

DESC: Distributed Energy Efficient Scheme to Cluster Wireless Sensor Networks

Peyman Neamatollahi¹, Hoda Taheri¹, Mahmoud Naghibzadeh²,
and Mohammad Hossein Yaghmaee²

¹ Dept. Computer Engineering, Islamic Azad University, Mashhad branch,
Young Researchers Club, Iran

{neamatollahi.peyman,h.taheri.mshd}@gmail.com

² Dept. Computer Engineering, Ferdowsi University of Mashhad, Iran
naghibzadeh@um.ac.ir

³ Dept. Computer Engineering, Islamic Azad University, Mashhad branch, Iran
yaghmaee@ieee.org

Abstract. Organizing sensor networks into clustered architectures is an effective approach for load balancing, and prolonging the network lifetime. This paper proposes a Distributed Energy Efficient Scheme to Cluster Wireless Sensor Networks (DESC). This protocol achieves energy efficiency using two techniques: on demand clustering and a multi-criteria cluster formation. Whenever a cluster head consumes a prespecified part of its energy, it indirectly informs other nodes to hold cluster head elections for the upcoming round. Therefore, clustering is performed sporadically (in contrast to performing it every round). In DESC, a node with higher amount of remaining energy is considered a more eligible candidate for election as a cluster head. Besides, each node computes a multi-criteria cost, and a regular (non-cluster head) node elects the cluster head with the most cost to connect to. Simulation results show that the protocol outperforms HEED and LEACH protocols in terms of network lifetime and energy savings.

Keywords: Sensor networks, clustering, network lifetime, energy efficient protocols, distributed algorithms.

1 Introduction

Wireless sensor networks (WSNs) provide reliable monitoring from very long distances [1]. 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 [2]. As mentioned in [3] and [4], nodes have typically low mobility and are limited in capabilities, energy supply and bandwidth. Therefore, all aspects of the sensor node, from the hardware to the protocols, must be designed to be extremely energy efficient.

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. In this way, clustering achieves a significant improvement in terms of energy consumption. 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. [5]) 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. [1], [3], [6], [7], and [8]). As mentioned in [9], most of these protocols are either iterative or probabilistic. In probabilistic protocols (e.g. [7], [8], and [10]), the decision to become CH is reached probabilistically. On the other hand, in iterative protocols (e.g. [3]), 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. [5], [6]), while in dynamic clustering (like [1], [3], [7], [8], 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.

This paper presents a distributed dynamic clustering protocol named DESC. The protocol is distributed as each node has a partial view of the network and can, therefore, make its own decision as to whether become a CH or join a cluster based on local information. Clustering is dynamic as the clusters are not formed a priori nor are fixed during the network lifetime. In this protocol, each node employs an iterative process to decide its status. For each node, this repetition phase finishes when the node either elects itself as a CH or finds a CH to join. Compared to famous protocols, HEED [3] and LEACH [1], the goal of DESC is to achieve significant energy saving results in extending network lifetime. This proposed clustering approach does not make any assumptions regarding the distribution of the nodes or node capabilities, e.g., location-awareness. The protocol only assumes that sensor nodes can vary their transmission power. Compared to many previously presented protocols (e.g. [1] and [3]), DESC decreases both the overhead caused by the clustering's message complexity and the number of CH elections. Clustering is performed when at least one CH has consumed a significant part of its residual energy. With the assistance of its transferred data packet, the CH node informs the BS of its status. Then, the BS scatters special synchronization pulses in a multi-hop fashion throughout the area. Upon receiving these pulses, each node prepares to perform clustering for the upcoming round. In the protocol, CH selection is based on cost and remaining energy. Therefore, each node must compute a multi-criteria cost. The cost is proportional to node degree and node centrality. The regular nodes elect the most cost CH in their respective close proximity in order to connect to. Simulation results demonstrate that the benefits of DESC over HEED and LEACH protocols, in terms of improving network lifetime and energy savings are noticeable.

The rest of the paper is organized as follows: Section 2 describes the network model and clustering problem. A new energy efficient clustering scheme is outlined in Section 3. Section 4 presents the simulation results by comparing energy consumption, network lifetime, and the number of CH elections with other well known algorithms. Finally, the conclusion is presented.

