This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE Globecom 2011 proceedings.

A Distributed Energy-efficient Re-Clustering Solution for Wireless Sensor Networks

Yichao Jin[1], Dali Wei[2], Serdar Vural[1], Alexander Gluhak[1] and Klaus Moessner[1]

1 CCSR, Faculty of Engineering and Physical Sciences, University of Surrey, UK
Email: {yichao.jin, s.vural, a.gluhak, k.moessner}@surrey.ac.uk
2 Jiangsu Tianze Infoindustry Company Ltd, China. Email: dlwei@ieee.org

***Abstract*—Clustering algorithms are widely used in Wireless Sensor Networks (WSNs), which however incurs significant energy consumption at Cluster Headers(CHs). Therefore, a re-clustering operation is typically used to balance the workload, where different CHs are selected and clusters are reorganized. However, a considerable number of control messages is initiated during this process which inevitably consumes on-board node energy. Hence, the question of how often the network should perform the re-clustering operation needs to be addressed. In this paper, a distributed re-clustering solution is proposed, which provides an energy-efficient re-clustering rate to conserve node energy while also equalizing the node energy consumption across the network. The proposed algorithm calculates the approximate amount of energy required to reorganize the clusters and to deliver the sensory data. By properly predicting the levels of the energy consumptions values, the appropriate frequency of performing the re-clustering operation can be determined, which reduces control message overhead. To the best of our knowledge, this is the first work that analytically analyzes the overhead in re-clustering a WSN, groups re-clustering rounds to reduce this overhead, and simultaneously equalizes node lifetimes. Performance results show that the proposed algorithm outperforms two other popular clustering algorithms in node energy conservation and node lifetime equalization.**

I. Introduction

Energy efficiency is a profound requirement of Wireless Sensor Networks (WSNs) due to the strictly limited amount of energy in battery-powered sensor nodes. Distributed hierarchical clustering [1], [2], [3] is a promising solution to conserve sensor energy, where Cluster-Heads (CHs) are polled via *message gossiping* among nodes in local areas, and once selected, they act as local controllers of network operations. Thus, CHs often consume much more energy than the cluster *member nodes*, and hence they become '*energy hot-spots*' of the network [4]. Early energy depletion of these critical nodes can cause serious consequences, such as network service disruption and communication link breakages. Therefore, proper *re-clustering* of the network can distribute the high workload among all nodes and avoid the failure of the CHs, which is vital to ensure a sufficiently long network lifetime.

A. Problem statement

Similar to some existing clustering algorithms (e.g. [5], [6], [7]), this work primarily addresses periodic data gathering applications and assumes a single-hop communication between a single data sink and CHs. In each data gathering period, every node contributes a total amount of l bits of data sent to and processed at the CH and then delivered to the sink. Each such period is defined as a *Data Collection Round (DCR)*. An example is given in Fig.1. For simple demonstration purposes, only three nodes A, B and C take part in clustering and one node is selected to be the CH in each DCR. Conventional clustering algorithms have a re-clustering operation in every DCR, and a node that has already been served as a CH might be selected again in later DCRs (Fig.1(a)). Clearly, this method incurs a large amount of control message overhead and additional service delays.

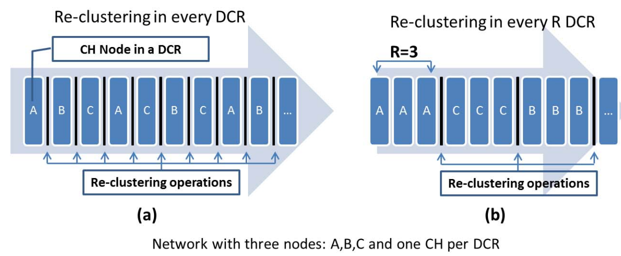


Fig. 1: Comparison of different Re-clustering rates

B. Our contribution

To avoid high energy cost of too frequent re-clustering rounds, we determinate a suitable period of CH service time that ideally allows each node to act as a CH only once during the entire network lifetime, as seen in Fig.1(b). To achieve this, we define a *re-clustering rate*, R, as the total number of consecutive DCRs between two re-clustering operations. Hence, the re-clustering operation performs every R DCRs. Aiming at extending the network lifetime, a **D**istributed **E**nergy-efficient **R**e-**C**lustering algorithm (DERC) is proposed, in which the appropriate value of R is determined by a prior analysis of energy consumption amounts in the transmission of both control messages and actual sensory data. To the best of our knowledge, this is the first study to analytically compute the appropriate frequency of re-clustering process to reduce control overhead. Furthermore, DERC offers better distribution of the CHs and forms suitable cluster sizes in the network in a way that allows an efficient and uniform use of the overall energy resources of a WSN. Heterogeneous node energy levels are considered for a realistic WSN deployment.

