A Local Cluster Head Election Algorithm in Wireless Sensor Networks

Hoda Taheri
Dept. Computer Engineering
Young Researchers Club, Mashhad branch,
Islamic Azad University
Mashhad, Iran
taheri@ieee.org

Peyman Neamatollahi
Dept. Computer Engineering
Young Researchers Club, Mashhad branch,
Islamic Azad University
Mashhad, Iran
neamatollahi@ieee.org

Mohammad Hossein Yaghmaee
Dept. Computer Engineering
Mashhad branch, Islamic Azad University
Mashhad, Iran
yaghmaee@ieee.org

Mahmoud Naghibzadeh
Dept. Computer Engineering
Ferdowsi University of Mashhad
Mashhad, Iran
naghibzadeh@um.ac.ir

***Abstract*—Clustering is one of the basic approaches for designing energy-efficient and scalable distributed sensor networks. HEED is a well known energy efficient clustering protocol. In this protocol, the message overhead is low and cluster heads are distributed fairly across the network. Our distributed clustering protocol, HEED*, improves HEED protocol to achieve energy efficiency with three techniques: 1) local clustering, i.e., whenever a cluster head consumes a prespecified part of its energy, it only informs its member nodes to hold cluster head elections for the upcoming round. Therefore, clustering is performed locally (in contrast to performing it globally). 2) Reducing the number of message exchanges per iteration through eliminating unnecessary cluster head messages. 3) In HEED*, a node with higher amount of remaining energy is considered more eligible candidate for election as a cluster head. Besides, each node computes a communication cost, and a regular (non cluster head) node elects the cluster head with the least communication cost to connect to. Simulation results show that the protocol outperforms HEED protocol in terms of network lifetime.**

Keywords-sensor networks; clustering; network lifetime; energy efficiency; distributed algorithms;

I. INTRODUCTION

Wireless Sensor Network (WSN) is composed of a large number of sensor nodes, which are densely deployed either inside the phenomenon or very close to it [1]. WSNs are basically data gathering networks in which data are highly correlated and the end user needs a high level description of the environment sensed by the nodes [2]. The requirements of these networks are ease of deployment, long system lifetime, and low-latency data transfers. The main task of a sensor node in a sensor field is to detect events, perform quick local data processing, and then to transmit the data [1]. As mentioned in [3] and [4], nodes have typically low mobility and are limited in capabilities, energy supply and bandwidth. The sensor network should perform for as long as possible. On the other hand, battery recharging may be inconvenient or impossible. Therefore, all aspects of the sensor node, from the hardware to the protocols, must be designed to be extremely energy efficient [5].

In direct communication WSN, the sensor nodes directly transmit their sensing data to the Base Station (BS) without any coordination between the two. However, in Cluster-based WSNs, the network is divided into clusters. Each sensor node exchanges its information only with its cluster head (CH), which transmits the aggregated information to the BS. Aggregation and fusion of sensor node data at the CHs cause a significant reduction in the amount of data sent to the BS and so results in saving both energy and bandwidth resources. On the other hand, clustering is particularly crucial for scaling the network to hundreds or thousands of nodes [6]. In many applications, cluster organization is a natural way to group spatially close sensor nodes in order to exploit the correlation and eliminate the redundancy that often shows up in the sensor readings [7]. However, these benefits, compared to those of the direct communication WSN, result in extra overhead due to the cluster formation's message exchanges.

This paper proposes an improvement to the HEED clustering protocol [8], called HEED*, to improve network lifetime by reducing the number of cluster head elections. It uses a battery depletion ratio to determine when new cluster head elections should occur, providing a balanced distribution of load on nodes in the network. The protocol is mostly similar to HEED. The first primary difference is that at each iteration, instead of automatically performing a cluster head election, it only performs a new local election if the battery level of a CH node has dropped by a certain percent. That CH node sends *elec-msg* to its member. In this way, all the nodes in that cluster perform CH election again to find new CH node. Therefore, the CH election is performed locally i.e. all the nodes in the network will not participate in the CH election. The second primary difference is that HEED* diminishes the messaging overhead, by integrating communication cost value in CH messages. The third primary difference is the cluster head election algorithm, which leads to nodes with the highest remaining battery levels being favored for selection. Simulation results demonstrate that the benefits of HEED* over HEED protocol, in terms of improving network lifetime and energy savings are noticeable.