2 Preliminaries

2.1 Network Model

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.
- The sensor nodes are quasi-stationary.
- Nodes are not equipped with GPS-capable antennae.
- Nodes are energy constrained and are left unattended after deployment.
- Distance can be measured based on the wireless radio signal 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 $T_{NO} \gg T_{CP}$. Note that, in contrast with other dynamic clustering protocols that perform clustering in each round, DESC clusters the nodes on demand rather than at each round. Therefore, it is possible that some rounds do not include T_{CP} ; instead T_{NO} is extended during these rounds. As a result, the length of the T_{CP} interval is fixed but the length of the T_{NO} interval varies throughout the network lifetime. In DESC, node readings are periodically reported to the BS. Therefore, a TDMA frame is created in each CH to remove interference within a cluster. The protocol uses special synchronization pulses to alert the sensor nodes that clustering will be triggered in the beginning of the next round. These pulses are propagated in a multi-hop fashion (as approach presented in [12]).

2.2 The Clustering Problem

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, $F = \min(L_1, L_2, \dots, L_n)$.
- H is the time elapsed until only one Half of Nodes remain Alive (HNA). Therefore, $H = \text{median}(L_1, L_2, \dots, L_n)$.
- L is the time elapsed until the Last Node Dies (LND) or, $L = \max(L_1, L_2, \dots, L_n)$.

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 [13].

Note that, in the clustering process, every iteration takes time, t_c . Period t_c should be long enough to receive messages from any neighbor within the cluster radius. Because the nodes are quasi-stationary, neighbor discovery is not required every time clustering is performed. Hence, the neighbor set of every node does not vary very frequently. In multi-hop networks, the nodes automatically update their neighbor sets by periodically sending and receiving heartbeat messages.

3 The DESC Protocol

In this section, the DESC protocol and its pseudo code are illustrated. The operation of DESC 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 [1]). 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 [3]'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.

3.1 On Demand Clustering

A novelty of DESC is that it decreases overhead by performing the setup phase on demand instead of in each round. 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 sets a prespecified bit in a data packet which is ready to be sent to the BS in the current TDMA frame. Upon receiving the forwarded CH data packet (sent in a multi-hop fashion), the BS informs the sensors to hold the setup phase at the beginning of the upcoming round. This could be achieved by having the BS send out, in a multi-hop fashion, specific synchronization pulses to nodes. These pulses are quickly dispersed throughout the network. When every node receives a pulse, it prepares itself to perform clustering. Therefore, CH election and consequently cluster formation are performed on demand. As a result, the overhead created by consecutive setup phases is tremendously reduced. Consequently, there is a decrease in the energy dissipation of nodes and an increase in network lifetime.

3.2 Multi-Criteria Cost

In the DESC protocol, each node computes a cost according to an equation (2). The nodes with the greater cost have a higher priority of being elected as CH. On the other hand, every regular node joins the CH which has the highest cost in its vicinity.

$$\text{cost}_v = \sum_{j=1}^{\text{num of neighbors}} \frac{1}{\text{dist}^2(v,j)} \quad (1)$$

The combination of two main features, node centrality (the centrality of a node among its neighbors) and node degree (the number of neighbors), are considered when computing the cost:

- The lower the value of the neighbors' distances from node v , the higher the value of the cost_v . This means that the location of node v is more central among its neighbors and is, thus, more suitable for election as a CH. The node centrality feature in CH election has some advantages for consideration:
 - Member nodes can communicate with their respective CH node by consuming less energy.
 - Radio wave interferences between CHs are reduced.
- The greater the number of node v 's neighbors, the higher the value of cost_v . In other words, node v is more suitable for election as a CH and so, dense clusters are constructed in the area. In many applications, dividing the network into dense clusters is an important goal to group spatially close sensor nodes. The aim is to exploit the correlation and eliminate the redundancies that often exist in the sensor readings.

3.3 Clustering Process

The clustering process of DESC 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 [3]):

$$CH_{prob} = \text{MAX} (C_{prob} * \left(\frac{E_{residual}}{E_{max}} \right), p_{min}).$$

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. 1.

- *Main Processing Phase*: S_{CH} shall be defined as follows: $S_{CH} = S_{final_CH} \cup S_{tentative_CH}$, where $S_{final_CH} = \{\text{All final_CHs from iteration 1 to iteration } i\}$ and $S_{tentative_CH} = \{\text{All tentative_CHs from iteration 1 to iteration } i\}$. 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 highest cost among the *tentative_CHs* 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_CHs* 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 DESC 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. 1.