C. Related work

CH selection and cluster formation play an important role in energy conservation in WSNs. With regards to CH selection, a number of previous studies have been proposed for load balancing [5], [6], [8]. The LEACH protocol [5] is a well-known algorithm, in which a node decides by itself whether or not to become a CH with a certain probability. Although LEACH performs well in extending network lifetime in homogeneous WSNs where nodes have equal initial energy levels, nevertheless, its performance in heterogeneous WSNs is poor [6]. For this reason, EEHC [6] is proposed to select CHs based on their initial node energy; i.e. where nodes with more energy have a higher probability of becoming CHs. HEED [8], on the other hand, is based on the residual node energy levels. Both algorithms have a fairly uniform distribution of CHs in heterogeneous WSNs.

Another approach is to enhance cluster formation. Algorithms [9], [10], [11] require non-CH nodes to join the nearest CH in order to reduce the energy consumption of intra-cluster communications. Alternatively, to address the unbalanced energy consumption caused by inter-cluster communication, a group of studies allocate different number of nodes to clusters based on their distances to the data sink [7], [12], [13], [14]. For instance, in EECS [7], since CHs in areas with longer distance to the sink require higher transmission power to access the sink, fewer number of member nodes are allocated to clusters that are far away from the sink.

All of the above algorithms disregard the re-clustering rate and assume that a re-clustering operation is performed in every DCR. Thus, these algorithms incur extra energy consumption on additional control messages, which is addressed in this paper.

The rest of the paper is organized as follows. The detailed mathematical analysis of the re-clustering rate and node lifetime equalization are provided in Section 2. In Section 3, the performance results are demonstrated. Finally, Section 4 concludes the paper.

II. THE DERC ALGORITHM

A. Network structure and energy models

The network has the following properties: N nodes are randomly deployed in an area of size $a \times b$ with a uniform distribution of locations, creating a node density of $\sigma = \frac{N}{ab}$. The data sink is a computationally powerful node with sufficient resources. Once deployed, the sink and the sensors have fixed locations. Nodes can adjust their transmission powers to change their communication ranges. Each node learns its approximate distance to the data sink through Received Signal Strength Indication (RSSI) or other distance estimation methods. Nodes have unequal levels of initial battery energy at the time of network deployment, i.e. a node j has an initial energy of $E_0(j) = E_0(1 \pm x_j)$, where E_0 is the average deployment energy and x_j denotes the extra percentage of energy at node j. A popular transmission energy model (e.g. [5], [6], [15]) is considered in DERC, given by:

$$E_t = \begin{cases} (e_t + \varepsilon_{fs} \cdot d^2) \cdot l, & : \quad d < d_0 \quad (freespace) \\ (e_t + \varepsilon_{mpf} \cdot d^4) \cdot l, & : \quad d \geq d_0 (multi-path) \end{cases}$$

$$E_r = e_r \cdot l, \tag{1}$$

where E_t and E_r represent energy consumption of transmitting and receiving l bits of data. The baseline energy consumption in operating the transmitter and receiver radios are expressed as e_t and e_r, respectively. The amount of transmission energy consumption depends on the distance d between two nodes and a certain distance threshold d_0 [5].

B. Re-clustering rate estimation

In this section, a suitable re-clustering rate is estimated in order to reduce clustering overhead and extend network lifetime. The network lifetime (L) is defined as the period of time until the instance when the first node fails due to energy depletion [16]. Note that there exist other more complex network lifetime definitions [17], e.g. the time duration from the beginning of network operation until a certain percentage of nodes are still online and able to deliver the required performance. However, such definitions are quite application-specific and may not apply to all WSNs. Therefore in this paper, a simple but more common definition of network lifetime is adopted.

