An Energy- And Proximity-based Unequal Clustering Algorithm for Wireless Sensor Networks

M. Mehdi Afsar[1] and Mohamed Younis[2]
[1] Young Researchers and Elite Club, Mashhad Branch, Islamic Azad University, Mashhad, Iran
[2] Dep. of Computer Science and Elec. Eng., Univ. of Maryland, Baltimore County, Baltimore, MD 21250, USA
E-mail: m.afsar@qiau.ac.ir and younis@csee.umbc.edu

***Abstract*—Wireless Sensor Networks (WSNs) are usually constrained energy and bandwidth. Many solutions, like network clustering, have been proposed in order to overcome these limitations. While this solution is deemed efficient, the cluster-heads closer to the base-station would forward more data packets than farther ones, and thus their energy drains at a faster rate. In this paper, we propose an Energy- and Proximity-based Unequal Clustering algorithm (EPUC) to solve this problem. Basically EPUC imposes a condition on the distance among cluster-heads that is adaptively adjusted, so that the inter-cluster-head proximity is smaller as they get closer to the base-station. In addition, the cluster population is set while factoring in the inter-cluster relaying activities in order to balance the load on cluster-heads. We evaluate the performance of EPUC through simulation and confirm its effectiveness of EPUC using network lifetime metrics.**

***Index Terms*—Wireless Sensor Networks; Clustering; Topology Management; Energy-Aware Design.**

I. Introduction

A WSN is composed of many miniaturized sensor nodes that are able to probe their surroundings and transmit the data to an in-situ base-station (BS). Since the nodes are battery-operated and WSNs often operate in inaccessible environments where batteries cannot be replaced, energy-conservation measures have to be employed at the network architecture and operation levels [4]-[16]. Clustering is one of the popular techniques for achieving both scalability and energy efficiency in WSNs. The idea is to form a two-tier network architecture. Nodes are grouped into clusters, each is led by a cluster-dead (CH) that collects data from the cluster members. This constitutes the lower tier. In the second tier an inter-CH topology is formed so that the collected data from the individual clusters are disseminated to the BS.

In the past few years, many clustering algorithms were proposed for WSNs. Most of these algorithms realize that the CH may become a bottleneck since it is usually more loaded than a sensor node, and opt to boost the CH lifetime when designating CHs and forming clusters. Conventional mechanisms for achieving such an objective include selecting CHs based on remaining energy and balancing the size of the clusters [1]. However, published clustering algorithms have mostly overlooked what we refer to as the hot spot problem in the inter-CH topology. Basically, since all data traffic are routed to the BS, CHs that are closer to the BS would forward more data packets than others and thus they deplete energy at a faster rate. If unaddressed, this problem could significantly degrade the network performance since frequent reclustering will be needed in order to prevent all nodes in the vicinity of the BS from running out of energy and keep it reachable. Rotating the role of CH among the cluster nodes will not be an effective solution since the inherent traffic pattern will sure affect all nodes that play the role of CH for clusters close to the BS. Soro *et al.* [14] has shown that this problem can be overcome by decreasing the size of the clusters that are close to the BS in order to lighten the intra-cluster load on the involved CHs.

In this paper, we propose a novel Energy- and Proximity-based Unequal Clustering algorithm (EPUC) to overcome the hot spot problem and extend the network lifetime. EPUC opts to achieve spatial distribution of clusters that even the energy consumption rate of all CHs. To do so, EPUC first divides the area into some virtual tracks around the BS. Nodes located in the same track form clusters with similar sizes. The cluster formation process is distributed. After determining their track, a set of nodes self-elect themselves as CH Candidates (CCHs) based on their energy reserve. Then, unequal cluster sizes for the different tracks are pursued and determined based on two criteria: the distance to the BS and the distance among the CCHs. EPUC is validated through simulation and is shown to outperform competing clustering schemes in the literature in terms of the network lifetime. The parameter settings are further analyzed mathematically to achieve the desired topology.

The rest of this paper is organized as follows. The related work is discussed in section II. The assumed network model is described in section III. We explain the proposed EPUC algorithm in section IV. Section V reports the simulation results. Finally, the paper is concluded in section VI.

II. Related Work

In this section, we set EPUC apart from published unequal clustering approaches. The main factor considered by prior work on unequal clustering is the distance between the CHs and the BS so that larger clusters are formed as the distance to the BS grows. The Unequal Clustering Scheme (UCS) proposed by Soro and W. Heinzelman [14] is among the early work that opts to form clusters whose sizes are proportional to their distance to the BS. However, UCS assumes the CHs are resource-rich nodes in terms of the energy supply and that

978-1-4799-3780-6/14/$31.00 ©2014 IEEE

they are to be located at the center of each cluster. Obviously, these assumptions may not apply in all setups and make the practicality of UCS questionable.

Meanwhile, the energy-efficient unequal clustering mechanism (UCR) [10][7] pursues a randomized CH election process. Each node generates a random probability 'u' between 0 and 1. If u exceeds a predefined threshold, the node becomes a tentative CH. Each node computes its competition range that is proportional to the distance to the BS. A set of candidates with different competition ranges are picked in order to achieve unequal clustering. If a tentative CH finds its residual energy greater than all other contenders (i.e., those with the same competition range), it would become a final CH. EAUCF [6][5] pursues a similar approach which focuses on assigning different competition ranges to the tentative CHs. EAUCF uses the residual energy and the distance to the BS of the sensor nodes to vary the competition ranges, and employ fuzzy-logic is the underlying optimization engine to set the competition radius for maximal overall network lifetime. EADUC [18] also elects CHs based on the average residual energy of nodes in the vicinity of the CH candidates, and uses variable competition ranges to construct unequal clusters. Uneven competition ranges are determined using a weighted function of the residual energy and distance of the nodes to the BS. No guidelines have been provided on how to determine the weighting factors. EPUC includes the distance condition to control the size of the clusters and not only the cluster density, as UCR, EAUCF and EADUC do. This enables fine-grained balancing of the load and further extends the network lifetime. We compare the performance of EPUC to that of UCR in section V.