- *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 highest 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. 1.

<p>Phase I. Initialization</p> $S_{nbr} \leftarrow \{v: v \text{ lies within my cluster range}\}$ $cost \leftarrow \sum_{v=1}^{num \text{ of neighbors}} \frac{1}{dist^2(me,v)}$ $CH_{prob} \leftarrow \text{MAX}(C_{prob} \times (E_{residual}/E_{max}), p_{min})$ $is_final_CH \leftarrow \text{FALSE}$ <p>Phase III. Finalization</p> <p>IF ($is_final_CH = \text{FALSE}$)</p> <p>IF ($(S_{final_CH} \leftarrow \{v: v \text{ is a } final_CH\}) \neq \emptyset$)</p> <p> $my_CH \leftarrow \text{MOST_COST}(S_{final_CH})$</p> <p> $\text{JOIN_CLUSTER}(my_CH, NodeID)$</p> <p>ELSE</p> <p> $CH_msg(NodeID, final_CH, cost)$</p>	<p>Phase II. Main Processing</p> <p>REPEAT</p> <p>IF ($(S_{CH} \leftarrow \{v: v \text{ is a } tentative_CH \text{ or } final_CH\}) \neq \emptyset$)</p> <p>IF ($(S_{final_CH} \leftarrow \{v: v \text{ is a } final_CH\}) = \emptyset$)</p> <p> IF ($NodeID = \text{MOST_COST}(S_{CH})$ AND $CH_{prob} = 1$)</p> <p> $CH_msg(NodeID, final_CH, cost)$</p> <p> $is_final_CH \leftarrow \text{TRUE}$</p> <p>ELSE IF ($CH_{prob} = 1$)</p> <p> $CH_msg(NodeID, final_CH, cost)$</p> <p> $is_final_CH \leftarrow \text{TRUE}$</p> <p>ELSE IF ($\text{Random}(0,1) < CH_{prob}$)</p> <p> $CH_msg(NodeID, tentative_CH, cost)$</p> <p> $CH_{previous} \leftarrow CH_{prob}$</p> <p> $CH_{prob} \leftarrow \text{MIN}(CH_{prob} \times 2, 1)$</p> <p>UNTIL $CH_{previous} = 1$</p>
--	---

Fig. 1. Pseudo code of DESC protocol

4 Simulation Results

In this section, a comparison between the simulation results in DESC, LEACH and HEED protocols is performed via Matlab software. The following assumptions and system parameters (similar to [1]) are used:

- The nodes always have data to send to the end user and the nodes situated in close proximity to others have correlated data.
- The CHs perform ideal data aggregation.

- A simple model for the energy dissipation of radio hardware is assumed, in which the receiver dissipates energy to run the radio electronics and the transmitter dissipates energy to run the power amplifier and radio electronics. Thus, for transmitting a k -bit message over distance d , the radio expends:

$$E_{Tx}(k, d) = \begin{cases} kE_{elec} + k\epsilon_{fs}d^2, & d < d_0 \\ kE_{elec} + k\epsilon_{mp}d^4, & d \geq d_0 \end{cases}$$

For receiving this message, the radio expends:

$$E_{Rx}(k) = E_{Rx-elec}(k) = kE_{elec}.$$

- The rest of the parameters are listed in Table 1.

Table 1. 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 μ W
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}

In the simulation experiment, node sizes of 100, 200, 300, and 400 are chosen. The simulations for two scenarios are conducted:

- Scenario 1. A network with nodes deployed randomly over an area of size 100×100 m². The BS is located at the coordinate (50, 175).
- Scenario 2. A network with nodes deployed randomly over an area of size 200×200 m². The BS is located at the coordinate (100, 275).

For each of these scenarios with different numbers of nodes, the cluster range is chosen such that approximately five percentages of nodes are being elected as CH. In the following, the proper value of α is discussed, and then energy consumption, network lifetime, and the number of CH elections are evaluated.

