, *HEED* (Hybrid Energy-Efficient Distributed clustering), that periodically selects cluster heads according to a hybrid of their residual energy and a secondary parameter, such as node proximity to its neighbors or node degree. HEED does not make any assumptions about the distribution or density of nodes, or about node capabilities, e.g., location-awareness. The clustering process terminates in $O(1)$ iterations, and does not depend on the network topology or size. The protocol incurs low overhead in terms of processing cycles and messages exchanged. It also achieves fairly uniform cluster head distribution across the network. A careful selection of the secondary clustering parameter can balance load among cluster heads. Our simulation results demonstrate that HEED outperforms weight-based clustering protocols in terms of several cluster characteristics. We also apply our approach to a simple application to demonstrate its effectiveness in prolonging the network lifetime and supporting data aggregation.**

***Index Terms*—System design, simulations, sensor networks, clustering, energy efficiency, network lifetime**

I. Introduction

Sensor networks have recently emerged as an important computing platform [1], [2]. Sensor nodes are typically less mobile and more densely deployed than mobile ad-hoc networks (MANETs). Sensor nodes must be left unattended e.g., in hostile environments, which makes it difficult or impossible to re-charge or replace their batteries (solar energy is not always an option). This necessitates devising novel energy-efficient solutions to some of the conventional wireless networking problems, such as medium access control, routing, self-organization, bandwidth sharing, and security. Exploiting the tradeoffs among energy, accuracy, and latency, and using hierarchical (tiered) architectures are important techniques for prolonging network lifetime [1].

Network lifetime can be defined as the time elapsed until the first node (or the last node) in the network depletes its energy (dies). For example, in a military field where sensors are monitoring chemical activity, the lifetime of a sensor is critical for maximum field coverage. Energy consumption in a sensor node can be due to either "useful" or "wasteful" sources. Useful energy consumption can be due to (i) transmitting/receiving data, (ii) processing query requests, and (iii) forwarding queries/data to neighboring nodes. Wasteful energy consumption can be due to (i) idle listening to the media, (ii) retransmitting due to packet collisions, (iii) overhearing, and (iv) generating/handling control packets.

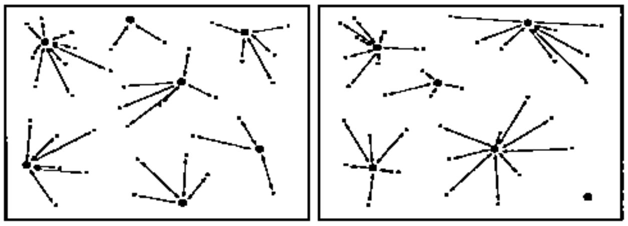
Several MAC protocols attempt to reduce energy consumption due to wasteful sources, e.g., [3], [4], [5], [6]. A number of protocols have also been proposed to reduce useful energy consumption. These protocols can be classified into three classes. Protocols in the first class control the transmission power level at each node to increase network capacity while keeping the network connected [7], [8]. Protocols in the second class make routing decisions based on power optimization goals, e.g., [9], [10], [11], [12], [13]. Protocols in the third class control the network topology by determining which nodes should participate in the network operation (be awake) and which should not (remain asleep) [14], [15], [16]. Nodes in this case, however, require knowledge of their locations via GPS-capable antennae or via message exchange.

Hierarchical (clustering) techniques can aid in reducing useful energy consumption [13]. Clustering is particularly useful for applications that require scalability to hundreds or thousands of nodes. Scalability in this context implies the need for load balancing and efficient resource utilization. Applications requiring efficient data

– This research has been sponsored in part by the Schlumberger Foundation technical merit award.

aggregation (e.g., computing the maximum detected radiation around an object) are natural candidates for clustering. Routing protocols can also employ clustering [17], [18]. In [19], clustering was proposed as a useful tool for efficiently pinpointing object locations. Clustering can be extremely effective in one-to-one, one-to-many, many-to-one, one-to-any, or one-to-all communication. For example, in many-to-one communication, clustering can support data fusion and reduce communication interference.

The essential operation in sensor node clustering is to select a set of cluster heads among the nodes in the network, and cluster the rest of the nodes with these heads. Cluster heads are responsible for coordination among the nodes within their clusters (intra-cluster coordination), and communication with each other and/or with external observers on behalf of their clusters (inter-cluster communication). Fig. 1 depicts two states of a clustered network, where the larger nodes denote cluster heads. Observe that clusters are non-overlapping and a unique cluster head exists in every cluster at any particular time. Periodic re-clustering can select nodes with higher residual energy to act as cluster heads. Network lifetime is prolonged through (i) reducing the number of nodes contending for channel access, (ii) summarizing network state information and updates at the cluster heads through intra-cluster coordination, and (iii) routing through an overlay among cluster heads, which is capable of controlling the network diameter.



(a) State i

(b) State i+1

Fig. 1. Two states of a clustered network

Many protocols proposed in the literature minimize energy consumption on routing paths. While these approaches increase energy efficiency, they do not necessarily prolong network lifetime if certain nodes are "popular," i.e., present on most forwarding paths in the network. Even if dynamic routing (in which data is forwarded to nodes with the highest residual energy) is used, it may cause such problems as unbounded delay and routing loops. With clustering, a popular node is guaranteed to "lose its popularity" (no longer serve as cluster head) after a fixed interval of time. Of course, node popularity due to interest in the data it provides can only be reduced by deploying several redundant nodes, and rotating among them (e.g., [14]).

Clustering protocols have been investigated as either stand-alone protocols for ad-hoc networks, e.g., [20], [21], [18], [22], [23], [24], or in the context of routing protocols, e.g., [7], [17], [25], [13]. In this work, we present a stand-alone distributed clustering approach that considers a hybrid of energy and communication cost. Based on this approach, we present a protocol, HEED (Hybrid Energy-Efficient Distributed clustering), which has four primary goals: (i) prolonging network lifetime by distributing energy consumption, (ii) terminating the clustering process within a constant number of iterations/steps, (iii) minimizing control overhead (to be linear in the number of nodes), and (iv) producing well-distributed cluster heads and compact clusters. HEED does not make any assumptions about the distribution or density of nodes, or about node capabilities, e.g., location-awareness. To the best of our knowledge, no previously proposed clustering protocol addressed these goals in an integrated manner.

The remainder of this paper is organized as follows. Section II describes the network model and states the problem that we address in this work. Section III briefly surveys related work. Section IV presents the HEED protocol and argues that it satisfies its goals. Section V shows its effectiveness via simulations, and compares it to other clustering techniques. Section VI discusses applications that can use our approach, and compares HEED with an optimized energy-efficient version of LEACH [13]. Finally, Section VII gives concluding remarks and directions for future work.