The rest of the paper is organized as follows: Section II introduces the related works. Section III describes the problem statement. A new energy efficient clustering scheme is outlined in Section IV. Section V presents the simulation results by comparing network lifetime, and the number of CH elections with HEED protocol. Finally, the conclusion is presented.

II. Related Works

Research on clustering in WSNs has focused on developing centralized and distributed protocols to compute sets of CHs and to form clusters. Centralized approaches (e.g. [9]) are rather inefficient in the case of large scale networks since collecting the entire amount of necessary information at the central control (BS) is both time and energy consuming. Distributed approaches are more efficient for large scale networks. In these approaches, a node decides to become a CH or to join a cluster based on the information obtained solely from neighbors within its proximity. Several distributed clustering protocols have been proposed in literature (e.g. [5], [6], [8], [10], and [11]). As mentioned in [12], most of these protocols are either iterative or probabilistic. In probabilistic protocols (e.g. [10], [11], and [13]), the decision to become CH is reached probabilistically. On the other hand, in iterative protocols (e.g. [8]), the nodes perform an iterative process to decide whether to become a CH or not. From another point of view, clustering protocols are considered as being static and dynamic. In static clustering, the clusters are permanently formed (e.g. [6], [9]), while in dynamic clustering (like [5], [7], [8], [10], and [11]), protocol operation is divided into rounds; clusters are formed for a round and then should be formed again for the next round. In doing so, extra overhead is imposed on the system. The following presents a review of famous distributed dynamic clustering protocol (HEED) used in this study's simulation.

HEED protocol [8], which is an iterative clustering protocol that uses the energy of nodes and a communication cost to elect CHs. In HEED, each node computes the communication cost depending on whether variable power levels, applied for intra-cluster communication, are permissible or not. If the power level is fixed for all of the nodes, then the communication cost can be proportional to (i) node degree, if load distribution between CHs is required, or (ii) 1/node degree, if producing dense clusters is required. The authors defined $AMRP$ the average of the minimum power levels needed by all M nodes within the cluster range to access the CH u, i.e. $AMRP(u) = \frac{\sum_{i=1}^{M} MinPwr(i)}{M}$. If variable power levels are admissible, $AMRP$ is used as the cost function. In this approach, every regular node elects the least communication cost CH in order to join it. On the other hand, the CHs send the aggregated data to the BS in a multi-hop fashion.

III. Problem Statement

The following properties are assumed in regard to the sensor network being studied:

- The nodes can use power control to change the amount of transmit power. Also, each node performs signal processing functions and has the computational power to support different MAC protocols.
- The nodes have ideal sensing capabilities. In other words, the quality of the node's sensing does not change inside the cluster range regardless of the distance from the node.
- The sensor nodes are quasi-stationary. This is typical for sensor network applications.
- Nodes are not equipped with GPS-capable antennae, meaning they are location-unaware.
- In addition to being of equal importance, the capabilities of nodes, such as processing and communicating, are similar.
- Nodes are energy constrained and are left unattended after deployment. Therefore, battery recharge is not possible.
- Each node has an initial amount of energy, E_{max}, and the BS is not limited in terms of energy, memory, and computational power.
- Distance can be measured based on the wireless radio signal power.
- Links are symmetric, i.e., two nodes v_1 and v_2 can communicate using the same transmission power.

T_{CP} and T_{NO} are defined as follows:

- T_{CP} (the period of the clustering process) is the time interval used by the clustering protocol to cluster the network.
- T_{NO} (the network operation interval) is the time between the end of a T_{CP} interval and the start of the subsequent T_{CP} interval.

In order to reduce overhead, it must be ensured that $T_{NO} \gg T_{CP}$.

