This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2010 proceedings

A Distributed Topology Management Protocol for Wireless Sensor Networks

Hend Alqamzi
Electrical Engineering Department
The UAE University
Al Ain, United Arab Emirates

Tiffany Jing Li
Department of Electrical and Computer Engineering
Lehigh University
Bethlehem, PA 18015

Abstract— **A new clustering approach, termed Distributed Energy Efficient clustering Protocol (DEEP), is proposed for wireless sensor networks. Using a non-iterative cluster formation operation, the protocol spends an extremely low overhead energy compared to the existing protocols and terminates faster than the energy-expensive iterative processes. The distributed head election algorithm guarantees that the periodically-elected leaders have the highest residual energy among their members in each data reporting cycle, effectively balancing the energy consumption among sensors. The DEEP also accounts for sensor limitations as well as practical concerns such as wireless collision that has not been considered by the existing clustering protocols. It intelligently exploits the overhearing ability of sensors to alleviate the loss of clustering control packets due to collisions. The proposed protocol is very effective in forming well-distributed clusters that ensure the required load balancing, the connectivity of clusters, and the minimum data communication energy during the data collection stage. In addition, the DEEP does not make any advanced assumptions about the required number of clusters, the network density, the energy consumption pattern of sensors or their clock synchronization and capabilities. For a thorough evaluation, we have compared the performance of DEEP with an existing clustering protocol. The simulation results show the effectiveness of DEEP in reducing the energy expenditure besides assuring other desirable features. We have also examined its performance in prolonging the network lifetime in the context of a practical routing protocol.**

I. Introduction

Network clustering divides the sensors in the network field into groups or clusters. Each cluster has a leader and a number of associated members that use minimum transmission power to communicate with their head. Clustering is an effective topology management approach that can reduce energy consumption and hence prolong the overall network life span. Since data transmission is an energy expensive task, clustering allows the majority of sensors (i.e. cluster members) to communicate data over smaller distances (i.e. to their nearby leaders), leaving only a few nodes (i.e. cluster heads) communicate with a distant base station or a relatively far next-hop cluster head. Thus, clustering significantly reduces the consumed energy and improves the network scalability. It also provides a natural avenue for efficient data aggregation (e.g., to compute the maximum detected concentration of chemical agents) and faster and simpler (hierarchical) packet routing. Moreover, clustering enables spectrum reuse and channel recycle, saving nodes' energy and time to compete to capture the media in order to deliver data directly to the base station.

Major design challenges of network clustering protocols include that the protocol needs to be completely distributed such that each node makes its decision locally based on its exchanged information with neighbors that are within its radio range. Also, the clustering process must terminate very fast using minimal processing time and minimal message exchange. In addition, clustering must account for collision of control messages. Thus, it must be capable of handling lost heads' announcements and members' join messages. Furthermore, clustering should consider non-synchronized nodes in the field. Moreover, the protocol should produce cluster heads that are well-distributed and connected (for multi-hop clustered networks, where cluster heads collaborate with each other to deliver their data to the base station).

Many clustering protocols have been proposed for wireless sensor networks with different assumptions and objectives. For example, in [1], the number of clusters are assigned a priori using a centralized approach, whereas the Low Energy Adaptive Hierarchy (LEACH) [2] uses a distributed approach that limits the number of clusters based on analytical derivations. Also, some protocols require time synchronization among sensors, which make them suitable only for applications using small number of sensors. Reference [3] assumes non-homogeneous sensors and considers the power control along with clustering. In addition, the cluster head election rule is based on the number of neighbors [4], [5], the node IDs [6], the node that first sends a packet [7], or a random number, which is compared with a probability function including the residual energy [2], [8] . Although the clustering process of the Hybrid Energy Efficient Distributed clustering (HEED) [8], and LEACH [2] uses an energy-based probability function, LEACH [2] expends higher energy compared to HEED [8] as each node requires the residual energies of *all* sensors in the network to make its clustering decision. However, since HEED [8] selects the final clusters after a number of iterations, each involves exchange of head announcements and members registration messages, it leads to a high clustering overhead, and hence consumes considerable energy.