II. Problem Statement

Let the *clustering process interval*, T_{CP}, be the time taken by the clustering protocol to cluster the network. Let the *network operation interval*, T_{NO}, be the time between the end of a T_{CP} interval, the start of the subsequent T_{CP} interval. We must ensure that $T_{NO} \gg T_{CP}$ to reduce overhead.

A. Network Model

Assume a set of sensors is dispersed on a rectangular field. We assume the following properties about the network:

1) The nodes in the network are quasi-stationary.
2) The network serves multiple mobile/stationary observers, which implies that energy consumption is not uniform for all nodes.

3) Nodes are location-unaware, i.e. not equipped with GPS-capable antennae.
4) All nodes have similar capabilities (processing/communication), and equal significance.
5) Nodes are left unattended after deployment.
6) Each node has a fixed number of transmission power levels.

Our first assumption about mobility is typical for sensor networks. Clustering can still be performed, however, if only nodes that announce their willingness to be cluster heads are quasi-stationary during the T_{CP} interval in which they are selected, and the ensuing T_{NO} interval. Nodes that travel rapidly in the network may degrade the cluster quality, because they alter the node distribution in their cluster. The second network property motivates the requirement for re-clustering to select new cluster heads and re-distribute energy consumption. The third property justifies why some proposed protocols, such as [15], [26] are not suitable for our network. In addition, if scalability is an important concern, determining node locations using message exchange, as in [16], will not be efficient. The fourth and fifth properties of the network motivate the need for prolonging network lifetime and balancing cluster head loads. Note that node synchronization should not be essential. In Section IV-C and Section V-D, we show that unsynchronized nodes can still execute HEED independently, but cluster quality may be affected.

Note that in our model, *no* assumptions are made about any of the following:

1) homogeneity of node dispersion in the field,
2) network density or diameter,
3) distribution of energy consumption among sensor nodes,
4) proximity of querying observers.

B. The Clustering Problem

Assume that N nodes are dispersed in a field and the above assumptions hold. Our goal is to identify a set of cluster heads which cover the entire field. Each node v_i, where $1 \leq i \leq N$, is then mapped to exactly one cluster c_j (which has exactly one cluster head), where $1 \leq j \leq N_c$, and N_c is the number of clusters ($N_c \leq N$). The node can directly communicate with its cluster head (via a single hop). The following requirements must be met:

1) Clustering is completely distributed. Each node independently makes its decisions based on local information.
2) Clustering terminates within a fixed number of steps (regardless of network diameter).
3) At the end of each T_{CP}, each node is either a cluster head, or a non-head node (which we refer to as regular node) that belongs to exactly one cluster.
4) Clustering should be efficient in terms of processing complexity and message exchange.
5) Cluster heads are well-distributed over the sensor field.
6) The ratio of clusters having single nodes (only cluster heads) to the number of clusters, N_c, is small (in case of dense networks).

III. Related Work

Many protocols have been proposed for ad-hoc and sensor networks in the last five years. Reducing energy consumption due to wasteful sources has been primarily addressed in the context of adaptive MAC protocols, such as PAMAS [3], DBTMA [5], EAR [4], and S-MAC [6]. For example, S-MAC [6] periodically puts nodes to sleep to avoid idle listening and overhearing. TinyOS [27] focuses on fair bandwidth sharing among all nodes, and introduces random delays to unsynchronize nodes.

Data dissemination protocols proposed for sensor networks consider energy efficiency a primary goal [11], [10], [28], [12]. SPIN [11] attempts to reduce the cost of flooding data, assuming that the network is source-centric (i.e., sensors announce any observed event to interested observers). Directed diffusion [10], on the other hand, selects the most efficient paths to forward requests and replies on, assuming that the network is data-centric (i.e., queries and data are forwarded according to interested observers).

Clustering can be a side effect of other protocol operations. For example, in topology management protocols, such as GAF [15], SPAN [16], and ASCENT [14], nodes are classified according to their geographic location into equivalence classes. A fraction of nodes in each class (representatives) participate in the routing process, while other nodes are turned off to save energy. In GAF, geographic information is assumed to be available based on a positioning system such as GPS. SPAN infers geographic information through broadcast messages and routing updates. GAF, SPAN, and ASCENT share the same objective of using redundancy in sensor networks to turn radios on and off, and prolong network lifetime. In CLUSTERPOW [7], nodes are assumed to be non-homogeneously dispersed in the network. A node uses the minimum possible power level to forward data packets, in order to maintain connectivity while increasing the network capacity and saving energy. The Zone Routing Protocol (ZRP) [29] for MANETs divides the network into overlapping, variable-sized zones.

Several clustering techniques, such as K-Means, G-Means, or hierarchical clustering [30] have been proposed for partitioning datasets based on a parameter, e.g., distance. These approaches are not directly applicable to our problem because they iteratively optimize a cost function. This entails centralized control and excessive message exchange to propagate information. Several alternative distributed clustering approaches have been proposed for mobile ad-hoc networks and sensor networks. The Distributed Clustering Algorithm (DCA) [20] assumes quasi-stationary nodes with real-valued weights. The Weighted Clustering Algorithm (WCA) combines several properties in one parameter (weight) that is used for clustering. In [18], the authors propose using a spanning tree (or BFS tree) to produce clusters with some desirable properties. Energy efficiency, however, is not the primary focus of this work. In [25], the authors propose passive clustering for use with on-demand routing in ad-hoc networks. Earlier work also proposed clustering based on degree (connectivity) or lowest identifier heuristics [17]. Clustering time complexity in all of the above approaches is dependent on the network diameter, unlike HEED which terminates in a constant number of steps.

LEACH [13] is an application-specific data dissemination protocol that uses clustering to prolong the network lifetime. LEACH clustering terminates in a constant number of steps (like HEED), but it does not guarantee good cluster head distribution and assumes uniform energy consumption for cluster heads. In contrast, HEED makes no assumptions on energy consumption and selects well-distributed cluster heads (as discussed later). In [23], the authors use LEACH-like randomized clustering, but they provide methods to compute the optimal values of the algorithm parameters a priori. This approach, however, assumes that the nodes are homogeneously dispersed according to a Poisson process. It also does not guarantee good cluster head distribution. In [19], a multi-level hierarchical structure is proposed, where cluster heads are selected according to their residual energy and degree. This work, however, does not give enough practical details, such as how to combine residual energy and degree to select cluster heads.

IV. The HEED Protocol

In this section, we describe our protocol in detail. First, we define the parameters used in the clustering process. Second, we present the protocol design and pseudo-code. Finally, we prove that the protocol meets its requirements.