In HYbrid Multi-hop routiNg (HYMN) [13][2] a mix of hierarchical and flat architectures is used in order to solve the hot spot problem. Since the closer nodes to the BS should forward more data than others, HYMN considers a small area close to the BS as the Sink Connectivity Area (SCA) which uses the flat routing method proposed in [15] for forwarding the received data from farther nodes to the BS. In the area out of the SCA, the network is clustered by M-LEACH [12]. In the zone out of the SCA, HYMN selects CHs randomly; this makes the approach unreliable because the nodes with even low residual energy have equal chance to get elected as the CH. Also, the exact location of the boundary between the hierarchical and flat architectures is not obvious. More importantly, how combining two different types of architectures makes the practicality issue of the approach questionable. In EPUC, these problems are fixed by unequal clustering across the entire network and considering the residual energy of nodes in the CH election. We also compare the performance of EPUC with HYMN in section V.

Location-based Unequal Clustering Algorithm (LUCA) [9] assumes that the nodes have information about their distance to the BS through a GPS receiver, and the nodes can self-organize into unequal clusters using their location information and distance to the BS. The reliance on GPS diminishes the practicality of LUCA and increases its energy overhead. Meanwhile the Energy-Balancing unequal Clustering Approach for Gradient-based routing (EBCAG) [11] approach opts to control the distribution of CHs based on the hop-count rather than physical proximity to the BS. Each node maintains a gradient value that is defined as its minimum hop-count to the BS. Candidate CHs are randomly picked for each gradient value. Data gathered from the cluster members should follow the direction of descending gradient to reach the BS. The size of a cluster is also determined by the gradient value of its CH based on an estimate of the number of member nodes relaying their data from clusters with a higher gradient value. However, EBCAG suffers two problems: (1) EBCAG selects tentative CHs randomly without considering their energy reserve; (2) it is assumed that all nodes of gradient i are reachable to all nodes with gradient $i+1$, which is not generally the case.

III. System Model

We assume a WSN of N nodes that are randomly dispersed in a square field $M \times M$. The spatial distribution of nodes is assumed to follow uniform random distribution such that:

$$N = \lambda |A| = \lambda M^2, \tag{1}$$

where $|A|$ is the area of the field and λ is the density. The BS is assumed to be located just outside the area of interest, at a distance no more than R_t from area boundary, where R_t is the sensor's transmission range. Unlike a sensor node, the BS is capable of long haul transmission and all sensor nodes are assumed to be within its transmission range. Nonetheless, there might be some nodes out of the BS's transmission range. Since EPUC is distributed, even isolated nodes can perform their operations by obtaining required information from their neighbors. Both the sensor nodes and the BS are stationary. Each node can use different power levels in order to communicate with other nodes. All sensor nodes are homogeneous and also location unaware. In-network data aggregation is employed to eliminate redundant data packets. At network setup clusters are formed, and then the network starts data collection. Note that CH re-election is performed at the beginning of each round in order to further increase the network lifetime. Some notations and message description are presented in Table I.

The model for energy dissipation is derived from the first radio model proposed in [8]. Although for multi-hop transmission most communications are preformed in *free space* model, some transmissions might be performed in *multi-path* model whose distance is greater than threshold distance $d_0 = \sqrt{\varepsilon_{fs}/\varepsilon_{mp}}$, where ε_{fs} and ε_{mp} are the amplifier energy of free space and multi-path models. Accordingly, the energy needed to transmit a l-bit packet to distance d is,

$$E_t = \begin{cases} l(E_{el} + \varepsilon_{fs} d^2) & d \leq d_0, \\ l(E_{el} + \varepsilon_{mp} d^4) & d > d_0, \end{cases} \tag{2}$$

where E_{el} is the electronics energy. Also, to receive a l-bit packet a node consumes,

$$E_r = l E_{el}. \tag{3}$$

TABLE I
NOTATIONS AND DESCRIPTION OF THE MESSAGES

Notation	Meaning
A	The area covered by the network
N	The number of sensor nodes
M	The side length of the network
CH	A Cluster Head node
CCH	A Candidate Cluster Head node
R_{comp}	The range of competition for CCH election
R_c	The cluster range for cluster formation
R_{ini}	The initial cluster range of the last track
R_t	The transmission range of a sensor node
ID	A node identifier
$d_{(i,j)}$	The distance between nodes i and j
d_{max}/d_{min}	The maximum/minimum distance to the BS
d_T	The distance between the tracks
S_T	The sensor node track number
D_{thr}	The threshold distance for CH election
T	The number of tracks
T_n	The n-th track
N_{cl_n}	The number of nodes within a cluster located in track n
ρ	The aggregation coefficient
λ	The node density
k	The number of the CHs
k_n	The number of the CHs within track n
E_{ch_n}	The consumed energy by a CH located in track n
E_{res}	The residual energy of the node
E_{max}	The maximum energy of the node
t_w	The waiting time for hearing from CCH in the vicinity
Control Messages	**Description**
CCH-Inf	Include (ID, E_{res}, T_n)
CCH-ADV	Include (ID, T_n)
CH-ADV	Include (ID, T_n)
Join-Req	Include (ID, T_n)
hello	Include (d_{max}, d_{min}, R_{ini})
Route	Include (ID, $d_{(ID,BS)}$)

IV. DETAILED EPUC ALGORITHM

In this section, we explain the proposed EPUC algorithm. At first, the CH election procedure algorithm is described. Then, cluster formation and data transmission are explained.