In order to prolong L, we aim to have a sufficient re-clustering frequency to evenly distribute the CH loads and ideally have equal rates of energy consumption among nodes. For simplicity reasons, we assume that nodes in the same cluster have approximately the same lifetime. For instance, $\tau(i)$ is the lifetime of all nodes in cluster C_i. Thus, we can target at equalizing the lifetime values that represent individual clusters, rather than individual nodes, given by:

$$\tau(i) = \tau(1) = \tau(2) = \cdots = \tau(M) = L, \tag{2}$$

where M is the total number of clusters in the network.

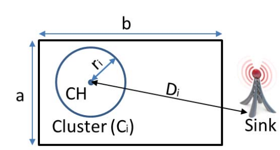


Fig. 2: Cluster model in single-hop WSNs

1) Problem formulation: With a node density of σ, we assume a cluster C_i is formed by approximately a total number of $\pi r_i^2 \sigma$ nodes, where r_i is the cluster radius as seen in Fig.2. In order to have an equalized energy consumption within C_i, the role of CH needs to be rotated evenly among the nodes, which incurs control overhead. Thus, since each node in C_i should ideally serve as a CH exactly once during the network lifetime, the ideal re-clustering rate R in C_i is:

$$R = \frac{\tau(i)}{\pi r_i^2 \sigma} \Rightarrow R \approx \frac{L}{\pi r_i^2 \sigma} \tag{3}$$

However, the cluster size r_i may vary in the network depending on the clustering algorithm used. The larger a cluster is, the less time each node has as a CH. To ensure that the lifetime of the clusters are equalized, according to (3), the value of R should be different in clusters with different sizes. Nevertheless, in a practical WSN, re-clustering is an operation that is performed across the network. That is to say, the same value of R must be used in all clusters. Hence, we have the final re-clustering rate R_f:

$$R_f \approx \frac{L}{\pi r_{max}^2 \sigma}, R_f \in \mathbf{N}^+, \tag{4}$$

where $r_{max} = \max(r_1, r_2, \ldots, r_M)$ is the largest cluster size in the network. In order to determine R_f, the relationship between r_{max} and the expected network lifetime L needs to be derived.

2) Estimation of r_{max} and R_f : We define the total energy consumption of a cluster C_i in a DCR as $E_{DCR}(i)$, which is formulated as a function of the cluster radius r_i, the CH distance D_i to sink, and the re-clustering rate R as: $E_{DCR}(i) = f(r_i, D_i, R)$. In the following, we use $f(r_i, D_i, R)$ as a cost function to explain how we solve the problem. Details of $f(r_i, D_i, R)$ are provided in Section II-C.

The total initial energy of the cluster C_i can be calculated by $E_0(C_i) = \sum_{j \in C_i} E_0(1 \pm x_j) \approx \overline{E_0}\pi r_i^2 \sigma$, where $\overline{E_0}$ is the average initial node energy. Now, we have the lifetime of a cluster C_i in terms of DCRs as:

$$\tau(i) = \frac{\overline{E_0}\pi r_i^2 \sigma}{f(r_i, D_i, R)} \Rightarrow L \approx \frac{\overline{E_0}\pi r_i^2 \sigma}{f(r_i, D_i, R)}. \tag{5}$$

Equation (5) is plugged in Equation (2) for all clusters $C_i (i = 1, \cdots, M)$. With this, the objective is to find a value of L that provides cluster lifetime equalization. We also strive to maximize this equalized value. Hence, when multiple solutions for Equation (2) exits, we choose the one with the largest L. To achieve this, our strategy is to greedily increase a trial value of L to get the corresponding set of cluster radius values $\{r_1, r_2, \cdots, r_M\}$ based on the distance D_i. The upper bound of L is reached when we start to get imaginary solutions for r_i.

The solution of (2) and (5) also requires a known value of the re-clustering rate R. However, the final value of R, R_f, can only be found via (4), which in turn requires the solution set $\{r_1, r_2, \cdots, r_M\}$ of (2) and (5). To resolve this causality dilemma, an iterative algorithm is used to feed the calculated value of R into (2) and (5) to obtain a set of r_i values, and then retrieve a new value of R. This iterative process is continued until the change in R is negligible (less than a threshold of δ). The relationship between r_i and D_i with different node densities are shown in Fig.3 and Algorithm 1 outlines this strategy.