A. Clustering Parameters

The overarching goal of our approach is to prolong network lifetime. For this reason, cluster head selection is primarily based on the residual energy of each node. Note that residual energy measurement is not necessary, since the energy consumed per bit for sensing, processing, and communication is typically known. To increase energy efficiency and further prolong network lifetime, we also consider intra-cluster "communication cost" as a secondary clustering parameter. For example, cost can be a function of neighbor proximity or cluster density.

We use the primary parameter to probabilistically select an initial set of cluster heads, and the secondary parameter to "break ties." A tie in this context means that a node falls within the "range" of more than one cluster head, including the situation when two tentative cluster heads fall within the same range. To understand what "range" denotes in our context, observe that a node typically has a few (e.g., 6) discrete transmission power levels. As the power level increases, the sphere of coverage grows. Thus, the *cluster range* or *radius* is determined by the transmission power level used for intra-cluster announcements and during clustering. We refer to this as the *cluster power level*. The cluster power level should be set to one of the lower power levels of a node, to increase spatial reuse, and reserve higher power levels for inter-cluster communication. These higher power levels should cover two or more cluster ranges to guarantee that the resulting inter-cluster overlay will be connected. If this condition cannot be satisfied, then clustering is clearly not applicable. The cluster power level dictates the number of clusters in the network. It is practically difficult to determine an optimal cluster power level, because network topology changes due to node failures and energy depletion.

In case of multiple candidate cluster heads, cluster heads yielding lower intra-cluster communication cost are favored. This cost is a function of (i) cluster properties, such as cluster size, and (ii) whether or not variable power levels are permissible for transmission within a cluster, i.e., if each node is allowed to use the minimum power level to reach its cluster head or if all intra-cluster communication must use the same power level. If the power level used for intra-cluster communication is fixed for all nodes, then the cost can be proportional to (i) node degree, if the requirement is to distribute load among cluster heads, or (ii) $\frac{1}{node\ degree}$, if the requirement is to create dense clusters. This means that a node joins the cluster head with minimum degree to distribute cluster head load (possibly at the expense of increased interference and reduced spatial reuse), or joins

TABLE I
DEFINITIONS OF COMMUNICATION COST ACCORDING TO GOALS AND INTRA-CLUSTER COMMUNICATION POWER

Goal \ Power	Same	Minimum
Load distribution	node degree	AMRP
		node degree
Dense clusters	$\frac{1}{node\ degree}$	AMRP
		closest node

the one with maximum degree to create dense clusters. We use the terms *minimum degree cost* and *maximum degree cost* to denote these cost types. Observe that inter-cluster communication is not incorporated in the cost function since local information is insufficient in this case.

Now consider the case when variable power levels are allowed for intra-cluster communication. Let $MinPwr_i$ denote the minimum power level required by a node v_i, $1 \leq i \leq M$, to communicate with a cluster head u, where M is the number of nodes within the cluster range. We define the *average minimum reachability power (AMRP)* as the mean of the minimum power levels required by all M nodes within the cluster range to reach u, i.e., $AMRP = \frac{\sum_{i=1}^{M} MinPwr_i}{M}$. If each node is allowed to select the appropriate power level to reach its cluster head, then AMRP provides a good estimate of the communication cost. The AMRP of a node is a measure of the expected intra-cluster communication energy consumption if this node becomes a cluster head. Using AMRP as cost in selecting cluster heads is superior to just selecting the closest cluster head, since it provides a unified mechanism for all nodes, including cluster heads, to break ties among tentative cluster heads. If a node has to select its cluster head among nodes *not* including itself, the closest neighbor within its cluster range (the neighbor reached using the smallest power level) can be selected as its cluster head. Table I summarizes the different options for computing the communication cost.

B. Protocol Operation

As previously discussed in Section II, clustering is triggered every $T_{CP}+T_{NO}$ seconds to select new cluster heads. At each node, the clustering process requires a number of iterations (steps), which we refer to as N_{steps}. Every step takes time t_c, which should be long enough to receive messages from any neighbor within the cluster range. We set an initial percentage of cluster heads among all nodes N, C_{prob} (say 5%), assuming that an optimal percentage cannot be computed a priori. C_{prob} is only used to limit the initial cluster head announcements, and has no direct impact on the final clusters. Before a node starts executing HEED, it sets its probability of becoming a cluster head, CH_{prob}, as follows:

$$CH_{prob} = C_{prob} \times \frac{E_{residual}}{E_{max}} \qquad (1)$$

where $E_{residual}$ is the estimated current residual energy in the node, and E_{max} is a reference maximum energy (corresponding to a fully charged battery), which is identical for all nodes. The CH_{prob} value of a node, however, is not allowed to fall below a certain threshold p_{min} (e.g., 10^{-4}). This restriction is essential for terminating the algorithm in $O(1)$ steps, as we will show later.

During any step i, $i \leq N_{steps}$, every "uncovered" node (as defined below) elects itself to become a cluster head with probability CH_{prob}. After step i, the set of tentative cluster heads, S_{CH}, is set to $\{$cluster heads after step $i-1$ $\cup$ new heads selected in step $i\}$. A node v_i selects its cluster head (*my_cluster_head*) to be the node with the lowest cost in S_{CH} (S_{CH} may include v_i itself if it is selected as a tentative cluster head). Every node then doubles its CH_{prob} and goes to the next step. The pseudo-code for each node is given in Fig. 2. Note that if different power levels can be used for intra-cluster communication, then line 1 in phase I must be modified as follows: Discover neighbors within every power level $Pwr_i \leq Pwr_c$, where Pwr_c is the cluster power level. In this case only, we assume that if cluster head u can reach a node v with power level l, then v can reach u with level l as well. Note also that if a node elects to become a cluster head, it sends an announcement message *cluster_head_msg(Node_ID, selection status, cost)*, where the selection status is set to *tentative_CH*, if its CH_{prob} is less than 1, or *final_CH*, if its CH_{prob} has reached 1. A node considers itself "covered" if it has heard from either a *tentative_CH* or a *final_CH*. If a node completes HEED execution without selecting a cluster head that is *final_CH*, it considers itself uncovered, and announces itself to be a cluster head with state *final_CH*.

C. Correctness and Complexity

The protocol provided in Fig 2 meets the requirements listed in Section II-B, as discussed next.

Observation 1: HEED is completely distributed (requirement 1). A node can either elect to become a cluster head according to its CH_{prob}, or join a cluster according to overheard cluster head messages within its cluster range. Thus, node decisions are based solely on local information.