A. CH Election

The main aim of CH election is to form small clusters in the proximity of the BS, and relatively larger clusters as getting further from the BS. Unlike prior work, the distance metric factors in two aspects: the distance between CHs and the BS and the distance among neighboring CHs. EPUC first designates some nodes with the highest residual energy as candidate CHs. Then it applies the distance condition to select the final set of CHs. This subsection describes the CH election process. The setting of the various parameters will be explained in detail in subsection B.

First EPUC models the area as tracks around the BS as follow. The BS broadcasts a *hello* message within its maximum transmission range (d_{max}) so that all nodes can receive it. The *hello* message contains d_{max}, d_{min}, and R_{ini}. The value of d_{min} can reflect the distance to the closest 1-hop neighbor the BS. The setting of R_{ini} will be discussed later in the section. Each node receives the message, calculates its distance to the BS based on RSSI, and infers its track number. The tracks are defined so that a connected inter-CH routing topology is formed. Basically, a CH in track T_n transmits the aggregated data to a CH in track T_{n-1}. Thus, in the worst case the distance between these CHs will be $2d_T$. Therefore, in order to ensure the connectivity, d_T should be

$$d_T = \frac{R_t}{2}. \tag{4}$$

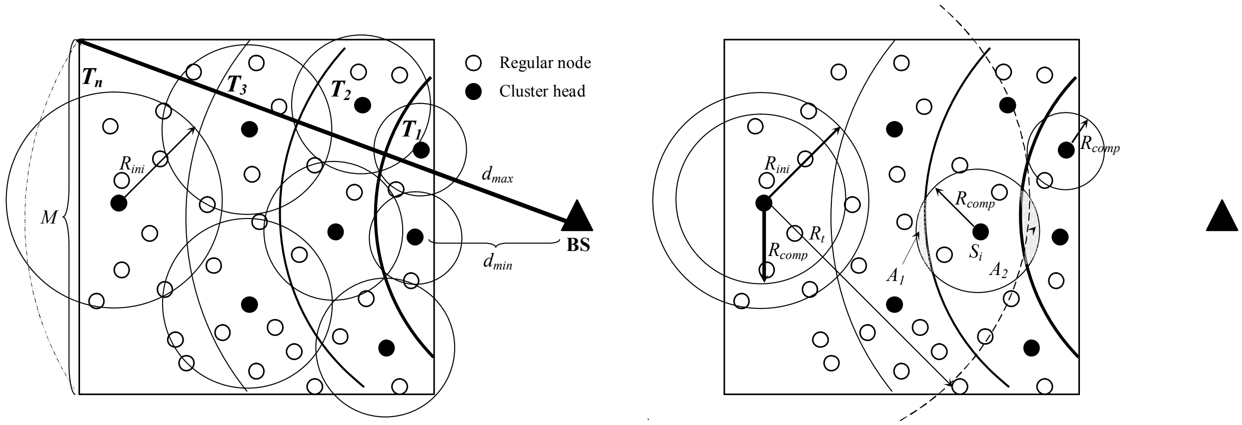
Each sensor node then infers its track number by

$$S_T = \lceil \frac{d_{(ID,BS)} - d_{min}}{d_T} \rceil. \tag{5}$$

Once each node knows its track number, the CH election starts. First some nodes are elected as CCHs, then a subset of them is picked as CHs. The criterion for CH candidacy in EPUC is the residual energy, so that the CHs stay operational for the longest time and thus the network lifetime and service reliability are improved. Each node broadcasts a message, called *CCH-Inf*, to all nodes within its competition range R_{comp} and waits for t_w seconds to receive similar messages from all its neighbors. Note that this message contains the residual energy (E_{res}) of node, the node ID and S_T. Once the node received such a message, it compares its E_{res} with that of its neighbors. If the node found that its E_{res} is larger than all neighbors, it announces itself as CCH by broadcasting the *CCH-ADV* message to all the nodes within a range equals D_{thr}. Otherwise, the node leaves the competition and waits for t_w seconds to receive the *CH-ADV* message. Note that the waiting time should be rational; not too short as some nodes may not receive the messages of neighbors, and not too large to overload the network (an analytical estimate for t_w is provided in subsection B). Note also that there are some *isolated* nodes which declare themselves as CHs directly.

The final CHs are elected among the CCHs which have a distance greater than or equal to D_{thr} from their CCH neighbors. As mentioned earlier, in order to form the unequal clusters, different D_{thr} are used in the network. More accurately, each track has its specific D_{thr}, different from other tracks. For example, the closer tracks to the BS, like T_1 in Fig. 1(a), have smaller D_{thr} than farther tracks to the BS, like T_3. When a CCH receives a *CH-ADV* from a CCH within its track and that is less that R_c, it joins the sender as a cluster member. Otherwise, the CCH elects itself as CH and declares its new status by broadcasting the *CH-ADV* to all the nodes within its cluster range R_c. Noting that the CH competition is performed within the tracks separately; however, there are conditions in which the CCHs receive the *CCH-ADV* from the nodes of a different track. In such a case, they quit the competition and change their status to ordinary node (without join the cluster since it is in a different track). This suggests that the elected CHs are typically located at the center of the tracks which minimizes the overlapping between clusters. Fig. 1 shows unequal clustering performed by EPUC and the pseudo code of which is presented in Algorithm 1.

After CH selection, clusters are formed. Each node finds the closest CH based on RSSI of the *CH-ADV* message. The node then sends a *Join-Req* message. Each node periodically senses the environment and sends its data to the CH. The CH after gathering and aggregating the data, transmits them to the BS through a multi-hop path among the CHs.



(a) Tracked and unequal clustered network by EPUC.

(b) Different ranges of a node in EPUC architecture. As is seen, node's R_t covers the next track, and R_{comp} differs from track to track.

Fig. 1. A general overview of a typical EPUC formed topology.