The performance of DERC greatly depends on the relationship between r_i and D_i, as well as the re-clustering rate R_f. These parameters are determined before the network starts to operate, which are then pre-programmed into the nodes.

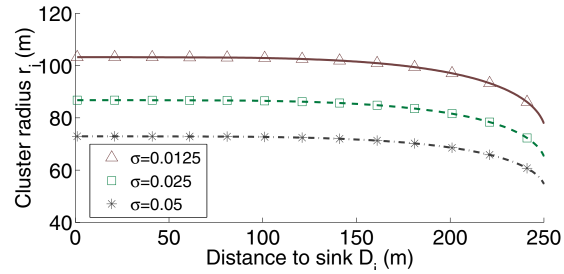


Fig. 3: Relationship between r_i and D_i ($R_f = 3$)

Algorithm 1 Computation of the re-clustering rate R_f

R $\leftarrow 1$, $R_f = 0$
while $R_f == 0$ **do**
  $L \leftarrow 1$
  $f(r_i, D_i, R) \leftarrow R$
  for Each $D_i \in \{D_1, D_2, \cdots, D_M\}$ **do**
    Compute r_i (equation (5))
    $\{r_1, r_2, \cdots, r_M\} \leftarrow r_i$
  end for
  while $\{r_1, r_2, \cdots, r_M\}$ are Real and Non-negative **do**
    $L \leftarrow L + 1$
    for Each $D_i \in \{D_1, D_2, \cdots, D_M\}$ **do**
      Compute r_i (equation (5))
      $\{r_1, r_2, \cdots, r_M\} \leftarrow r_i$
    end for
  end while
  $L \leftarrow L - 1$
  $r_{max} \leftarrow \max(r_1, r_2, \cdots, r_M)$
  Compute R_f (equation (4))
  if $|R_f - R| > \delta$ **then**
    $R = R_f$
    $R_f = 0$
  end if
end while
return $\{r_1, r_2, \cdots, r_M\}$, r_{max} and R_f

C. Cluster energy consumption model

In this section, the energy consumption model $f(r_i, D_i, R)$ used to calculate the energy consumption $E_{DCR}(i)$ is presented, however, note that any other model can also be fit this framework to estimate R_f. Furthermore, though the current model is based on single-hop communications, but it can be extended to multi-hop scenarios.

$E_{DCR}(i)$ consists of: 1) the energy consumption $E_{Cluster}(i)$ for control message overhead to form the cluster C_i during a re-clustering round; 2) the data transmission energy consumption $E_{Data}(i)$. Since $E_{Cluster}(i)$ takes place only once in every R DCRs, $\frac{1}{R}$ of $E_{Cluster}(i)$ is dedicated to each DCR, hence $E_{DCR}(i)$ is equal to:

$$E_{DCR}(i) = \frac{1}{R} E_{Cluster}(i) + E_{Data}(i). \tag{6}$$

1) Energy consumption in cluster reformation: The selection of CHs has two phases. During the first phase, a set of nodes elect themselves as CH-candidates with a probability *T*. Since we consider heterogeneous node energy levels, this probability is further scaled down based on the node residual energy: $P(j) = T\frac{E_0(j)}{\overline{E_0}}$, where $\overline{E_0}$ is the initial average energy of all nodes in the network. In the second phase, a competition mechanism similar to EECS [7] is used. However, in DERC, the candidate nodes compete with each other by broadcasting their residual energy levels within the own competition range r_i based on the distance to sink D_i. Upon receiving a message from another candidate with a higher energy level, a CH-candidate defers from acting as a CH. Finally, all the remaining CH-candidates successfully become the final CHs. Since node locations are uniformly distributed and the non-CH nodes join the closest CH, a cluster area of C_i can be approximately represented by a circular region of radius r_i. Thus, there are on the average $\pi r_i^2 \sigma T$ CH candidates that compete with each other in a cluster and each candidate node listens to the competition messages resulting in a total listening of $\pi r_i^2 \sigma T - 1$ messages per candidate. Designating the length of a control packet as l_0, the signalling energy consumption of CH selection in C_i during a DCR is:

$$E_{CHselection}(i) = \pi r_i^2 \sigma T((e_t + \varepsilon_{fs} r_i^2) l_0 + (\pi r_i^2 \sigma T - 1) l_0 e_r). \quad (7)$$