Fig. 2. HEED protocol pseudo-code

```
I. Initialize
1.  $S_{nbr} \leftarrow \{v: v$ lies within my cluster range$\}$
2.  Compute and broadcast cost to $\in S_{nbr}$
3.  $CH_{prob} \leftarrow max(C_{prob} \times \frac{E_{residual}}{E_{max}}, p_{min})$
4.  is_final_CH $\leftarrow$ FALSE

II. Repeat
1.  If (($S_{CH} \leftarrow \{v: v$ is a cluster head$\}) \neq \phi$)
2.     my_cluster_head $\leftarrow$ least_cost($S_{CH}$)
3.     If my_cluster_head = NodeID
4.        If ($CH_{prob}$ = 1)
5.           Cluster_head_msg(NodeID,final_CH,cost)
6.           is_final_CH $\leftarrow$ TRUE
7.        Else
8.           Cluster_head_msg(NodeID,tentative_CH,cost)
9.  ElseIf ($CH_{prob}$ = 1)
10.    Cluster_head_msg(NodeID,final_CH,cost)
11.    is_final_CH $\leftarrow$ TRUE
12. ElseIf Random(0,1) $\leq CH_{prob}$
13.    Cluster_head_msg(NodeID,tentative_CH,cost)
14. $CH_{previous} \leftarrow CH_{prob}$
15. $CH_{prob} \leftarrow min(CH_{prob} \times 2, 1)$
Until $CH_{previous} = 1$

III. Finalize
1.  If (is_final_CH = FALSE)
2.     If (($S_{CH} \leftarrow \{v: v$ is a final cluster head$\}) \neq \phi$)
3.        my_cluster_head $\leftarrow$ least_cost($S_{CH}$)
4.        join_cluster(cluster_head_ID, NodeID)
5.     Else Cluster_head_msg(NodeID, final_CH, cost)
6.  Else Cluster_head_msg(NodeID, final_CH, cost)
```

Lemma 1: HEED terminates in $N_{steps} = O(1)$ steps (requirement 2).

Proof. The worst case occurs when a node has a very low $E_{residual}$. This node will start executing HEED with CH_{prob} set to p_{min}. However, CH_{prob} doubles in every step, and phase II of the protocol terminates one step (iteration) after CH_{prob} reaches 1. Therefore,

$$2^{N_{steps}-1} \times p_{min} \geq 1$$

and hence

$$N_{steps} \leq \lceil log_2 \frac{1}{p_{min}} \rceil + 1 \tag{2}$$

Therefore, $N_{steps} \approx O(1)$. □

With the appropriate choice of the minimum probability of becoming a cluster head, the number of steps can be bounded by a reasonable constant (requirement 2). For example, for $p_{min} = 10^{-4}$, a low-energy node will need 15 steps in phase II. When $E_{residual}$ is close to E_{max}, the number of steps is much lower, and depends on the value of C_{prob}. For example, for $C_{prob} = 5\%$, high-energy nodes will exit HEED in only 6 steps. Thus, nodes with high residual energy will terminate HEED earlier than nodes with lower residual energy. This allows low energy nodes to join their clusters.

Lemma 2: At the end of phase III of the HEED protocol, a node is either a cluster head or a regular node that belongs to a cluster (requirement 3).

Proof. Assume that a node terminates its execution of HEED without electing to become a cluster head or joining a cluster. This implies that the condition in line 1 of phase III is satisfied, while the condition in line 2 is not satisfied (hence, line 4 is not executed). In this case, line 5 will be executed, and the node will become a cluster head, which is a contradiction. □

Observation 2: After executing HEED, a node is covered by at most one cluster head.

Lemma 3: HEED has a worst case processing time complexity of $O(N)$, where N is the number of nodes in the network (requirement 4).

Proof. Phase I in the HEED protocol takes a processing time of at most N to compute the cost, if the cost definition is the AMRP. Similarly, phase III also takes a processing time of at most N to arbitrate among the nodes which declared their willingness to be cluster heads with state *final_CH*. For Phase II, the time taken to arbitrate among cluster heads (for all passes) is at most $N_{steps} \times N$ cluster heads. From Lemma 1, N_{steps} is a constant. Therefore, the total time is still $O(N)$. All other steps have an $O(1)$ time complexity. Therefore, the total processing complexity is $O(N)$. □

Lemma 4: HEED has a worst case message exchange complexity of $O(N)$, where N is the number of nodes in the network (requirement 4).

Proof. During the execution of HEED, a tentative cluster head generates at most N_{steps} cluster head messages. Hence, the number of these messages in the network is upper-bound by $N_{steps} \times N$. A regular node is silent until it sends a join message to a cluster head. The number of these join messages is strictly less than

N, since at least one node will decide to be a cluster head with state *final_CH* during the clustering process. Therefore, the number of messages exchanged during the clustering process is $O(N)$. □

Lemma 5: The probability that two nodes within each other's cluster range are both cluster heads is small, i.e., cluster heads are well-distributed (requirement 5).

Proof. Consider the following worst case scenario. Assume that v_1 and v_2 are two isolated neighboring nodes (i.e., each one does not have any other neighbor in close proximity). We compute the probability, p_{nbr}, that at the end of phase III, both of them are cluster heads (we assume that they are fully synchronized). Assume that neither of the two nodes decides to be a cluster head before its CH_{prob} reaches 1. Otherwise, one of them will concede to the other. Two cases may occur in this scenario:

Case 1: The CH_{prob} values of v_1 and v_2 are different enough such that they do not execute the same number of steps in phase II. Without loss of generality, assume that $CH_{prob1} > CH_{prob2}$. In this case, v_1 will elect to become a cluster head with state *final_CH* before v_2. Hence, v_2 will receive a cluster head message and register with v_1. The same argument applies for unsynchronized nodes, because they will likely terminate their computations at different times. That is why we state in Section II-A that synchronization is not critical for HEED operation.