Algorithm 1 Distributed pseudo code of CH election in EPUC for node i

```
CH Elction Phase

  Tracking

1.  wait to receive the hello message
2.  IF the hello message is received THEN
3.      estimate d_{i,BS} and find out S_T based on Eq. (5)
4.  ELSE
5.      go to step 1
6.  ENDIF

  Local competition

7.  estimate E_res
8.  broadcast CCH-Inf within R_comp
9.  wait t_w to receive CCH-Inf
10. IF CCH-Inf is received THEN
11.     check the received CCH-Inf messages
12.     IF the CCH-Inf is received from a node within the same track
13.         IF ∀j, E_res(i) > E_res(j) THEN
14.             broadcast CCH-ADV within the range equals D_thr
15.         ELSEIF ∀j, E_res(i) ≥ E_res(j) and ∃j,
            E_res(i) = E_res(j) THEN
16.             IF ∀j, i > j THEN
17.                 broadcast CCH-ADV within the range equals D_thr
18.             ELSE
19.                 wait t_w for CH-ADV
20.             ENDIF
21.         ELSE
22.             wait t_w for CH-ADV
23.         ENDIF
24.     ElSE
25.         ignore the message
26.     ENDIF
27. ELSE
28.     broadcast CCH-ADV within the range equals D_thr
29. ENDIF

  Distance Condition

30. IF the current node is a CCH THEN
31.     wait t_w to receive CCH-ADV
32.     IF CCH-ADV is received THEN
33.         IF CCH-ADV is received from a node within the
            same track
34.             join the sender
35.         ELSE
36.             IF d_{i,j} ≥ D_thr THEN
37.                 broadcast CH-ADV within R_c
38.             ELSE
39.                 wait t_w for CH-ADV
40.             ENDIF
41.         ENDIF
42.     ELSE
43.         broadcast CH-ADV within R_c
44.     ENDIF
45. ELSE
46.     wait t_w for CH-ADV
47. ENDIF
```

B. Theoretical Analysis

In this section, we analyze the parameter settings for generating unequal clustering, e.g., D_{thr}. As mentioned earlier, the main aim of EPUC is to evenly distribute the load among all CHs. To do so, the distance among the CHs in each track should be adaptively adjusted so that the number of member nodes and thus the consumed energy of the CHs in different tracks could be equalized.

Let E_{ch_n} and $E_{ch_{n-1}}$ indicate the consumed energy by a CH of tracks n (the last track, see Fig. 1(a)) and $n-1$, respectively, which are computed as follow:

$$E_{ch_n} = E_r(N_{cl_n} - 1) + \rho l E_{da} N_{cl_n} + E_t, \tag{6}$$

where E_{da} is data aggregation energy, N_{cl_n} is the number of members of a cluster within track n and ρ is the aggregation coefficient, and

$$\begin{aligned} E_{ch_{n-1}} = {} & E_r(N_{cl_{n-1}} - 1) + \rho l E_{da} N_{cl_{n-1}} + E_r(k_n/k_{n-1}) \\ & \times \rho N_{cl_n} + E_t, \end{aligned} \tag{7}$$

where k_n and k_{n-1} is the number of clusters within tracks n and $n-1$. For the sake of simplicity, we consider all packets from track n are equally split on the k_{n-1} CHs in track $n-1$, so k_n/k_{n-1} is the share of received data that a CH gets from previous track. Since the condition of $E_{ch_{n-1}} = E_{ch_n}$ should be satisfied, so by combining Eq. (6) and (7), and also according to Eq. (2) and (3), we have

$$N_{cl_{n-1}} = \left(\frac{E_r + \rho l E_{da} - \rho(k_n/k_{n-1})E_r}{E_r + \rho l E_{da}}\right) N_{cl_n}. \tag{8}$$

According to Eq. (8), the members of a cluster in track $n-1$ should be fewer than that in track n, in order to satisfy the condition $E_{ch_n} = E_{ch_{n-1}}$. Now using Eq. (1) and (8), we can compute the desired cluster radius of $(n-1)$-th track as

$$N_{cl_{n-1}} = \lambda \pi R^2_{c_{n-1}}, \tag{9}$$

so

$$R_{c_{n-1}} = \sqrt{\frac{N_{cl_{n-1}}}{\pi \lambda}}. \tag{10}$$

With respect to Eq. (8) and (10)

$$R_{c_{n-1}} = \sqrt{\frac{(\frac{E_r+\rho lE_{da}-\rho(k_n/k_{n-1})E_r}{E_r+\rho lE_{da}})N_{cl_n}}{\pi\lambda}}, \quad (11)$$

or

$$R_{c_{n-1}} = R_{c_n}\sqrt{\frac{E_r+\rho lE_{da}-\rho(k_n/k_{n-1})E_r}{E_r+\rho lE_{da}}}, \quad (12)$$

where $R_{c_n} = R_{ini}$. With respect to Eq. (12), two parameters k_{n-1} and k_n affect the cluster radius of a track. Here, we can compute the desired distance among the CHs of each track. Since the communication between the CHs is performed by multi-hop, so it is desirable that the clusters have minimum overlapping. On the other hand, D_{thr} should be less than the maximum transmission power of a node or $D_{thr} < R_t$ (note $R_t \geq 6R_c$ [17]), in order for the multi-hop network to remain connected. Therefore, the distance condition between the CHs within each track should be twice of R_c of that track, or

$$D_{thr_n} = 2R_{c_n}. \quad (13)$$

To estimate R_{comp} and R_{ini} let us consider a sensor node, like S_i in Fig. 1(b), located within track i. The competition area of the node with radius R_{comp} is a circle around the node that is computed as

$$A_{comp} = \pi R_{comp}^2 - (A_1 + A_2), \quad (14)$$

where A_1 and A_2 are the areas located in the other tracks and are not included in the competition range of S_i (see Fig. 1(b)). Now, the average number of nodes that compete with S_i is