In this paper, we present a new distributed clustering

978-1-4244-6404-3/10/$26.00 ©2010 IEEE

approach known as Distributed Energy Efficient clustering Protocol (DEEP), which differs from the existing approaches in three aspects: First, it guarantees that the elected leader has the highest residual energy among its members based on its actual residual energy level rather than a non-guaranteed probability function. Second, it accounts for some practical concerns in the wireless media such as collisions that are usually ignored in most of the existing clustering protocols. Third, it does not make any advanced assumptions about the target number of clusters, the network density, the energy consumption pattern of the sensors or their synchronization and capabilities.

The rest of the paper is organized as follows. Section II presents the operations of the DEEP protocol. Section III evaluates and compares DEEP with another clustering protocol through simulations. Finally, Section IV concludes this paper.

II. The DEEP Protocol

Load balancing is one of the primary elements to prolong the network lifetime. Thus, the clustering process must be triggered periodically to select a different set of high energy leaders every cycle.

At the beginning, each node exchanges its ID and residual energy with its local neighbors that are within its communication range R. Once all sensors get the required energy information about their close neighbors, each node compares its current remaining energy level with the residual energies of its neighbors. The nodes that have the highest energy levels among their neighbors announce that they are the heads for that neighborhood. To avoid the collisions of two advertising heads located at neighboring clusters, each cluster head backs off for a period of time that is inversely proportional to its residual energy before its announcement. In other words, the head that has the highest energy waits for shorter period compared to another head with relatively lower energy. Next, heads wait for a time of T_w to receive the registration (join) messages from its potential members. The nodes with lower residual energies in each neighborhood wait for time T_w to receive all head advertisements. Once they receive the announcements, they must also back off to avoid collisions, and then send registration messages to the heads that are closest to them. Note that if the nodes are not aware of the location information, they can measure the signal strength of the received advertisements to select the nearest head. If the time has elapsed without hearing from any surrounding cluster head due to packet collisions or lack of heads around, the node then listens to the join messages sent by neighbors, and join the head that is selected by its nearest neighbor. If such information is still not available, then there is a possibility that there are two nodes in the neighborhood having the same remaining energy (highest) and neither has advertised. If both nodes become heads, the resulting cluster heads at the end will not be distributed properly (fairly) in the field, besides possible collisions caused by their simultaneous announcements. Therefore, to limit the number of advertisements made by such heads (having same energies) and ensure that each cluster range has only one head, the node that has the highest ID number among the two becomes a head whereas the other waits to join it. If a node still didn't hear announcements or registration messages, it declares itself a head. This means that this node is isolated with no neighbors around [1].

In order to minimize clusters with only a single node (clusters with head only and no members), each head checks its registered members, if no one has joined its cluster and it has a nearby head (within a range of $2R$) with higher residual energy, it joins that head. Otherwise, it remains a single node cluster. For load balancing, the clustering process must be triggered periodically to select a different set of high energy leaders every cycle. It is worth noting that the protocol is capable of handling the expected collisions of both types of packets: the advertisement packet as well as the join packet. In spite of the use of a back-off scheme, collision may occur with a very low probability. If the advertisement packet is lost on its way to one of the members, such member can either join another cluster head if it is within the range R of other clusters and has received their announcements, or find its way back to this cluster head by listening to the join packets sent by neighboring nodes. Similarly, if the registration packet is lost on its way to a head that has already received other join messages in the neighborhood, then the head will come to know about this member at the data collection phase. However, if this head didn't receive any registration message, it might form a single node cluster or join another head. Thus, in the former case, the member will send its data to the head and form a non-single cluster, whereas in the latter case, the member will overhear the join message of its head to another cluster. In this case, the member can choose to join the new selected cluster if it is within its range, join another cluster, or choose to form a single node cluster based on the available information.

III. Performance Evaluation

A. Simulation Model

A set of 100 sensors are deployed randomly in $100m \times 100m$ square area. Sensors are provided with an energy of $200 Joules$ within a standard deviation of 2%. They have the ability to adjust their transmission power to reach different distances. The performance of DEEP is evaluated through 100 simulation experiments. In each experiment, sensors are assigned different locations and residual energies. The communication range (cluster range R) is limited to $30m$, and this is how far heads can send clustering advertisements in a single-hop communication. We compare DEEP to HEED [8], which is one of the latest and efficient clustering protocols for sensor networks in terms of energy and number of iterations to form a cluster.