Suppose the above assumptions hold and that n nodes are distributed in a field. In consideration of energy saving issues, the goal is to identify a collection of CHs which cover the entire area. Each node v_i, where $1 \leq i \leq n$, must be mapped to exactly one cluster c_j, where $1 \leq j \leq n_c$, and n_c is the number of clusters ($n_c \leq n$). L_i shall denote the lifetime of node i. The network lifetime will be defined as follows:

- F is the time elapsed until the First Node Dies (FND). In other words, $F = min\ (L_1, L_2, \ldots, L_n)$.
- H is the time elapsed until only one Half of Nodes remain Alive (HNA). Therefore, $H = median\ (L_1, L_2, \ldots, L_n)$.
- L is the time elapsed until the Last Node Dies (LND) or, $L = max\ (L_1, L_2, \ldots, L_n)$.

The major purpose here is to maximize F, H, and L, which requires using the energy of all nodes uniformly. A node must have the ability to directly communicate with its CH and by a single-hop fashion. A CH has two critical responsibilities: (1) intra-cluster coordination and (2) inter-cluster communication. Multi-hop routing is used for inter-cluster communication. CHs can utilize a routing protocol to compute inter-cluster paths for communicating in a multi-hop fashion with the BS, e.g. the power-aware routing protocol in [14].

IV. THE HEED* PROTOCOL

In this section, the HEED* protocol and its pseudo code are illustrated. The operation of HEED* is divided into rounds and each round is comprised of two phases:

- The setup phase, which includes CH election and consequently cluster formation. In addition, in this phase, every CH coordinates with its members to send sensing data during the following phase.
- The steady state phase, which is broken up into TDMA frames. During each frame, every regular node, at the time of its respective time slot, sends sensing data to its CH (similar to [5]). At the end of each TDMA frame, every CH forwards the aggregated data to the BS through the CHs.

This protocol has the following characteristics which resemble [8]'s:

- The steady state phase is similar.
- The clustering process is iterative.
- A chosen CH advertises only to its neighbors.
- The clustering procedure ends in O(1) iterations.
- Each node can directly communicate with its CH.
- During the clustering process, a node can be either a *tentative_CH* or a *final_CH* or it can be covered.
- At the end of the setup phase, CHs form a network backbone, such that packets are routed from the CHs to the BS in a multi-hop fashion over CHs.

A. Local Clustering

A novelty of HEED* is that it decreases overhead by performing the setup phase on demand instead of in each round, as shown in Fig. 1. To do so, when the clustering process finishes (at the end of each setup phase), every CH saves its residual energy in its memory, for example in its E_{CH} variable. During the steady state phase, whenever a CH finds that its $E_{residual}$ falls below αE_{CH} (α is a constant number and $0 < \alpha < 1$), the CH sends a message named *elec-msg* to its members at the end of current TDMA frame. By sending the *elec-msg*, the CH node informs its members to hold the setup phase at the beginning of the upcoming round. Therefore, CH election and consequently cluster formation are performed locally rather than globally. This means that all the nodes in the network are not required to participate in CH election process. Therefore, during T_{cp} interval, some nodes may perform CH election while other nodes wait until this duration finishes. As a result, the overhead created by consecutive setup phases is tremendously reduced, because clustering is done not only sporadically but also locally. Consequently, there is a decrease in the energy dissipation of nodes and an increase in network lifetime.

E_{CH}: Residual energy of the CH node in the end of each setup phase.
α: A floating point number which is constant and $0 \leq \alpha \leq 1$.
$V = \{v | v\ is\ a\ CH\ node\ and\ E_{residual}(v) > 0\}$,
$W = \{w | w\ is\ a\ regular\ node\ and\ E_{residual}(w) > 0\}$.
During the steady state phase:
$\forall x \in V$: **IF** $E_{residual}(x) < \alpha E_{CH}(x)$ **THEN**
The node x informs its cluster members by sending *elec-msg* at the end of current frame.
$\forall y \in W$: **IF** $the\ elec-msg\ is\ received$ **AND** $y \in x$ **THEN**
The node y becomes ready to hold the clustering for the upcoming round.