$$N_{comp} = A_{comp} \times \lambda. \quad (15)$$

Thus, the maximum number of control messages n_{max} (the message complexity overhead) that a node incurs during CCH competition equals transmitting one *CCH-Inf* by the node, receiving $N_{comp}\times$*CCH-Inf* from its neighboring competitive nodes, and receiving/transmitting one *CCH-ADV*. Thus

$$n_{max} = N_{comp} + 2. \quad (16)$$

Thus setting R_{comp} will be subject to trade-off; while it is important that sufficient number of CH candidates compete, the overhead should be reasonable. If we take $u = (A_1 + A_2)$, R_{comp} may be calculated by combining Eq. (14), (15) and (16)

$$R_{comp} = \sqrt{\frac{(n_{max}-2)+u\lambda}{\pi\lambda}}. \quad (17)$$

In order to define R_{ini}, consider the following application. The energy consumption of a CH located in the furthest track to the BS in a data round, i.e., each node reports its data once, with the maximum number of cluster members is

$$E_{ch} = E_r(N_{max}-1) + lE_{da}N_{max} + E_t, \quad (18)$$

where E_{ch} is the consumed energy by a CH in a round, and N_{max} is the maximum number of cluster members. Assuming that the maximum initial energy of a node is E_{max}, the CH operates for some rounds equals $n_r = E_{max}/E_{ch}$. According to Eq. (18) and if we take n_r as an average number of rounds, the maximum number of cluster members is achieved by

$$N_{max} = \frac{(E_{max}/n_r)+E_r-E_t}{E_r+lE_{da}}. \quad (19)$$

Thus, regarding $A_{ini} = \lambda N_{max}$, R_{ini} is computed as

$$R_{ini} = \sqrt{\frac{(E_{max}/n_r)+E_r-E_t}{(E_r+lE_{da})\pi\lambda}}. \quad (20)$$

Finally, we analyze t_w that depends a great deal on the number of nodes in R_{comp}. As computed earlier, if N_{comp} is the average number of nodes in R_{comp}, t_w is computed as

$$t_w = N_{comp} \times t, \quad (21)$$

where t is the average time interval between node's transmission, e.g., based on the MAC layer protocol.

V. Performance Validation

A. Simulation Setup

The operation of EPUC is simulated using MATLAB. We vary λ from 0.04 to 0.000625, in order to study the effect of network scales and density on the performance. Note that changing the node density while fixing the size of the area will change the number of nodes in the network, according to Eq. (1). The BS is located at $(M+50, M/2)$ and the node position follows a uniform random distribution. The following metrics are used to evaluate the performance:

- **FND:** The time interval between the start of the operations and the time the first node dies.
- **FNU:** The time interval until the first node becomes unreachable to the BS.
- **LND:** The time interval between the start of the operations and the time the last node dies.
- **CEP:** The average energy per packet for all nodes.
- **ECH:** The average residual energy of all elected CHs.

In our simulations, we use the radio model of [8]. Also we take the duration of each round to be equal to the time it takes to gather the data from all network nodes. In our simulation, we do not consider packet losses and we assume that all messages are successfully received by their destinations. We have implemented EEDC [3], HYMN, and UCR protocols to use as baseline for performance comparison. EEDC is an equal-sized clustering approach which picks CHs based on their residual energy while maintaining a constant distance among them. UCR and HYMN are described in section II. We have set the simulation parameters for these protocols consistent with their best performance, as stated in their respective publications. For EEDC, we let R_{comp} and D_{thr} vary from $15m$ to $30m$. In HYMN, the SCA is set to be $50m < SCA \leq ((M/4)+50)m$, and the desired CH probability p varies from 0.05 to 0.15. Finally for UCR, we let the initial competition range R_0 to be $[30, 70]$, the predefined threshold for tentative CH election $T = 0.3$ and the constant coefficient in defining the competition range $c = 0.3$. The other simulation parameters are summarized in Table II. The individual results are the average over 50 runs. When subjected to 95% confidence interval the results stayed within 6-10% of the same mean.

TABLE II
SIMULATION PARAMETERS

Parameter	Value
N	400
M	$(100 \sim 800)m$
λ	$(0.000625 \sim 0.04)$
BS	(M+50,M/2)
d_0	$87m$
ε_{fs}	$10pJ/bit/m^2$
ε_{mp}	$0.0013pJ/bit/m^4$
E_{el}	$50nJ/bit$
E_{da}	$5nJ/bit/signal$
R_t	$100m$
Initial Energy	$2J$
Data Frame	$100Byte$
Control Frame	$25Byte$

B. Simulation Results

In this section, the simulation results are presented. First we compare snapshots of the topologies that were created by EPUC and the baseline approaches for a sample configuration. Then, the performance under the various metrics is reported.

1) Cluster distribution: The cluster topologies formed by the compared protocols using Voronoi diagrams, for a sample of the simulated configurations are depicted in Fig. 2. We use Voronoi cells to show the distribution of CHs across the network and to hint the cluster sizes. As is seen in Fig. 2(a), EEDC forms equal-sized clusters that are evenly distributed over the network. The clustered network topology of HYMN is depicted in Fig. 2(b). As shown, there is no cluster in the SCA because the architecture of this zone is flat. On the other hand, in the zone out of the SCA, the clusters are formed by M-LEACH and significantly vary in size; some areas have several small clusters and other areas have only one large cluster. Although UCR performs unequal clustering and the clusters closer to the BS have relatively a smaller size than farther ones, the size of clusters is large and the approach fails to factor in the energy overhead for the CHs, as seen in Fig. 2(c). EPUC solves this problem using tracks and adaptive threshold distance among the CHs. As is seen in Fig. 2(d), 2(e) and 2(f), adjusting the number of tracks effectively controls the size of clusters in the vicinity of the BS. This helps these CHs save their energy for forwarding the received data from farther CHs to the BS, and as a result, the network operation becomes more energy-efficient and the topology stays fully-connected.