[1] This case is very rare in a dense sensor network.

TABLE I

SIMULATION PARAMETERS

Parameter	Value
Radio electronics energy E_{elec}	$50nJ/bit$
Transmit amplifier energy (free space) ε_{fs}	$10pJ/bit/m^2$
Transmit amplifier energy (two ray)ε_{2r}	$0.0013pJ/bit/m^4$
Broadcast packet size	$25bytes$
Data packet header	$25bytes$
Data packet size	$100bytes$
Combine energy E_c	$5nJ/bit/signal$
Threshold distance d_{th}	$75m$
Base station at	$(50, 175)m$

HEED nodes generate a random number between 0 and 1. If the generated number is less or equal to the probability CH_{prob}, which is a function of the residual energy of the node, then the node becomes a *tentative* cluster head and sends an announcement message to its neighbors within its cluster range R along with the calculated cost. HEED selects cluster heads based on a primary parameter, which is the residual energy and a secondary parameter that represents the communication cost. Variations of HEED arise from different secondary parameters. First, a node may join the cluster head that has the minimum node degree cost (minimum number of members) if the goal is to distribute the cluster head load. Second, a node may choose to join the leader with maximum degree to create dense clusters. Third, a node may select the cluster that has the Average Minimum Reach ability Power (AMRP) to create cluster heads that consume minimum communication energy. AMRP is defined by HEED as the mean of the minimum power levels required by all M nodes within the cluster range to reach the cluster head, i.e. $AMRP = \sum_{i=1}^{M} \frac{MinPwr_i}{M}$. Since the results of all HEED versions are identical as shown in [8] at the selected cluster range R of $30m$, we only simulate the minimum degree cost.

We use an energy model similar to the energy model used by HEED [8] to facilitate the comparison. Since nodes executing the DEEP protocol require the residual energy information of their neighbors that are within their cluster range R, the dissipated energy in such exchanged messages is included in the clustering energy consumption for a fair comparison with HEED. Note that nodes running the HEED protocol do not need the residual energies of their neighbors. The simulation parameters are listed in Table I.

We use a simple routing protocol that follows the clustering phase in each cycle (round). Each cluster head collects a data packet from each member, combines them, and then forwards the aggregated messages to the base station (single-hop communication). This process is repeated by heads five times in each round. Once a new round starts, new cluster heads are elected (head role is rotational).

B. Performance Analysis

In the radio model used by [8], the energy consumed by a transmit amplifier is proportional to the square distance between the transmitter and the receiver, d^2, whereas the energy consumed in a data transmission across longer distances (such as from a cluster head to the base station) is proportional to d^4. Using this model, the energy spent in transmitting an l-bit message across a long distance, d, is

$$E_{Tx} = lE_{elec} + l\varepsilon_{2r}d^4, \tag{1}$$

where E_{elec} is the radio electronics energy and ε_{2r} is the transmit amplifier energy for the two-ray model. Likewise, the energy expended in sending an l-bit message across a short distance is given by:

$$E_{Tx} = lE_{elec} + l\varepsilon_{fs}d^2, \tag{2}$$

where ε_{fs} is the transmit amplifier energy for the free-space model. The energy consumed in receiving an l-bit message is:

$$E_{Rx} = lE_{elec}. \tag{3}$$

During the data transmission phase, member nodes transmit the data messages to their head, and each cluster head transmits a single aggregated message to a remote base station in each TDMA frame. Assuming uniform energy consumption, the energy dissipated by a cluster head is

$$\begin{aligned} E_{CH} &= l_{data}E_{elec}(\frac{N}{K}-1) + l_{data}E_c(\frac{N}{K}-1) \quad (4) \\ &+ l_{data}E_{elec} + l_{data}\varepsilon_{2r}d_B^4, \end{aligned}$$

where N is the total number of nodes, K is the number of cluster heads, l_{data} is the number of bits in one data packet, E_c is the data aggregation energy, and d_B is the separating distance between the cluster head and the base station. This assumes that each cluster has the same number of members. The energy consumed by each member in a cluster is estimated by