Figure 1. Pseudo code of local clustering.

B. Clustering Process

The clustering process of HEED* is divided into three phases:

- ***Initialization Phase***: In the beginning of this phase, neighbor information must be updated. Afterwards, each node may compute its cost independently. This cost will not be broadcasted to neighbors as it is exchangeable through *CH_msg* messages. As previously mentioned, note that updating the neighbor information and computing costs are not required every time clustering is triggered. At first, the protocol sets an initial percentage of CHs among all sensor nodes, C_{prob}. Here, C_{prob} is set to 0.05. Each sensor node sets its own probability of becoming a CH, CH_{prob}, (the same as [8]):
$$CH_{prob} = MAX\ (C_{prob} * \left({E_{residual}}/{E_{max}}\right), p_{min}).$$
In the above, $E_{residual}$ is the current energy of the sensor node and E_{max} is the maximum energy corresponding to a fully charged battery. CH_{prob} is not allowed to fall below a certain threshold p_{min}, which has been selected to be inversely proportional to E_{max}. The threshold p_{min} limits the number of iterations of the *Main processing phase* to O(1). See the *Initialization phase* in Fig. 2.
- ***Main Processing Phase***: S_{CH} shall be defined as follows: $S_{CH} = S_{final\_CH} \cup S_{tentative\_CH}$, where $S_{final\_CH} =$

{All *final_CH*s from iteration 1 to iteration i} and $S_{tentative\_CH}$ = {All *tentative_CH*s from iteration 1 to iteration i}. At the beginning of this repetitive phase, a node with higher amount of residual energy has a greater chance, CH_{prob}, of becoming a *tentative_CH*. If a node becomes a *tentative_CH*, it broadcasts a message announcing its new status to all other sensor nodes within its cluster range. In the next iterations, if this particular node has the most remaining energy among the *tentative_CH*s in its proximity and its CH_{prob} reaches one, it will become a *final_CH* and shall broadcast a *final_CH* message within its cluster range. On the other hand, if a node receives a *final_CH* message, it can no longer be elected as a CH. Therefore, in the following phase, it must choose to connect to one of the *final_CH*s in its cluster radius, based on the cost of that *final_CH*. Finally, each sensor node doubles its CH_{prob} value and continues to the next iteration of the repetitive phase. When its $CH_{previous}$ reaches one, the sensor node stops executing this phase. As a result, the nodes with a higher amount of remaining energy will finish the execution of HEED* earlier than nodes with a lower amount of remaining energy. This permits the nodes with lower amount of energy to join their clusters. Observe that, in this phase, each *tentative_CH* (or *final_CH*) node can send a *CH_msg* only once. See the *Main processing phase* in Fig. 2.

- ***Finalization Phase***: During this phase, each sensor node makes a final decision about its status. If the node is not a *final_CH* and has received at least one *final_CH* message, it will elect the *final_CH* with the least cost to join it. If a node completes the clustering process and has not yet received any *final_CH* message, it will find itself uncovered and so shall introduce itself as the *final_CH*. See the *Finalization phase* in Fig. 2.

```
Phase I. Initialization
    Snbr ← {v: v lies within my cluster range}
    v computes the communication cost for itself.
    CHprob ← MAX (Cprob × (Eresidual/Emax), pmin)
    is_final_CH ← FALSE

Phase II. Main Processing
  REPEAT
    IF ((SCH ← {v: v is a tentative_CH or final_CH}) ≠ Ø)
      IF ((Sfinal-CH ← {v: v is a final_CH}) = Ø)
        IF (NodeID = BIGGEST_ENERGY (SCH) AND CHprob = 1)
            cluster_head_msg (NodeID, final_CH, cost, Eresidual)
            is_final_CH ← TRUE
    ELSE IF (CHprob = 1)
      cluster_head_msg (NodeID, final_CH, cost, Eresidual)
      is_final_CH ← TRUE
    ELSE IF (Random(0,1) < CHprob)
      cluster_head_msg (NodeID, tentative_CH, cost, Eresidual)
    CHprevious ← CHprob
    CHprob ← MIN (CHprob× 2, 1)
  UNTIL CHprevious = 1

Phase III. Finalization
    IF (is_final_CH = FALSE)
      IF ((Sfinal-CH ← {v: v is a final_CH}) ≠ Ø)
        my_cluster_head ← LEAST_COST (Sfinal-CH)
        JOIN_CLUSTER (my_cluster_head, NodeID)
      ELSE
        cluster_head_msg (NodeID, final_CH, cost, Eresidual)
```