Case 2: v_1 and v_2 will execute the same number of steps in phase II. In this case, at any step $i < N_{steps}$, neither v_1 nor v_2 decides to be a cluster head with probability $p_i = (1 - CH_{prob1})(1 - CH_{prob2})$. Let $prob_1$ denote the initial CH_{prob1}, and $prob_2$ denote the initial CH_{prob2}. During step i, $0 \leq i \leq N_{steps} - 2$, the current $CH_{prob1} = prob_1 \times 2^i$ and $CH_{prob2} = prob_2 \times 2^i$. Let p_{nbr} be the probability that neither v_1 nor v_2 elects to become a cluster head at any step i:

$$p_{nbr} = \prod_{i=0}^{N_{steps}-2} (1 - prob_1 \times 2^i)(1 - prob_2 \times 2^i) \quad (3)$$

When $prob_1 = prob_2 = p$, we get

$$p_{nbr} = \prod_{i=0}^{(\lceil log \frac{1}{p} \rceil - 1)} (1 - p \times 2^i)^2 \quad (4)$$

With typical values of the initial CH_{prob} for all nodes, the probability p_{nbr} is very small. For example, for p=3%, the resulting p_{nbr}=0.00016, while for p=5%, the resulting p_{nbr}=0.006. A loose upper bound for Eq. (4) is $p_{nbr} < e^{-2p(1+2+4+...+2^{(\lceil log \frac{1}{p} \rceil - 1)})}$, or $p_{nbr} < e^{-2p(2^{\lceil log \frac{1}{p} \rceil} - 1)}$. This probability, however, is expected to be much smaller in practical situations, in which a node is likely to have more than one neighbor. In addition, similar starting CH_{prob} values will not be the common case after the network operates for a few rounds. □

In all our experiments in Section V, no two neighboring nodes were chosen as cluster heads in HEED (note that centralized approaches to achieve this, such as graph coloring algorithms, are of course unsuitable in this case). This property remained valid with different transmission ranges, variable node density, and different cost types.

V. Performance Evaluation

In this section, we evaluate the performance of the HEED protocol via simulations. Unless otherwise specified, we assume that 1000 nodes are uniformly dispersed into a field with dimensions 2000 × 2000. We set the minimum probability for becoming a cluster head (p_{min}) to 0.0005. In this case, the maximum number of steps that HEED may take at any node is 12 (according to Lemma 1). Initially, $CH_{prob} = C_{prob} = 5\%$ for all nodes. Wireless transmission laws dictate that power attenuation be proportional to the square of the covered distance (assuming fixed transmission power). If the distances are small (hundreds of meters), then the power attenuation can be assumed to be linear with the transmission radius [31]. Practically, other factors may also affect the received power, such as noise or physical obstacles. For simplicity, we assume the absence of these factors in our experiments, and therefore use the distance between nodes to account for the required transmission power level among them. We vary the cluster radius (range) from 25 to 400 to study how the protocol works with low to high coverage ranges. Every result shown is the average of 100 experiments. Each experiment uses a different randomly-generated topology, where each node is assigned a different randomly-generated residual energy level between 0 and 1 Joule (J). Residual energy is discretized into 20 levels to increase ties.

We compare HEED to a generic weight-based clustering protocol that is suitable for quasi-stationary ad-hoc networks. DCA [20] and WCA [21] are examples of such weight-based clustering. In our experiments, the real-valued weight used for generic clustering is simply the node residual energy. During any step of the clustering process, a node does not make a decision about which cluster to join (or if it should become a cluster head itself) until all neighboring nodes with higher weights have already decided (similar to DCA [20]). This generic clustering (GC) protocol is a good baseline for comparison because it has the following features:

- Clustering is distributed and only based on local information,
- Selected cluster heads are guaranteed to be the nodes with the highest weights (residual energy) within their clusters,
- A node is associated with only one cluster head,
- No underlying assumptions about node dispersion in the field are made,
- The number of steps of the protocol is a function of network diameter, similar to most currently proposed clustering approaches in mobile ad-hoc networks,
- The time and message complexities are $O(N)$, and
- It is guaranteed that no two cluster heads are neighbors, i.e. cluster heads are well-distributed in the network field.

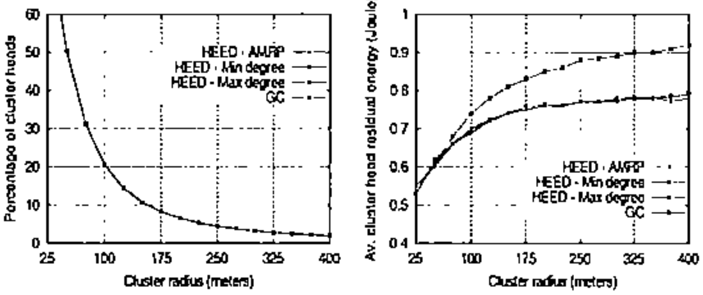
In this section, we compare HEED to the GC protocol in terms of: (i) number of steps required for the clustering process, (ii) ratio of the number of clusters to the number of nodes in the network, (iii) ratio of clusters with more than one node to the number of clusters, (iv) standard deviation of the number of nodes in a cluster, and maximum number of nodes in a cluster, and (v) average residual energy of the selected cluster heads. We also study the case where nodes are not fully synchronized. Observe that clustering metrics proposed in literature, such as Calinski's criterion [32], within-scatter [33], and the Bayesian Information Criterion (BIC) [34] are not the best metrics for our evaluation. This is because they are not easy to adapt to multiple heterogeneous parameters, like residual energy and node degree. In addition, these metrics favor a smaller number of compact clusters, which is not necessarily our goal in all cases.

A. Steps to Terminate

We compare the number of steps required for HEED and GC protocols to terminate. As previously discussed, the number of steps in HEED can be deterministically computed using Lemma 1, which is independent of the cluster radius. For GC, the number of steps grows quickly as the cluster radius increases. This can be attributed to the fact that a larger cluster radius implies more neighbors for each node. Thus, a node will have to wait longer for higher weight nodes to decide which clusters to join. Our experiments show that GC takes only 3 steps to terminate for a cluster radius of 25. The number of steps, however, grows to 85 for a cluster radius of 400. HEED takes 6 steps to terminate for all cluster ranges.

B. Cluster Head Characteristics

The number of selected cluster heads varies according to the specified cluster radius. The smaller the radius, the larger the required number of cluster heads to fully cover the entire network. HEED cluster heads are comparable to those selected by GC in terms of number, distribution, and energy availability. Lemma 5 proves that the selected cluster heads in HEED are well-distributed. Fig. 3(a) shows that the average number of cluster heads selected by both GC and HEED (with different cost types) are almost identical. This is not surprising, since both GC and HEED tend to select cluster heads that are not neighbors within a cluster radius. The percentage of cluster heads is very high (80%) for very small cluster ranges, and becomes smaller as the range increases.