$$E_{non-CH} = l_{data}E_{elec} + l_{data}\varepsilon_{fs}d_C^2, \tag{5}$$

where d_C is the distance from the member node to the cluster head. In an area of $A \times A$, and assuming uniform distribution of clusters, each cluster occupies approximately $\frac{A^2}{K}$ or πR^2 with a node distribution of $\rho(x, y)$. Using similar derivation as [2], the average squared distance from the nodes to their cluster head, which is assumed to be located at the center of this area, is given by:

$$E[d_C^2] = \iint (x^2 + y^2)\rho(x, y)dxdy = \iint r^2\rho(r, \theta)rdrd\theta. \tag{6}$$

Assuming that the area is a circle with radius R and $\rho(r, \theta)$ is constant for r and θ, the above equation becomes

$$E[d_C^2] = \rho\int_{\theta=0}^{2\pi}\int_{r=0}^{R} r^3 drd\theta = \frac{\rho\pi R^4}{2}. \tag{7}$$

For a uniform density of nodes in a cluster area, $\rho = \frac{1}{\pi R^2}$, the average distance is simplified to

$$E[d_C^2] = \frac{R^2}{2}, \tag{8}$$

Using the relation

$$\frac{A^2}{K} = \pi R^2. \tag{9}$$

The average distance between member nodes and their cluster head is related to the number of clusters K by the following:

$$E[d_C^2] = \frac{A^2}{2\pi K}. \tag{10}$$

Therefore, the average energy consumed by a member node in a cluster is

$$E_{non-CH} = l_{data}E_{elec} + l_{data}\varepsilon_{fs}\frac{R^2}{2}. \quad (11)$$

The energy dissipated in a single cluster during one TDMA frame is:

$$E_{cluster} = E_{CH} + (\frac{N}{K} - 1)E_{non-CH} \quad (12)$$

and the total energy consumed per frame is:

$$\begin{aligned} E_{total} &= KE_{cluster} \\ &= Nl_{data}(2E_{elec} + E_c + \varepsilon_{fs}\frac{R^2}{2}) \\ &+ \; Kl_{data}(\varepsilon_{2r}d_B^4 - E_c - E_{elec} - \varepsilon_{fs}\frac{R^2}{2}). \quad (13) \end{aligned}$$

Note that the total energy dissipated per frame increases with the number of clusters, the radius of clusters, and the distance to the base station. In order to find the optimal number of clusters that corresponds to the minimum dissipated energy in each frame, the equation (9) is used to substitute the cluster range R. Thus, the above equation becomes

$$\begin{aligned} E_{total} &= KE_{cluster} \\ &= Nl_{data}(2E_{elec} + E_c + \varepsilon_{fs}\frac{A^2}{2\pi K}) \\ &+ \; Kl_{data}(\varepsilon_{2r}d_B^4 - E_c - E_{elec} - \varepsilon_{fs}\frac{A^2}{2\pi K}). \quad (14) \end{aligned}$$

Then, the derivative of the total energy consumed, E_{total}, with respect to K is set to zero. So, the optimal number of clusters is

$$K_{optimal} = \sqrt{\frac{N\varepsilon_{fs}\frac{A^2}{2\pi}}{d_B^4\varepsilon_{2r} - E_c - E_{elec}}} \quad (15)$$

A 3-D plot of equation (14) in Fig. 1 shows that the energy consumed per TDMA frame decreases with the decreasing cluster heads and their communication distance with the base station. For $N = 100$ nodes, $A = 100m$, and using the parameters in Table I, the optimal number of clusters ranges from 1 to 40 clusters for cluster head distances to the base station between $75m$ and $185m$. However, the result obtained in this analysis assumes a uniform energy consumption of nodes in the network field as well as a uniform cluster size. Practically, and in our simulation, sensors are deployed randomly, and hence the average distances between nodes are not known a priori. Therefore, the number of members in a cluster varies from one cluster to another, and the distances between the periodically elected cluster heads and the base station may also vary. Hence, the energy consumption of cluster heads is not uniform.