Figure 2. Pseudo code of HEED* protocol.

TABLE I. PARAMETER SETTINGS

PARAMETER	VALUE
ϵ_{fs}	10 pJ/bit/m^2
ϵ_{mp}	0.0013 pJ/bit/m^4
E_{elec}	50 nJ/bit
E_{DA}	5 nJ/bit/signal
Idle power	13.5 mW
Sleep power	15 µW
Threshold distance (d_0)	75 m
Initial energy per node	2 J
Round time	20 sec
Round	5 frame
Data packet size	100 byte
Control packet size	25 byte
p_{min}	10^{-4}

V. SIMULATION RESULTS

In this section, a comparison between the simulation results in HEED* and HEED protocols is performed via Matlab software. The following assumptions and system parameters (similar to [5]) are used:

- All nodes have the equal amount of initial energy (2 J).
- The total number of nodes in the system is N = 100.
- The BS is located outside the supervised area at the coordinate (50, 175).
- The energy required for data aggregation is set as E_{DA}=5nJ/bit/signal and CHs perform ideal data aggregation.
- We assume a simple model for energy dissipation of the radio hardware in which the receiver dissipates energy to run the radio electronics and the transmitter dissipates energy to run the power amplifier and the radio electronics, similar to [5].
- Sensor nodes are randomly dispersed in a square field (between (0, 0) and (100, 100) m)
- Nodes always have data to send to the end user and nodes situated in each other's close proximity have correlated data
- The rest of the parameters are listed in TABLE I.

In the following, we show α-diagram in Fig. 3 and then the number of alive nodes, the network lifetime, total residual energy of the network, and number of elections from the beginning of network operation up to the specified round are discussed in Fig. 4-6.

As it is mentioned before, CH election is done on demand and also locally. In order to obtain the best α, we ran HEED* with different communication costs and different values of α. In Fig. 3, α differs from 0.1 to 0.9 and each plot demonstrates the average of 5 executions for the specified cost when the first node dies (FND). Figures 4-6 are plotted using α=0.9 as it approximately results in better network lifetime.

Fig. 4 shows the total number of nodes that remain alive over the simulation round with node degree, 1/node degree and AMRP communication costs. As it is shown in this figure, HEED* protocol considerably increases network lifetime more than HEED. The reasons are that:

- Number of elections for determining CHs reduced considerably because this protocol does not perform CH election in all rounds. Besides, the clustering process is done locally. Therefore, the extra overhead of determining CHs decrease.
- Number of *cluster_head_msg* messages exchanged in HEED* is less than HEED.
- In HEED*, costs are sent on demand and in the form of *cluster_head_msg* messages.
- In HEED*, CHs are elected only based on residual energy of node and regular nodes join to the clusters based on communication cost.

To better compare network lifetime for the mentioned communication costs, we can summarize the time the first node dies (FND), only one half of the nodes which are alive (HNA) and the last node dies (LND) in the Fig. 5. It is obvious that HEED* is better in any definition of network lifetime than HEED.

Fig. 6(a) shows total residual energy of all nodes in the network with the node degree communication cost. This figure also shows that HEED* protocol performs better than HEED protocol. The reason is that total remaining energy in the entire network decreases more slowly. The plots of other two costs are similar, so they are removed.