Upon being selected, each CH broadcasts a message to announce its role and allows other non-CH nodes to join. In order to ensure that each non-CH node in the network receives at least one announcement packet and associate to one of the CHs, this range is defined as ϕr_i, where the tuning parameter ϕ guarantees at least 99% average association probability. Hence, considering the number of nodes in a given area of $\pi(\phi r_i)^2$ to be a Poisson random variable (due to uniform node locations) with a CH density of σp_i, ϕ can be found by: $1 - e^{-\sigma p_i \pi (\phi r_i)^2} \geq 0.99 \Rightarrow \phi r_i \geq \sqrt{\frac{2 \ln 10}{\sigma p_i \pi}}$, where p_i is the probability for a node to become a CH. Since there is a single CH inside each cluster, the probability of a node in a cluster C_i to become a CH during the selection process can be approximated by: $p_i = \frac{1}{\pi r_i^2 \sigma}$. Thus, we have $\phi \geq \sqrt{2 \ln 10}$. A value of ϕ that is smaller than $\sqrt{2 \ln 10}$ may lead to a case in which some non-CH nodes receive no messages from CHs. On the other hand, a large ϕ value causes significant waste of energy consumption on transmissions of the announcement packets and overhearing of these messages.

Note that, due to uniform distribution of node locations, the average distance of the nodes between themselves and their CH $\overline{d}$ can be calculated as: $\overline{d} = \int_0^{r_i} \frac{2\pi \mathbf{d}^2}{\pi r_i^2} \, d\mathbf{d} = \frac{2}{3} r_i$. Therefore, the total energy consumption to form the cluster after the determination of the CH in C_i can be obtained as:

$$\begin{aligned} E_{Formation}(i) = & [(e_t + \varepsilon_{fs}(\phi r_i)^2) l_0 + (\pi(\phi r_i)^2 \sigma - 1) e_r l_0] \\ & + [(\pi r_i^2 \sigma - 1) l_0 (e_t + \varepsilon_{fs}(\frac{2}{3} r_i)^2 + e_r)] \\ & + [(e_t + \varepsilon_{fs}(\phi r_i)^2) l_0 + (\pi r_i^2 \sigma - 1) l_0 e_r]. \end{aligned} \quad (8)$$

The first term is the energy consumptions of the CH to announce its role, plus reception of this packet by other nodes. The second term considers the energy consumptions of each non-CH node to send a control packet to associate itself to the CH, plus the every costs of receiving these messages at the CH. The last term is the energy consumption for the CH to distribute the time schedules to its member nodes.

Eventually, the total energy consumption for the clustering process for a cluster C_i can be expressed as:

$$E_{Cluster}(i) = E_{CHselection}(i) + E_{Formation}(i). \quad (9)$$

2) Energy consumption in data communication: Data communication events from a member node to the sink consists of three steps: intra-cluster communication $E_{Intra}(i)$, data aggregation $E_{Agg}(i)$, and CH-to-sink communication $E_{Inter}(i)$. Therefore, the total communication energy consumption in a DCR for a cluster C_i is:

$$E_{Data}(i) = E_{Intra}(i) + E_{Agg}(i) + E_{Inter}(i). \quad (10)$$

For $E_{Intra}(i)$, similar to the calculation in Section II-C1, we consider both transmission and reception events in a cluster with one CH and $\pi r_i^2 \sigma - 1$ member nodes. The total intra-cluster energy consumption can be approximated as:

$$E_{Intra}(i) = (\pi r_i^2 \sigma - 1) l (e_t + \varepsilon_{fs}(\frac{2}{3} r_i)^2 + e_r), \quad (11)$$

where l is the average packet length.