C. Simulation Results

1) Cluster Head Features: Fig. 2 shows that the average number of cluster heads selected by DEEP is far lower than the average number of HEED clusters. This is expected as HEED tends to favor compact cluster sizes through its iterative clustering process, leading to more clusters in the network field. However, DEEP algorithm is executed individually by each node to check its eligibility of being a leader for a group of members that is included in its circular area which is bounded by its cluster range. Since DEEP is not an iteration-based process, each member individually and independently has only one chance to select the nearest cluster head without having the knowledge about the leader's current size. In a multi-hop cluster network where cluster heads cooperate with each other to deliver the data to a remote base station, having few clusters in the field results in few number of hops to the destination and hence, faster movement of data. On the other hand, in a single-hop cluster network where each head communicates directly with the base station, larger number of clusters means more competing heads to access the media, causing higher interference, delay, and collisions compared to smaller number of leaders (clusters) using CSMA protocol.

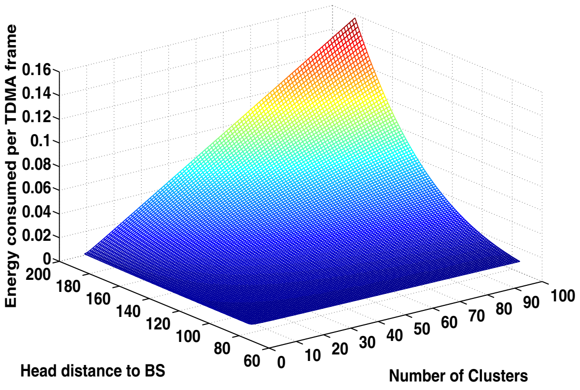

Fig. 1. Energy consumed for varying number of clusters and heads' distances to base station

It is worth noting that HEED cannot guarantee cluster heads with high residual energies. This is due to its use of a secondary parameter to decide the final cluster heads that satisfy the lowest communication cost. In addition, HEED uses the residual energies of the nodes to probabilistically select an initial set of cluster heads. Then, as heads and members proceed in their iterations, members may select heads which have lower energies but incur minimum costs due to cluster density or neighbor proximity. On the other hand, our DEEP guarantees that the elected leaders have the highest residual energy in their neighborhood except the case of isolated nodes (i.e. do not have neighbors), which seldom exists in dense sensor networks. This property is desirable as it provides load balancing in a *periodic election* of cluster heads.

2) Cluster Features: Fig. 3 represents the standard deviation of the total number of nodes in each cluster. The standard deviation measures the fairness of the distribution of nodes among all clusters in the network. A Smaller standard deviation indicates more balanced cluster sizes and more balanced data gathering load for cluster heads. As expected, HEED produces more balanced clusters than our DEEP through the iteration process. However, it is encouraging to see that DEEP also produces fairly uniform

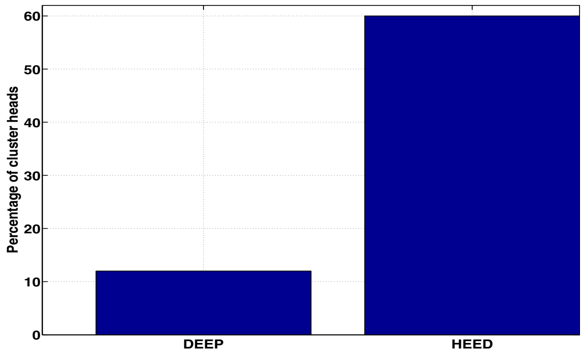

Fig. 2. Percentage of cluster heads

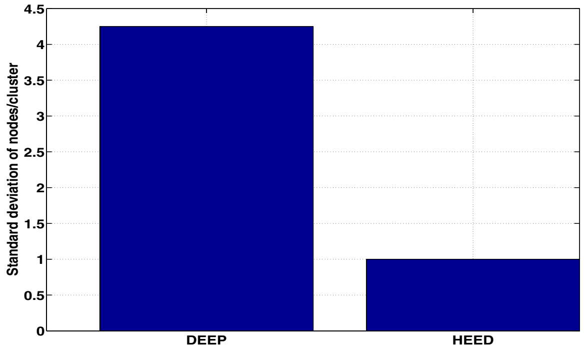

Fig. 3. Standard deviation of number of nodes per cluster

cluster sizes even without iterative adjustment.