(a) Percentage of cluster heads

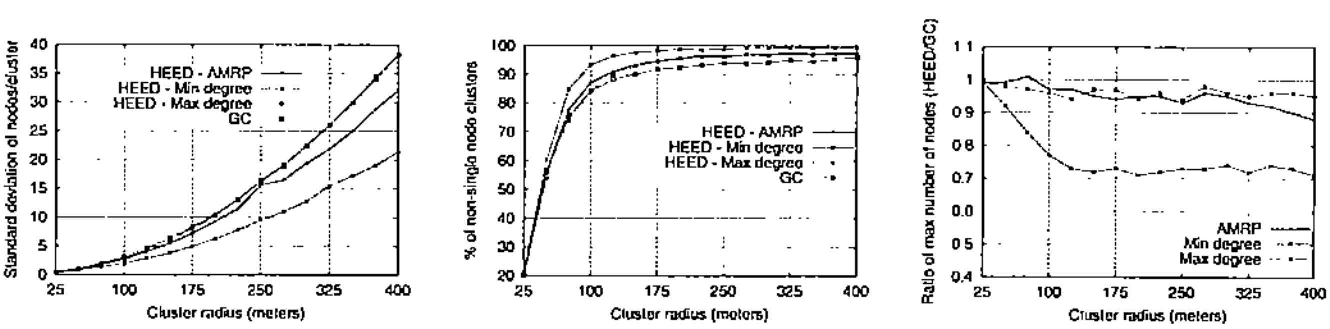
(b) Average residual energy per cluster head

Fig. 3. Characteristics of selected cluster heads

In HEED, tentative cluster heads are randomly selected based on their residual energy. Therefore, HEED cannot guarantee optimal head selection in terms of energy, since it uses the secondary parameter to resolve conflicts. GC, a weight-based approach, does guarantee that the highest energy node will be the cluster head within its cluster range. Fig. 3(b) compares the two protocols in terms of residual energy. The results show that the cluster heads selected by HEED have high residual energy, and their average residual energy is not far lower than that with GC (at most 12% difference).

C. Cluster Characteristics

Application requirements dictate which cluster characteristics are favored in particular contexts. If it is required to balance load on cluster heads, then it is important to have clusters with small variance in the number of nodes they cover. Fig. 4(a) illustrates the standard deviation of the number of nodes per cluster for each cost type (cost types were defined in Section IV-A). The maximum degree cost type and GC show similar



(a) Standard deviation of number of nodes/cluster

(b) Percentage of non-single-node clusters

(c) Ratio of maximum number of nodes in a HEED cluster to a GC cluster

Fig. 4. Characteristics of clusters

results. For minimum degree cost, the standard deviation is the lowest, because ties are broken by joining the smaller degree node, thus balancing the cluster sizes. AMRP results lie between the two extremes. Therefore, AMRP provides a a compromise between load balancing and cluster density.

Another appealing cluster property is minimizing clusters with only a single node (the cluster head). Single-node clusters arise when a node is forced to represent itself (because of not receiving any cluster head messages). A cluster may also contain a single node if this node decides to act as a cluster head, and due to cost definition, all its neighbors register themselves with other cluster heads. Fig. 4(b) illustrates the percentage of clusters with more than one node. The figure shows that HEED produces a higher percentage of non-single-node clusters than GC for all cost types. It is also worth mentioning that minimum degree cost results are superior to all other types because it balances cluster sizes.

We also consider the maximum number of nodes in a cluster. Fig. 4(c) shows that the maximum number of nodes in a cluster in HEED is on the average smaller than that of GC for all cost types, but especially for the minimum degree cost. Together with the results about variance in the number of nodes in a cluster, presented in Fig. 4(a), we can conclude that HEED produces balanced clusters.

It is important to note that we have repeated all our previous experiments with highly non-uniform node dispersion. We find that HEED performance relative to GC remains the same. We also observe that the average percentage of cluster heads is much lower in the non-uniform case than in the uniform case. This is at the expense of a much higher variance in the number of nodes per cluster. Moreover, the average residual energy of cluster heads is slightly higher (on the average) in the case of non-uniformly dispersed nodes. This applies to both GC and to HEED with different cost types.

D. Node Synchronization

In Section II-A, we claimed that node synchronization is not critical for the operation of HEED. We argued why this claim holds in the proof of Lemma 5 (Case 1). We have conducted a number of experiments to study the effect of synchronization on the average cluster head energy. To compare with a non-fully synchronized (i.e., pseudo-synchronized) case, we assume that every node starts the clustering process randomly within a 3 × t_c interval, i.e., within 3 steps of the start of clustering process. This is a reasonable choice since using $C_{prob} = 0.05$ implies that phase II terminates in 6 steps in the case of a fully-charged battery. Fig. 5 illustrates the average cluster head energy for networks with synchronized versus pseudo-synchronized nodes (labeled "unsynch"). Results indicate that the selected cluster heads in both cases have comparable residual energy. Results for other cluster and cluster head characteristics were also found to be similar to those presented above.

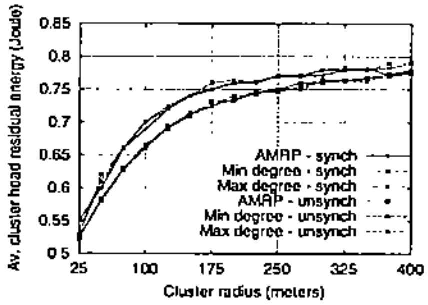


Fig. 5. HEED average cluster head energy for synchronized and pseudo-synchronized nodes

VI. Clustering Applications

Our approach can be used for constructing energy-efficient hierarchies for routing protocols, in which higher tier nodes should have more residual energy. Our approach can also be effective for sensor applications requiring efficient data aggregation. This is because prolonging network lifetime is especially important for unattended networks used in environmental monitoring. We consider one such application, LEACH [13], in this section.

In [13], a distributed clustering protocol for micro-sensor networks (LEACH) was introduced for prolonging the network lifetime. LEACH was proposed for an application in which sensor nodes are randomly distributed on a grid-like area and are continuously sensing the environment to send reports to a remote sink (e.g., observer/base station). The application assumes that nodes are equally significant and data aggregation is possible. LEACH clustering proved to be 4× to 8× more effective in prolonging the network lifetime than direct communication or minimum energy transfer (shortest path multi-hop routing).

In LEACH, a node elects to become a cluster head according to a target number of cluster heads in the network and its own residual energy. This can be performed in a single step if the node blindly elects itself according to whether or not it has previously acted as a cluster head. Another option is for a node to elect itself according the ratio of its residual energy to the total residual energy in the network. This approach, however, requires that residual energy of all nodes be propagated throughout the entire network, and thus has a higher communication overhead. LEACH clustering starts by computing the optimal number of clusters in the network k_{opt}, which is a function of the propagation model, energy consumed per bit, number of nodes, grid length, and distance between the cluster heads and the sink. When clustering is triggered, certain nodes broadcast their willingness to become cluster heads, and regular nodes join clusters according to cluster head proximity. Each cluster head then creates a TDMA schedule for its nodes and broadcasts it. Every node sends its data to its cluster head according to the specified TDMA schedule. Direct Sequence Spread Spectrum (DSSS) codes are used to minimize inter-cluster interference (therefore, we ignore collisions in our simulation). Each cluster head fuses the data it receives from its nodes into one frame and sends it to the sink. Clustering is triggered every $N_{trigger}$ TDMA frames.