4.1 Setting α Variable

As mentioned before, CH election (clustering execution) is performed on demand. When a CH consumes a prespecified part of its energy (i.e. $E_{residual} \leq \alpha E_{CH}$), it

indirectly informs the other nodes that clustering must be performed for the upcoming round. In order to obtain the proper α , DESC was run for the two scenarios described above (here, initial energy per node is assumed to be 0.2 Joule). In Fig. 2, α differs from 0 to 1 and each plot demonstrates the average of three executions for 100, 200, 300, and 400 number of nodes, until the first node dies (FND). Also plotted is an average of the four mentioned plots.

When α is equal to zero, no clustering is performed during the network lifetime (i.e. the static clustering approach that considers fixed CHs). In homogenous networks in which nodes have similar capabilities and the same amount of energy, CHs quickly deplete respective energies. Therefore, when a CH dies, the respective cluster becomes dysfunctional. When α equals one, this signifies that clustering is performed in each round, similar to the LEACH and HEED protocols. Considering the plot, which is the average of the four different numbers of nodes, figures 3-9 are plotted using $\alpha = 0.7$ as it approximately results in better network lifetime.

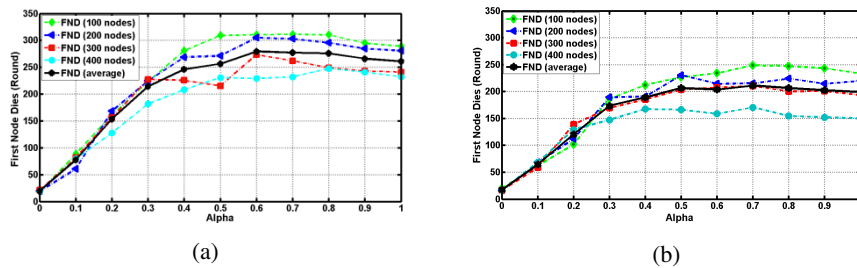


Fig. 2. α diagram in DESC for different numbers of nodes and their average. (a) Scenario 1 and (b) Scenario 2.

4.2 Energy Consumption Comparisons

In this subsection, the energy dissipation to cluster the WSN and the energy consumption to transmit the sensed data to the BS are evaluated. Note that in the figures belonging to this subsection, the vertical axis does not have an equal range of data because of the high energy dissipation of LEACH in Scenario 2. In Fig. 3, the average energy dissipation of protocols clustering the WSN per election is evaluated. The DESC protocol performs better because its clustering's message complexity is low and, similar to HEED, its messages are sent in the range of a cluster radius. The LEACH protocol reports the highest dissipation of energy because it does not apply a cluster radius to limit the range of message distribution. Therefore, the value of clustering's energy dissipation for LEACH depends on the network diameter. This evaluation is depicted in the difference found among the values of the LEACH protocol in Fig. 3 (a) and Fig. 3 (b).

Fig. 4 demonstrates the average energy dissipation per round of protocols clustering the WSN. The average for DESC is much lower than that of two other protocols.

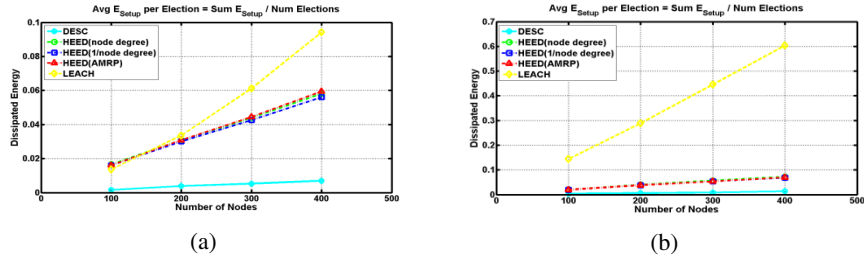


Fig. 3. Average energy dissipated to cluster WSN per election. (a) Scenario 1 and (b) Scenario 2.

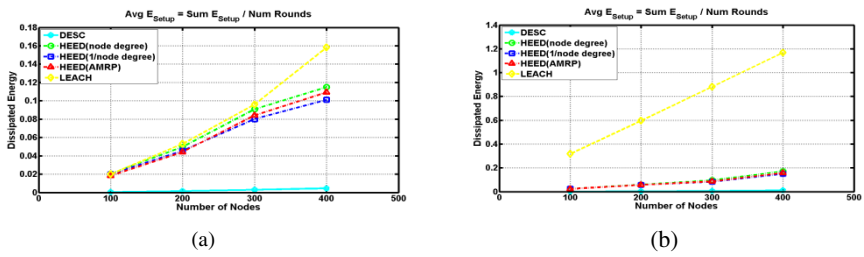


Fig. 4. Average energy dissipated to cluster WSN per round. (a) Scenario 1 and (b) Scenario 2.