2) Network lifetime: In this section, we first report the performance of EPUC when $\lambda = 0.01$. Then, EPUC is compared with the other approaches. We varied the number of tracks to capture the impact on performance. Also, we varied R_{ini} in order to achieve the best value for this specific scenario (i.e., $\lambda = 0.01$), and then in the remaining of this section, we validate our analyses, performed in section IV, using simulations.

The FND for EPUC when R_{ini} and T varies and R_{comp} is constant at $15m$, is depicted in Fig. 3(a). It is noticed from Fig. 3(a) that when the number of tracks is small (say two or three), a smaller R_{ini} (e.g., 10-20m) is more suitable for the network. This is because when R_{ini} is set to a large value (e.g., $50-60m$), D_{thr} grows for each track, according to Eq. (13), and consequently the size of clusters and the intra- and inter-cluster communication costs are increased. The effect of large R_{ini} seems to be mitigated with increasing the number of tracks since it more clusters will be formed. It is worth mentioning that according to Eq. (4) the number of tracks is recommended to be 4 or 5. Fig. 3(a) implies that R_{ini} should be roughly set to $d_T/2$ in that case (i.e., about 30m). FNU and CEP are not sensitive to R_{ini} or T, as seen in Fig. 3(b) and 3(c). It is worth noting that the LND performance is similar to FNU, and the plot is not shown due to space constraints.

Now, we relate the performed analyses in the previous section with simulation results. In order to compute R_{ini}, we use Eq. (20). Consider the following scenario. As $M = 200m$ and $T = 5$, so we can divide the network into 10 virtual rectangular, where the sides length of each rectangular is $100m$ and $40m$. Based on Eq. (1), the average number of nodes within this area is 40. Since we considered a CH within the last track and with the maximum number of cluster members (N_{max}), so $N_{max} = 39$ (one node as the CH). For such a case, $E_{ch} = 0.012J$, according to Eq. (18) and the maximum distance between the CH and next-hop is $d_T = 40m$. Regarding Eq. (20), $R_{ini} = 41m$. As shown in Fig. 3(a), the best performance is achieved when R_{ini} is varies from 30 to 50m. Therefore, the performed analyses in the previous is concurred with the simulation results.

In addition, we study the effect of R_{comp} on the different metrics for $T = 5$, in Fig. 3(d) and 3(e). When R_{comp} is set to a small value (e.g., 0-5m), the CHs may be elected among the nodes with low residual energy, and D_{thr} is the only factor that controls the CHs count. Thus, some nodes get elected as CH without qualifying their residual energy and hence die soon. Therefore, the FND performance with a small R_{comp} is low, while the LND is increased. The CEP when R_{comp} and T vary is depicted in Fig. 3(e). Although the results suggest that the network has the lowest CEP when R_{comp} is set to 0-5m, this low CEP is because when R_{comp} is small, the FND is small as well; so the number of alive nodes decreases, and consequently, the average consumed energy per packet diminishes. Again with respect to Fig. 3(d) and 3(e), the data confirms the analysis in section IV-B. Finally, Fig. 3(f) shows the average percentage of the CHs in each track and the total count for $T = 2$. As shown, the average percentage of CHs in T_1 is greater than which in T_2, because, based on Eq. (13), D_{thr} between the CHs in T_1 is less than that in T_2, i.e., $k_1 > k_2$.

Fig. 4 compares the performance of EPUC with EEDC, HYMN, and UCR. As is observable from Fig. 4(a), EPUC outperforms EEDC, HYMN, and UCR protocols in terms of the FND, FNU, and CEP. Note that HYMN has a large LND because some nodes of the SCA are still alive while other nodes in the network have died. Thus, the network has no

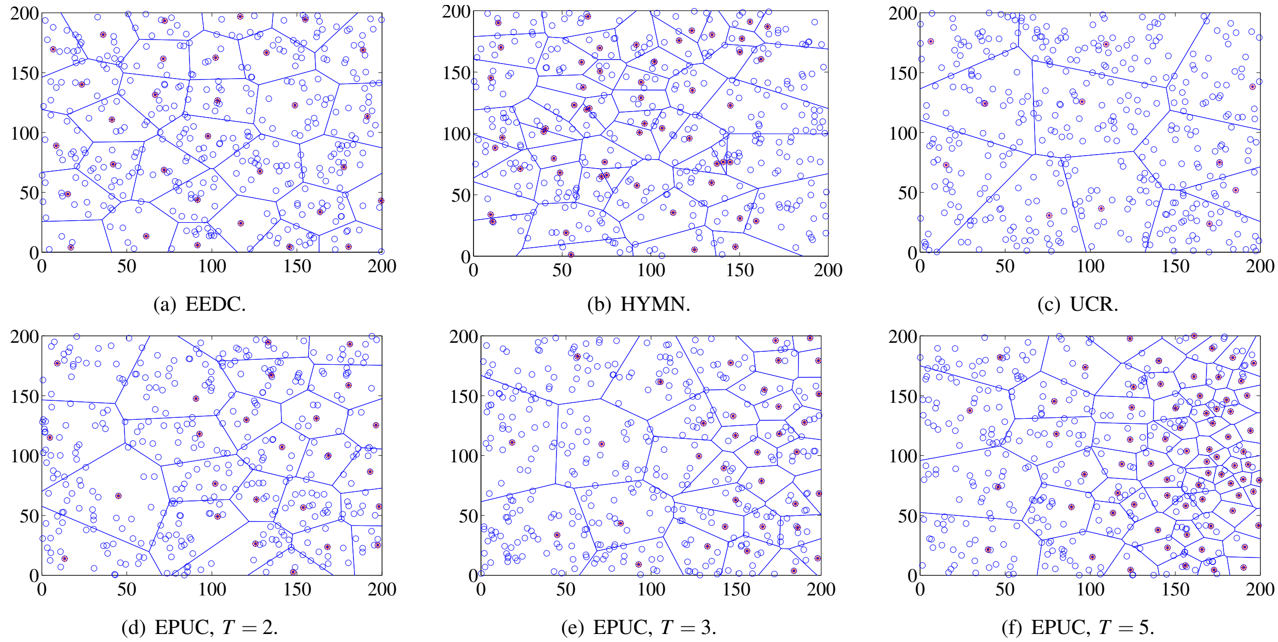


Fig. 2. The formed clusters using Voronoi diagrams in four different protocols when $\lambda = 0.01$.