TABLE II
Simulation Parameters

Parameter	Value
Network grid	From (0,0) to (100,100)
Sink	At (50,175)
Threshold distance (d_0)	75 m
Cluster radius	25 m
E_{elec}	50 nJ/bit
ϵ_{fs}	10 $pJ/bit/m^2$
ϵ_{mp}	0.0013 $pJ/bit/m^4$
E_{fusion}	5 $nJ/bit/signal$
Data packet size	100 bytes
Broadcast packet size	25 bytes
Packet header size	25 bytes
Round ($N_{trigger}$)	5 TDMA frames
Initial energy	2 J/battery

We compare our HEED clustering to an optimized LEACH (opt-LEACH) approach in which the routing protocol is assumed to propagate node residual energy throughout the network. Although this approach requires extensive message exchange (for residual energy information), it selects better cluster heads than the original LEACH, and thus prolongs network lifetime. A node executing opt-LEACH elects itself to become a cluster head at time t with probability $CH_{prob}(t)$, where $CH_{prob}(t)$ = $\min(\frac{E_i(t)}{E_{total}} \times k, 1)$. Here, E_i is the residual energy of node i, and $E_{total} = \sum_{i=1}^{N} E_i(t)$.

Most of our simulation parameters are similar to those in [13], and are listed in Table VI. In the simple radio model that we use, energy is expended to serve: (i) digital electronics, E_{elec}, (actuation, sensing, signal emission/reception), and (ii) communication, E_{amp}. E_{amp} varies according to the distance d between a sender and a receiver: $E_{amp} = \epsilon_{fs}$ assuming a free space model when $d < d_0$, while $E_{amp} = \epsilon_{mp}$ assuming a multipath model when $d \geq d_0$, where d_0 is a constant distance that depends on the environment. To transmit n_b bits for a distance d, the radio expends $n_b(E_{elec} + E_{amp} \times d^n)$ J, where $n = 2$ for $d < d_0$, and $n = 4$ for $d \geq d_0$. To receive n_b bits at the receiver, the radio expends $n_b \times E_{elec}$ J. This energy model assumes a continuous function for energy consumption.

A node is considered "dead" if it has lost 99.9% of its initial energy. For HEED, 5% is used as an initial tentative percentage of cluster heads (C_{prob}). For opt-LEACH, k_{opt} was selected to be 11 for 300–700 node networks, which falls in the range of k_{opt} computed according to [13]. Fig. 6 compares network lifetime with HEED to opt-LEACH, where network lifetime is the time until the *last* node dies. HEED clustering clearly improves network lifetime over opt-LEACH clustering for all cost types. This is because opt-LEACH randomly selects cluster heads (and hence cluster sizes), which may result in faster death of some nodes. This is avoided in HEED because final cluster heads are selected such that they are well-distributed across the network and communication cost is minimized. When we measure the number of rounds until the *first* node dies, we find that the first opt-LEACH node dies after approximately 150 rounds, while the first HEED node only dies after 350–550 rounds.

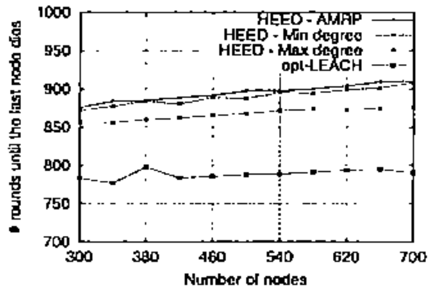


Fig. 6. Network lifetime using HEED and opt-LEACH (last node death)

We also measure the energy consumed in clustering as a fraction of the total dissipated energy in the network. For opt-LEACH, we assume that at the end of each round, each node sends its residual energy information to its cluster head, which aggregates this information and broadcasts it across the network using only one message. Fig. 7 illustrates the energy ratio for different numbers of nodes (the results of the three HEED cost types are almost superimposed). HEED expends less energy in clustering than opt-LEACH, although its clustering process requires more than one step for each node. This can be attributed to the energy consumed by opt-LEACH for propagating residual energy information.

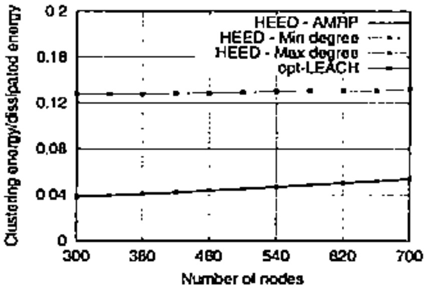


Fig. 7. Ratio of energy used in clustering to total dissipated energy

Finally, we study the effect of the distance between the sink and the network on the network lifetime (using the "last node death" definition of network lifetime). In this experiment, we compute the number of rounds in which the network was alive using different HEED cost types, opt-LEACH, and direct communication. We fix the x-coordinate of the sink and varied its height (y-coordinate). The distance is computed from the sink to the closest point to it on the network. The number of nodes was fixed at 500. Fig. 8 shows that HEED prolongs network lifetime, compared to opt-LEACH and to direct communication. Network lifetime severely deteriorates when using direct communication as the distance increases, which emphasizes the advantages of network clustering. Using direct communication may be tolerable only when the sink is very close to the network (which is not the case in this application), to avoid clustering overhead.

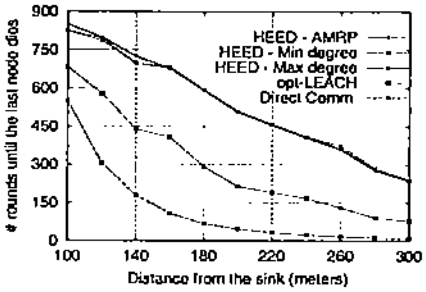


Fig. 8. Network lifetime for a 500-node network as the sink travels farther

VII. Conclusions and Future Work

In this paper, we have presented an energy-efficient distributed clustering approach for ad-hoc sensor networks. Our approach is hybrid: cluster heads are randomly selected based on their residual energy, and nodes join clusters such that communication cost is minimized. Based on this approach, we have introduced the HEED protocol, which terminates in a constant number of steps, independent of the network diameter. HEED operates in quasi-stationary networks where nodes are location-unaware and have equal significance. No assumptions are made about the node dispersion or density in the field. Simulation results show that HEED prolongs network lifetime, and the clusters it produces exhibit several appealing characteristics. HEED parameters, such as the minimum selection probability and network operation interval, can be easily tuned to optimize resource usage according to the network density and application requirements.