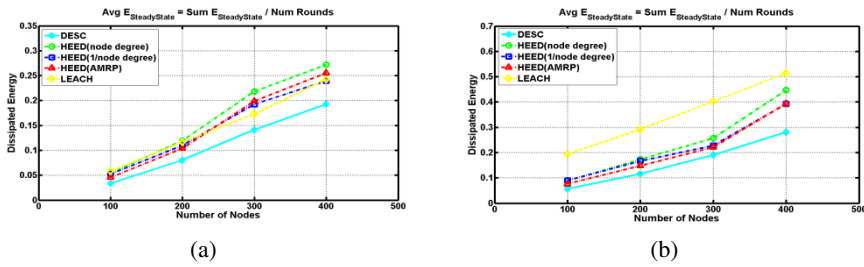


Fig. 5. Average energy consumption for data transmission in WSN per round. (a) Scenario 1 and (b) Scenario 2.

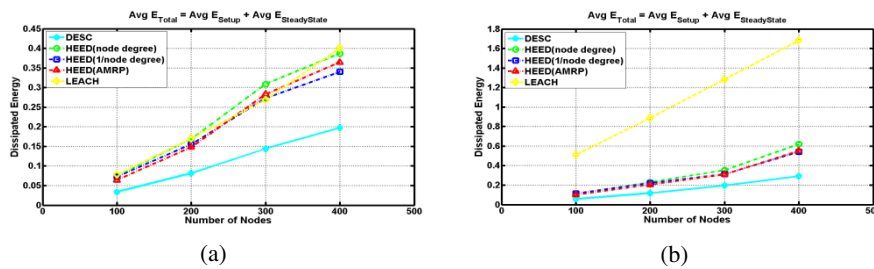


Fig. 6. Total average energy consumption in WSN per round. (a) Scenario 1 and (b) Scenario 2.

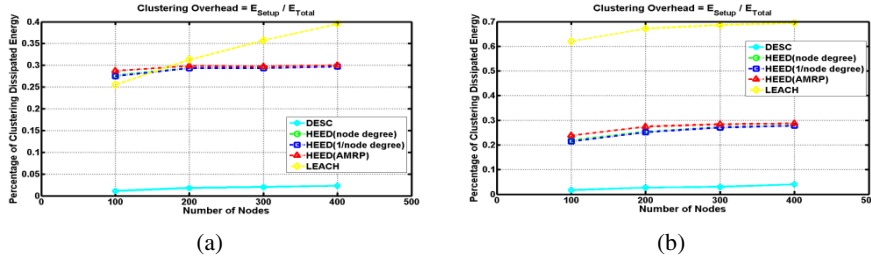


Fig. 7. Clustering overhead. (a) Scenario 1 and (b) Scenario 2.

This difference is caused by 1) low clustering message complexity and 2) not performing the CH election in each round. The average energy consumption for data transmission is depicted in Fig. 5. DESC consumes less energy in the steady state phase because it forms better clusters using a multi-criteria cost compared to the other protocols. HEED uses three different communication costs separately, but here a multi-criteria cost combining the node degree and node centrality is presented. Therefore, DESC forms better clusters than HEED and, of course, better than LEACH. In Fig. 5 (b), LEACH consumes the most energy in intra-cluster communication and also in the transferring of data to the BS. Its intra-cluster communication is not energy efficient because there is no guarantee of a fair distribution of CHs in the field. Therefore, it is possible that some CHs are located far away from their respective members. To sum up Fig. 4 and 5, the total average energy consumption per round for the three protocols is evaluated in Fig. 6. It is noticeable that DESC is an energy efficient protocol regardless of the network diameter.

Fig. 7 illustrates the ratio of energy dissipation when clustering the WSN to the total energy consumption for each protocol. Note that, in both scenarios, DESC achieves a tremendously lower clustering overhead compared to the HEED and LEACH protocols. From the figures provided in this subsection, it can be concluded that DESC is an energy efficient protocol.