By performing CH election on demand we can compare number of CH elections after each round for HEED and HEED* protocols in Fig. 6(b). As it is shown in this figure for the node degree communication cost, number of CH elections in HEED* is much less than this value in HEED. The plots for other two costs are approximately the same as this cost, so they are removed.

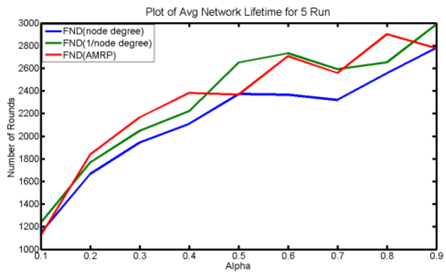
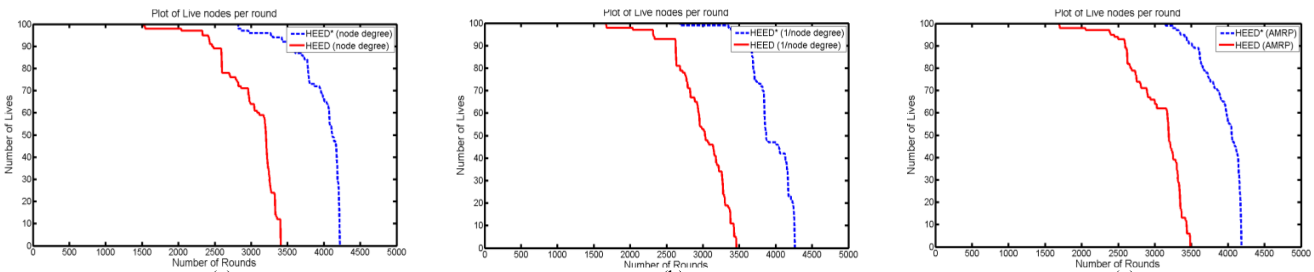
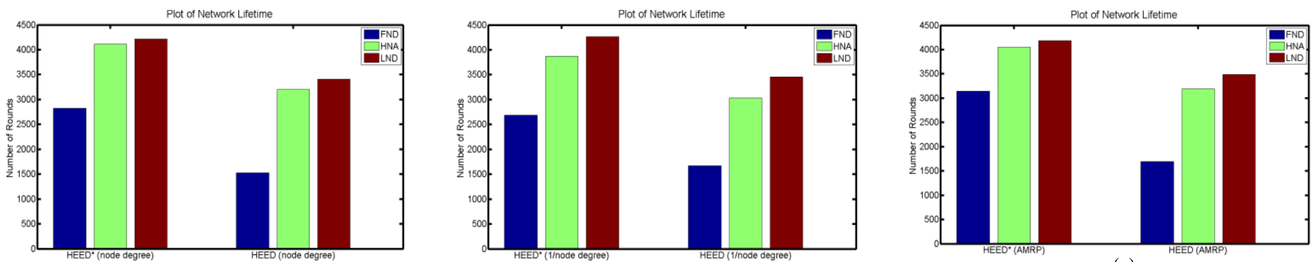


Figure 3. The α-diagram in HEED* with three communication costs.



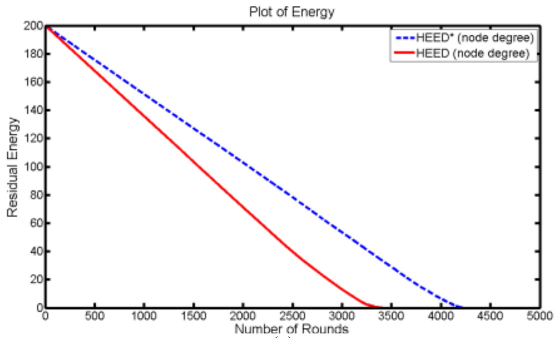
(a) (b) (c)

Figure 4. The network lifetime in HEED* with (a) node degree, (b) 1/node degree, (c) AMRP communication costs.