Our approach can be applied to the design of several types of sensor network protocols that require energy efficiency, scalability, prolonged network lifetime, and load balancing. Although we have only provided a protocol for building a single cluster layer, we can extend the protocol to multi-level hierarchies. This can be achieved by recursive application at upper tiers using bottom-up cluster formation (similar to [23]). We are currently investigating cluster size constraints in HEED. We are also incorporating HEED into a multiple hop power-aware routing model for sensor networks with multiple external mobile observers.

ACKNOWLEDGMENTS

We are grateful to Gopal Panduranagan (Purdue University) for his valuable comments on this work.

REFERENCES

[1] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the World with Wireless Sensor Networks," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2001), Salt Lake City, Utah*, May 2001.

[2] G. J. Pottie and W. J. Kaiser, "Wireless Integrated Newtork Sensors," *Communications of the ACM*, vol. 43, no. 5, pp. 51–58, May 2000.

[3] S. Singh and C. Raghavendra, "PAMAS: Power Aware Multi-Access protocol with Signalling for Ad Hoc Networks," *ACM Computer Communication Review*, vol. 28, no. 3, pp. 5–26, July 1998.

[4] Katayoun Sohrabi, Jay Gao, Vishal Ailawadhi, and Gregory J. Pottie, "Protocols for Self-Organization of a Wireless Sensor Network," *IEEE Personal Comm. Mag.*, vol. 7, no. 5, Oct. 2000.

[5] Z. Haas and S. Tabrizi, "On Some Challenges and Design Choices in Ad-Hoc Communications," in *IEEE MILCOM'98*, October 1998.

[6] W. Ye, J. Heidenmann, and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," in *Proceedings of IEEE INFOCOM*, New York, NY, June 2002.

[7] V. Kawadia and P. R. Kumar, "Power Control and Clustering in Ad Hoc Networks," in *Proceedings of IEEE INFOCOM*, April 2003.

[8] S. Narayanaswamy, V. Kawadia, R. S. Sreenivas, and P. R. Kumar, "Power Control in Ad-Hoc Networks: Theory, Architecture, Algorithm and Implementation of the COMPOW protocol," in *Proceedings of European Wireless 2002. Next Generation Wireless Networks: Technologies, Protocols, Services and Applications*, February 2002, pp. 156–162.

[9] J. Gomez, A.T. Campbell, M. Naghshineh, and C. Bisdikian, "Conserving Transmission Power in Wireless Ad Hoc Networks," in *Proceedings of ICNP'01*, November 2001.

[10] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," in *Proceedings of ACM/IEEE MOBICOM*, 2000.

[11] J. Kulik, W. Rabiner Heinzelman, and H. Balakrishnan, "Negotiation-Based Protocols for Disseminating Information in Wireless Sensor Networks," *ACM Wireless Networks*, vol. 8, no. 2-3, pp. 169–185, 2002.

[12] Jae-Hwan Chang and Leandros Tassiulas, "Energy conserving routing in wireless ad-hoc networks," in *Proceedings of IEEE INFOCOM*, Tel Aviv, Israel, Mar. 2000.

[13] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, October 2002.

[14] A. Cerpa and D. Estrin, "ASCENT: Adaptive Self-Configuring Sensor Networks Topologies," in *Proceedings of IEEE INFOCOM*, New York, NY, June 2002.

[15] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed Energy Conservation for Ad Hoc Routing," in *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, Rome, Italy, July 2001, pp. 70–84.

[16] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: an Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks," *ACM Wireless Networks*, vol. 8, no. 5, September 2002.

[17] C. R. Lin and M. Gerla, "Adaptive Clustering for Mobile Wireless Networks," in *IEEE J. Select. Areas Commun.*, September 1997.

[18] S. Banerjee and S. Khuller, "A Clustering Scheme for Hierarchical Control in Multi-hop Wireless Networks," in *Proceedings of IEEE INFOCOM*, April 2001.

[19] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: scalable coordination in sensor networks," in *Proceedings of ACM/IEEE MOBICOM*, August 1999.

[20] S. Basagni, "Distributed Clustering Algorithm for Ad-hoc Networks," in *International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN)*, 1999.

[21] M. Chatterjee, S. K. Das, and D. Turgut, "WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks," *Cluster Computing*, pp. 193–204, 2002.

[22] T. J. Kwon and M. Gerla, "Clustering with Power Control," in *Proceeding of MilCOM'99*, 1999.

[23] S. Bandyopadhyay and E. Coyle, "An Energy-Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks," in *Proceedings of IEEE INFOCOM*, April 2003.

[24] A. D. Amis, R. Prakash, T. H. P. Vuong, and D. T. Huynh, "Max-Min D-Cluster Formation in Wireless Ad Hoc Networks," in *Proceedings of IEEE INFOCOM*, March 2000.

[25] M. Gerla, T. J. Kwon, and G. Pei, "On Demand Routing in Large Ad Hoc Wireless Networks with Passive Clustering," in *Proceeding of WCNC*, 2000.

[26] F. Ye, H. Luo, J. Chung, S. Lu, and L. Zhang, "A Two-Tier Data Dissemination Protocol for Large-Scale Wireless Sensor Networks," in *Proceedings of ACM/IEEE MOBICOM*, Atlanta, Georgia, September 2002.

[27] Alec Woo and David Culler, "A transmission control scheme for media access in sensor networks," in *Proceedings of the ACM/IEEE MOBICOM*, 2001.

[28] D. Braginsky and D. Estrin, "Rumor Routing Algorithm for Sensor Networks," in *Proceedings of the First Workshop on Sensor Networks and Application (WSNA'02)*, September 2002.

[29] Z. J. Haas, M. R. Pearlman, and P. Samar, "The Zone Routing Protocol (ZRP) for Ad-hoc Networks," in *Internet Draft, draft-ietf-manet-zone-zrp-04.txt*, July 2002.

[30] J. Han and K. Micheline, *Data Mining Concepts and Techniques*, Morgan Kauffman, 2001.

[31] W. C. Y. Lee, *Mobile Cellular Telecommunications*, McGraw Hill, 1995.

[32] R. B. Calinski and J. Harabasz, "A Dendrite Method for Cluster Analysis," *Communications in Statistics*, vol. 3, pp. 1–27, 1985.

[33] G. W. Mulligan and M. C. Cooper, "An Examination of Procedures for Determining the Number of Clusters in a Data Set," *Psychometrika*, vol. 50, pp. 159–179, 1985.

[34] G. Schwartz, "Esatimating the dimension of a model," *Annals of statistics*, vol. 6, pp. 461–464, 1978.