4.3 Network Lifetime Comparisons

This paper compares the network lifetime of the discussed protocols for different numbers of nodes with the scenarios presented in Fig. 8. The round of FND, HNA, and LND are summarized in this figure. Because a simple multi-hop routing [13] is implemented to simplify the simulation, the value of FND for nodes 100-400 is in descending order in DESC and HEED protocols (Fig. 8 (a) and (b)). This power-aware routing chooses the minimum energy routes to transfer the data over CHs. Using the minimum energy path during the entire steady state phase will deplete the energy of the CHs on that path. Instead, one of the multiple paths with a certain probability can be used so that the whole network lifetime increases (similar to [14]). Fig. 10 shows that DESC clearly acts better in any definition of network lifetime than the HEED and LEACH protocols without any dependency on network diameter. Consecutively, DESC is a scalable clustering protocol in terms of the number of nodes and the network size.

4.4 Comparison of Number of CH Elections

By not performing the setup phase in all rounds of DESC, the total number of CH elections until LND for these protocols can be compared in Fig. 9. Both the LEACH

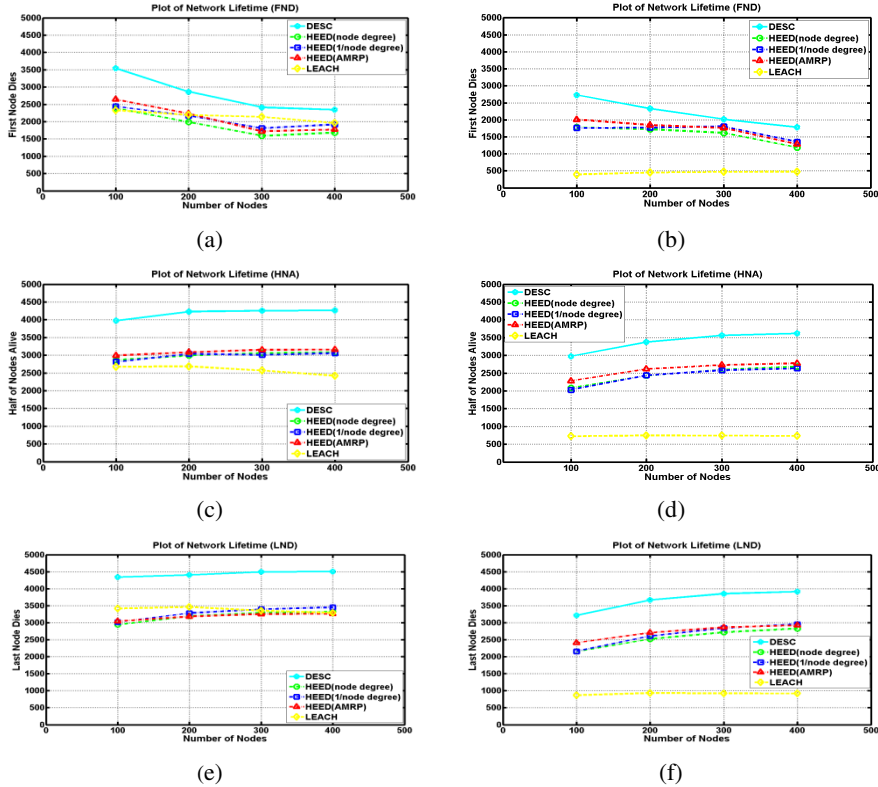


Fig. 8. The network lifetime. (a) FND with Scenario 1, (b) FND with Scenario 2, (c) HNA with Scenario 1, (d) HNA with Scenario 2, (e) LND with Scenario 1, and (f) LND with Scenario 2.

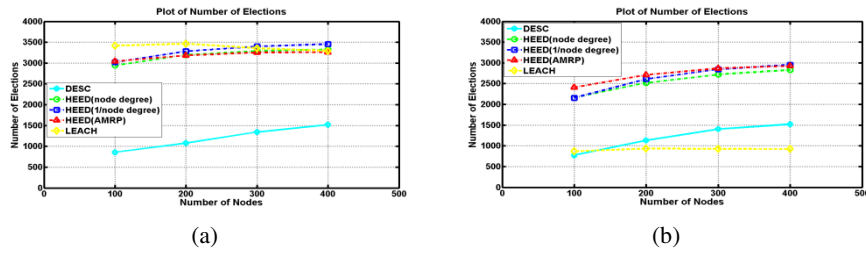


Fig. 9. The number of CH elections in the DESC, HEED, and LEACH protocols. (a) Scenario 1 and (b) Scenario 2.