3) Clustering Energy: Clustering protocols can be used to construct energy-efficient hierarchies for sensor applications that run unattended reporting continuously to an observer. In addition, the clustering approach is essential for applications that require in-network data aggregation. Therefore, the most appealing features of clustering protocols for such applications are the low overhead energy of the clustering process and the high residual energies of the elected cluster heads. Such characteristics achieve the longevity of the network life by minimizing the energy consumption and distributing the cluster head load equally among the nodes in the network.

Fig. 4 compares the clustering energy dissipated by both protocols. The clustering energy is measured as the energy spent by the network nodes in the clustering process as a fraction of the total energy consumed in the network, which involves the data forwarding energy. Although our DEEP propagates the residual energies in each neighborhood prior to clustering the network, it expends very low energy compared to HEED. This significant difference in the clustering overhead energy consumption is attributed to the energy expensive iteration-based process used by HEED that causes a large number of exchanged messages among heads and members.

IV. Conclusion

In this paper, we have introduced an energy-efficient distributed clustering protocol for wireless sensor networks known as DEEP. This protocol utilizes a different approach that obviates energy expensive iterations. Thus, it achieves a great reduction in overhead energy. It also guarantees that almost all the periodically elected cluster leaders exhibit the highest residual energies in their respective neighborhoods unless they are isolated without neighbors. In addition, DEEP cleverly exploits the overhearing nature of nodes in a wireless communication to alleviate the negative impact of message collisions due to wireless media. Moreover, it doesn't make any assumptions about the nodes' density, capability, or significance in the field.

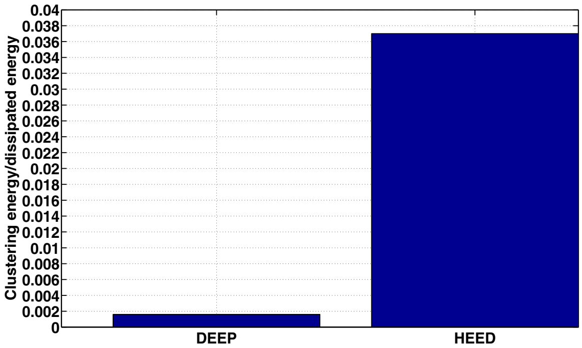

Fig. 4. Ratio of energy spent in clustering to total dissipated energy

Simulation results show that DEEP consumes very low energy compared to HEED, and produces a reasonable number of clusters with heads of high residual energies. DEEP is attractive to dense sensor applications that require scalability and/or in need of in-network data aggregation. Moreover, DEEP can be conveniently operated with various types of sensor routing protocols to prolong the network lifetime.

References

[1] S. Lindsey, C.Raghavendra, and k. Sivalingam. Data gathering in sensor networks using energy delay metric. In *Proceeding of International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, April 2001.

[2] W. R. Heinzelman, A. P.Chandrakasan, and H. Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications*, 1(4):660–670, October 2002.

[3] V. Kawadia and P. R. Kumar. Power control and clustering in ad hoc networks. In *Proceedings of IEEE INFOCOM.*, April 2003.

[4] M. Chatterjee, S. Das, and D. Turgut. Wca: A weighted clustering algorithm for mobile ad hoc networks. *Journal of Cluster Computing (Special Issue on Mobile Ad hoc Networks)*, 5:193–204, April 2002.

[5] H. Chan and A. Perrig. ACE: An emergent algorithm for highly uniform cluster formation. In *Proceedings of the First European Workshop on Sensor Networks (EWSN)*, January 2004.

[6] C. R. Lin and M. Gerla. Adaptive clustering for mobile wireless networks. *IEEE Journal of Selected Areas in Communications*, 15(7):1265–1275, 1997.

[7] M. Gerla, T. Kwon, and G. Pei. On demand routing in large ad hoc wireless networks with passive clustering. In *Proceedings of IEEE WCNC*, September 2000.

[8] O. Younis and S. Fahmy. HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Transactions on Mobile Computing*, 3(4):366–379, Oct-Dec 2004.