coverage from other zones in the field and there are only 4% of the nodes alive and in the vicinity of the BS. The simulation results for HYMN indicate that all nodes out of the SCA died after 5000 rounds and only the nodes of the SCA stay alive until the 9090 round.

The FND performance for the protocols when the density varies is depicted in Fig. 4(b). The results indicate that when the density is greater (i.e. $M = 100m$ so that $\lambda = 0.04$), EPUC along with EEDC, which performs equal-sized clustering, has the best FND. This is because the traffic load is low on nodes close to the BS. According to Fig. 4(b), when the density diminishes, unequal clustering would be more suitable than equal-sized clustering. For example, EPUC and UCR shows a better FND than EEDC and HYMN in smaller densities (e.g., $\lambda = 0.0025$-0.000625). In general, EPUC outperforms all approaches in all mentioned densities. Note that for small node densities the FND decreases, because the intra- and inter-cluster communication costs grow. More precisely, when N stays constant at 400 and the length of network sides increases, the inter-CH path length is unnecessarily extended, and thus the consumed energy is increased. As a conclusion, if we take the FND as the comparison criterion, EPUC averagely improves the network lifetime about by 30-40%, compared to the other baseline approaches.

Finally, in order to show the variation in the average residual energy of elected CHs, the ECH is plotted in Fig. 4(c). Note that we compare to maximum residual energy of the nodes in the network, measured by averaging the top 5% of the node population in terms of remaining energy. As shown, EPUC yields ECH that is very close to the maximum in the network. This is the case for EEDC with a bit lesser ECH than EPUC. This is because the residual energy is the top criterion of CH election in both approaches. Since CHs are randomly elected in UCR and HYMN, the ECH is lower than EPUC and EEDC and vary over the different rounds. The ECH is an important metric because the elected CHs should stay operational for the longest time in order to handle their tasks (e.g., receiving, aggregating and transmitting the data).

VI. Conclusion

This paper has presented an Energy- and Proximity-based Unequal Clustering (EPUC) mechanism for WSNs. EPUC opts to overcome the uneven energy consumption rate of nodes in the vicinity of the BS due to the increased data relaying activity. CHs are selected in EPUC based on proximity to the BS and their energy reserve. The area is divided into tracks centered at the BS and the cluster count is increased as we get closer to the BS. Nodes with the most remaining energy are designated as CHs through a track-based competition. Simulation results confirms the effectiveness of the proposed EPUC mechanism in prolonging the network lifetime, compared with existing approaches. In the future, we plan to expand EPUC to support a network with variable size of tracks under the connectivity constraint.

Acknowledgement

This work is supported by the National Science Foundation, award # CNS 1018171.

References

[1] A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer communications*, vol. 30, no. 14-15, pp. 2826–2841, 2007.

[2] A. E. Abdulla, H. Nishiyama, J. Yang, N. Ansari, and N. Kato, "Hymn: a novel hybrid multi-hop routing algorithm to improve the longevity of wsns," *IEEE Trans on Wireless Comm*, vol. 11, no. 7, pp. 2531–2541, 2012.