and HEED protocols perform the setup phase in each round. As shown in this figure, the number of CH elections in DESC is much lower than this value in the other two protocols. As shown in this figure, adding to the number of nodes increases the number of elections. The reason for this is that, by increasing the number of nodes, the value of LND increases (see Fig. 8 (e) and (f)). The number of CH elections in the LEACH protocol decreases in Scenario 2, when compared to Scenario 1. The reason for this is because the value of LND for LEACH in Scenario 2 is much lower than that of Scenario 1.

5 Conclusion

This paper proposes an energy efficient, distributed clustering protocol for WSNs, termed DESC. In this protocol, the clustering's message overhead is low and cluster heads are distributed fairly across the network. Quasi-stationary networks are assumed in which the nodes are location-unaware and have equal significance. In DESC, the CHs are elected based on the residual energy of the node and multi-criteria cost. On the other hand, the regular nodes join the highest cost CH. Also, (after the first setup phase) clustering is not performed until at least one of the CHs consumes a prespecified part of its energy; performing the setup phase in each round imposes much overhead on the network. The efficiency of DESC was evaluated by using Matlab to compare it with the well known protocols, HEED and LEACH. The simulation results demonstrate that DESC outperforms the other two protocols in terms of network lifetime. The main reasons for this are as follows: (1) the clustering process is performed on demand, (2) cluster formation is based on multi-criteria cost, and (3) clustering message complexity is negligible. Although we provided algorithm to cluster homogenous nodes in the area, a future work can be an extension on the protocol to cluster the heterogeneous nodes.

References

1. Heinzelman, W.B., Chandrakasan, A.P., Balakrishnan, H.: An application-specific protocol architecture for wireless micro sensor networks. *IEEE Trans. Wireless Commun.* 1(4), 660–670 (2002)
2. Akiyldiz, I., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks. *IEEE Commun. Mag.* 40(8), 102–114 (2002)
3. Younis, O., Fahmy, S.: HEED: A Hybrid, Energy-Efficient, Distributed clustering approach for Ad Hoc sensor networks. *IEEE Transactions on Mobile Computing* 3(4), 366–379 (2004)
4. Akkaya, K., Younis, M.: A survey on routing protocols for wireless sensor networks. *Elsevier Journal of Ad Hoc Networks* 3(3), 325–349 (2005)
5. Kaur, T., Baek, J.: A Strategic Deployment and Cluster-Header Selection for Wireless Sensor Networks. *IEEE Trans. on Consumer Electronics* 55(4) (2009)
6. Zhu, X., Shen, L., Yum, T.P.: Hausdorff Clustering and Minimum Energy Routing for Wireless Sensor Networks. *IEEE Trans. Vehicular Technology* 58(2) (2009)
7. Thein, M.C.M., Thein, T.: An Energy-Efficient Cluster-Head Selection for Wireless Sensor Networks. In: *International Conference on Intelligent Systems, Modeling and Simulation*, pp. 287–291 (2010)

8. Bandyopadhyay, S., Coyle, E.J.: An energy efficient hierarchical clustering algorithm for wireless sensor networks. In: INFOCOM 2003, vol. 3, pp. 1713–1723 (2003)
9. Younis, O., Krunz, M., Ramasubramanian, S.: Node Clustering in Wireless Sensor Networks: Recent Developments and Deployment Challenges. *IEEE Network* 20(3), 20–25 (2006)
10. Heinzelman, W.B., Chandrakasan, A.P., Balakrishnan, H.: Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In: Proc. of the Hawaii International Conference on System Sciences (2000)
11. Soro, S., Heinzelman, W.: Cluster head election techniques for coverage preservation in wireless sensor networks. *Ad Hoc Networks* 7(5), 955–972 (2009)
12. Greunen, J., Rabaey, J.: Lightweight Time Synchronization for Sensor Networks. In: *WSNA* (2003)
13. Singh, S., Woo, M., Raghavendra, C.: Power-aware routing in mobile ad hoc networks. In: *MobiCom* (1998)
14. Shah, R., Rabaey, J.: Energy aware routing for low energy ad hoc Sensor Networks. In: *WCNC* (2002)