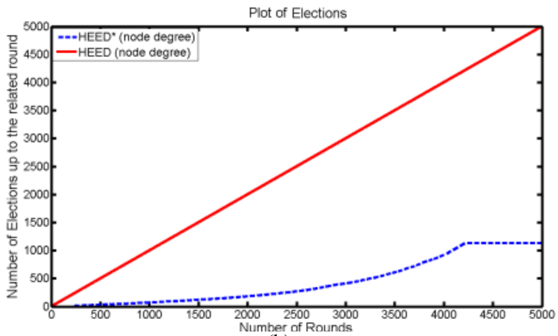


(a) (b) (c)

Figure 5. The comparison of different definitions of network lifetime in HEED* with HEED, (a) node degree, (b) 1/node degree, (c) AMRP communication costs.



(a)



(b)

Figure 6. (a) Total remaining energy in HEED* (b) The number of CH elections up to each round, with node degree communication cost.

VI. Conclusion

This paper proposes HEED*, an extension to HEED which is a popular distributed algorithm for energy-efficient clustering in WSN. Main differences with the original contribution, rely on that HEED* does not trigger reorganization at each round, but only when one of the elected CHs goes below a certain fraction of the energy level it had once elected. On the other hand, only that CH node and its cluster members participate in clustering process, i.e., clustering is performed locally. HEED* also diminishes the messaging overhead, by integrating communication cost value in CH messages. Besides, it takes the residual battery level of a node into account for holding a new round of election and forming clusters. The simulation results demonstrate that HEED* outperforms the HEED protocol in terms of network lifetime.

References

[1] I. Akiyldiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[2] W. B. Heinzelman, "Application-specific protocol architectures for wireless networks," *Ph.D. Thesis, Massachusetts Institute of Technology*, June 2000.

[3] O. Younis and S. Fahamy, "Distributed clustering in ad-hoc sensor networks: A hybrid, energy-efficient approach," in *Proc. IEEE INFOCOM*, Mar. 2004, vol. 1, pp. 629–640.

[4] K. Akkaya, M. Younis, "A survey on routing protocols for wireless sensor networks," *Elsevier Journal of Ad Hoc Networks*, vol. 3, no. 3, 325–349, 2005.

[5] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans. Wireless Commun.*, vol. 1, no. 4, pp. 660–670, Oct. 2002.

[6] X. Zhu, L. Shen, and T. P. Yum, "Hausdorff Clustering and Minimum Energy Routing for Wireless Sensor Networks," *IEEE Trans. Vehicular Technology*, vol. 58, no. 2, 2009.

[7] S. Soro, W. Heinzelman, "Cluster head election techniques for coverage preservation in wireless sensor networks", Ad Hoc Networks, vol. 7, no. 5, pp. 955–972, July 2009.

[8] O. Younis, S. Fahmy, "HEED: A Hybrid, Energy-Efficient, Distributed clustering approach for Ad Hoc sensor networks," *IEEE Transactions on Mobile Computing,* vol. 3, no. 4, pp. 366–379, 2004.

[9] T. Kaur, J. Baek," A Strategic Deployment and Cluster-Header Selection for Wireless Sensor Networks," *IEEE Trans. on Consumer Electronics*, Vol. 55, No. 4, 2009.

[10] M. C. M. Thein, T. Thein, "An Energy Efficient Cluster-Head Selection for Wireless Sensor Networks," *International Conference on Intelligent Systems, Modeling and Simulation*, 287-291, 2010.

[11] S. Bandyopadhyay and E. J. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks," in *Proc. IEEE INFOCOM*, Apr. 2003, vol. 3, pp. 1713–1723.

[12] O. Younis, M. Krunz, and S. Ramasubramanian, "Node Clustering in Wireless Sensor Networks: Recent Developments and Deployment Challenges*," IEEE Network,* May/June 2006.

[13] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," *Proc. of the Hawaii International Conference on System Sciences*, pp. 1–10, January 2000.

[14] S. Singh, M.Woo, and C. Raghavendra, "Power-aware routing in mobile ad hoc networks," in *Proc. 4th Annual ACM/IEEE Int. Conf. Mobile Computing Networking (MobiCom)*, Oct. 1998.