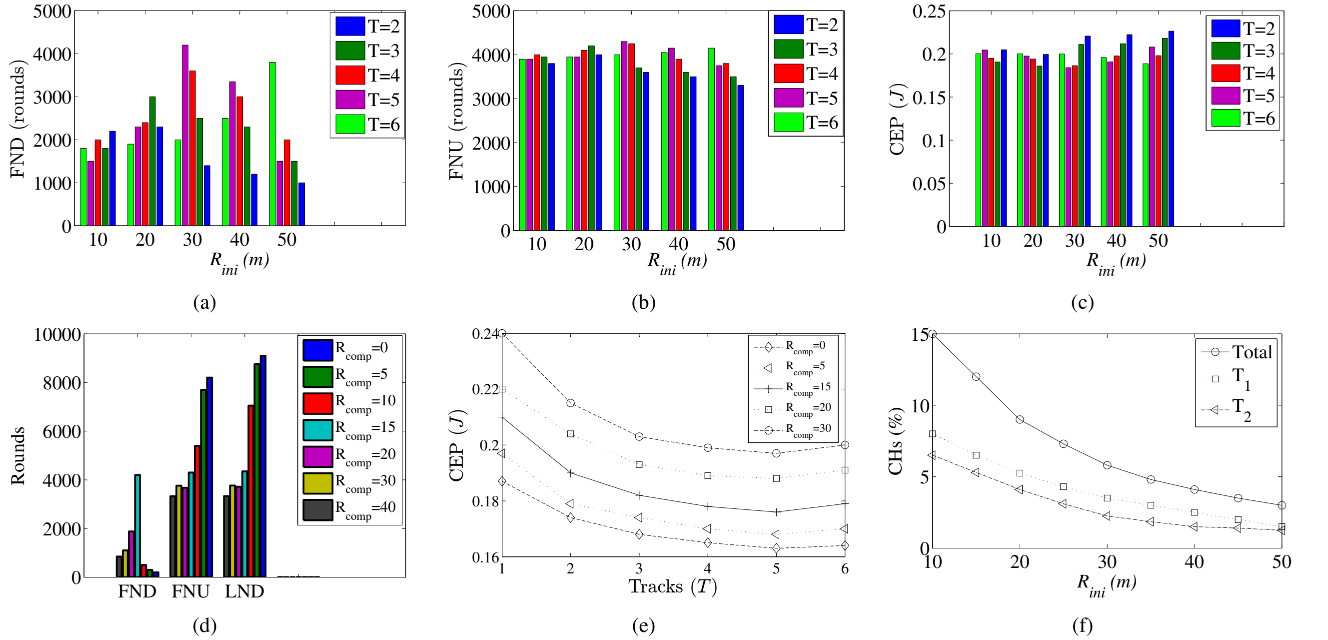


Fig. 3. The metric study of EPUC when $\lambda = 0.01$. (a), (b), and (c) The FND, FNU, and CEP metrics for EPUC when T and R_{ini} varies and $R_{comp} = 15m$. (d) Different metrics of EPUC when R_{comp} varies and T and R_{ini} are constant at 5 and 30m, respectively. (e) The CEP of EPUC when T and R_{comp} vary. (f) The average percentage of CHs of two-tracked EPUC in each layer and the total count.

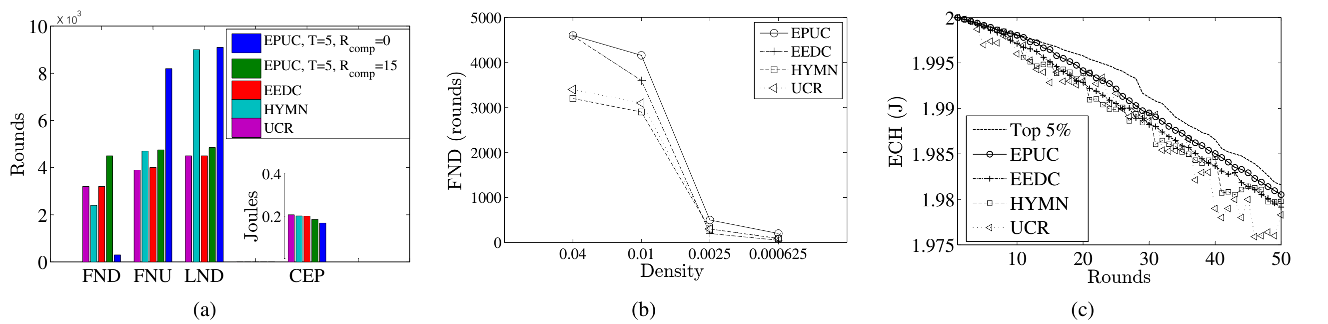


Fig. 4. Comparing EPUC with other protocols when $\lambda = 0.01$. (a) Different metrics of four protocols. (b) The FND in four protocols when λ varies from 0.04 to 0.0000625. (c) The ECH of four protocols compared with the top 5% residual energy of nodes in the network.

[3] M. M. Afsar and M.-H. Tayarani-N, "A novel energy-efficient and distance-based clustering approach for wireless sensor networks," *Advances in Intelligent Sys. & Comp.*, vol. 223, pp. 177–186, 2014.

[4] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Comm magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[5] H. Bagci and A. Yazici, "An energy aware fuzzy approach to unequal clustering in wireless sensor networks," *Elsevier, Applied Soft Computing*, vol. 13, pp. 1741–1749, 2013.

[6] ——, "An energy aware fuzzy unequal clustering algorithm for wireless sensor networks," in *Fuzzy Systems*, 2010, pp. 1–8.

[7] G. Chen, C. Li, M. Ye, and J. Wu, "An unequal cluster-based routing protocol in wireless sensor networks," *Wireless Networks*, vol. 15, no. 2, pp. 193–207, 2009.

[8] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans on Wireless Comm*, vol. 1, pp. 660–670, 2002.

[9] S. Lee, H. Choe, B. Park, Y. Song, and C. Kim, "Luca: An energy-efficient unequal clustering algorithm using location information for wireless sensor networks," *Wireless Personal Communications*, vol. 56, pp. 715–731, 2011.

[10] C. Li, M. Ye, G. Chen, and J. Wu, "An energy-efficient unequal clustering mechanism for wireless sensor networks," in *Conf. on Mobile Adhoc and Sensor Systems*, 2005, pp. 8–15.

[11] T. Liu, Q. Li, and P. Liang, "An energy-balancing clustering approach for gradient-based routing in wireless sensor networks," *Computer Communications*, vol. 35, pp. 2150–2161, 2012.

[12] V. Mhatre and C. Rosenberg, "Homogeneous vs heterogeneous clustered sensor networks: a comparative study," in *IEEE Conf. on Communications (ICC)*, vol. 6, 2004, pp. 3646–3651.

[13] H. Nishiyama, A. E. Abdulla, N. Ansari, Y. Nemoto, and N. Kato, "Hymn to improve the longevity of wireless sensor networks," in *IEEE Conf. on Global Communications (GLOBECOM)*, 2010, pp. 1–5.

[14] S. Soro and W. Heinzelman, "Prolonging the lifetime of wireless sensor networks via unequal clustering," in *IEEE Conf. on Parallel & Distributed Processing (IPDPS)*, 2005, pp. 8–15.

[15] C.-K. Toh, "Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks," *IEEE Communications Magazine*, vol. 39, no. 6, pp. 138–147, 2001.

[16] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[17] O. Younis and S. Fahmy, "Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Trans on Mobile Comp.*, vol. 3, pp. 366–379, 2004.

[18] J. Yu, Y. Qi, G. Wang, Q. Guo, and X. Gu1, "An energy-aware distributed unequal clustering protocol for wireless sensor networks," *Journal of Distributed Sensor Networks*, pp. 1–8, 2011.