Similar to the most of the clustering algorithms (e.g. [5], [6], [7]), the energy consumption of the computational procedure for data aggregation in a cluster C_i is:

$$E_{Agg}(i) = l(\pi r_i^2 \sigma) E_{bit}, \quad (12)$$

where E_{bit} is the energy cost per bit.

Finally, as we consider single-hop communication for inter-cluster communication, the $E_{Inter}(i)$ can be obtained simply as the energy consumption to transmit the processed data packet to the data sink.

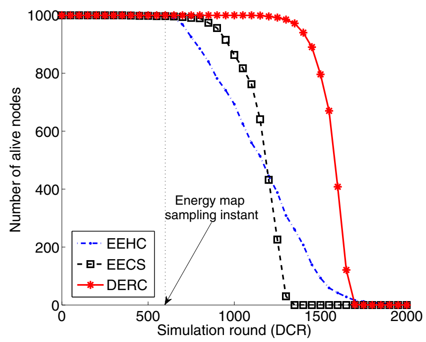
$$E_{Inter_i} = \begin{cases} l(e_t + \varepsilon_{fs} D_i^2) \cdot l, & : \quad D_i < d_0 \\ l(e_t + \varepsilon_{mpf} D_i^4) \cdot l, & : \quad D_i \geq d_0, \end{cases} \quad (13)$$

where D_i is the distance between the CH in the cluster C_i to the sink.

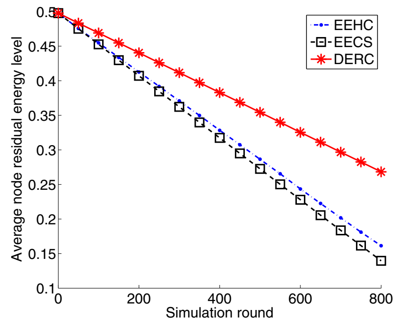
3) Network lifetime Estimation: Based on the energy consumption models in Section II-C1 and Section II-C2, we can finally combine all expressions for energy consumption. Equation (5) can then be reorganized as a biquadratic equation of r_i with an implicit relation with D_i and R:$Ar_i^4 + Br_i^2 + C = 0$, where:

$$\begin{aligned} A = & \varepsilon_{fs} \pi \sigma (\frac{l_0}{R}(T + \frac{4}{9}) + l\frac{4}{9}), \\ B = & \frac{l_0}{R} \pi \sigma ((T+1) e_t + (2\phi^2 - \frac{4}{9}) \frac{\varepsilon_{fs}}{\pi \sigma} + (\phi^2 + T + 2) e_r) \\ & + l((e_t + e_r + E_{bit}) \pi \sigma - \varepsilon_{fs} \frac{4}{9}) - \frac{\overline{E_0} \pi \sigma}{L}, \\ C = & \begin{cases} \frac{l_0}{R}(e_t - 4e_r) + l(\varepsilon_{fs} D_i^2 - e_r) \cdot l, & : \quad D_i < d_0 \\ \frac{l_0}{R}(e_t - 4e_r) + l(\varepsilon_{mpf} D_i^4 - e_r) \cdot l, & : \quad D_i \geq d_0. \end{cases} \end{aligned} \quad (14)$$

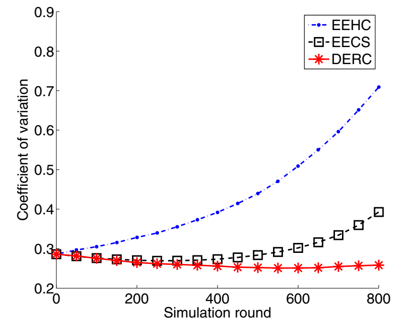
Now, we can use this model to compute r_i and R_f by using Algorithm 1.



(a) Number of alive nodes

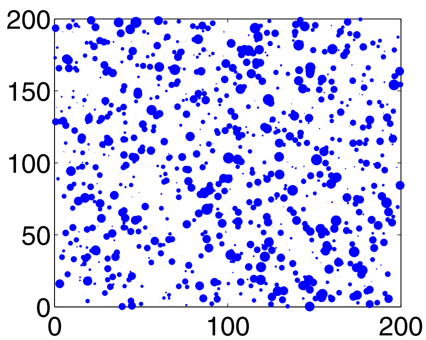


(b) Average node energy levels

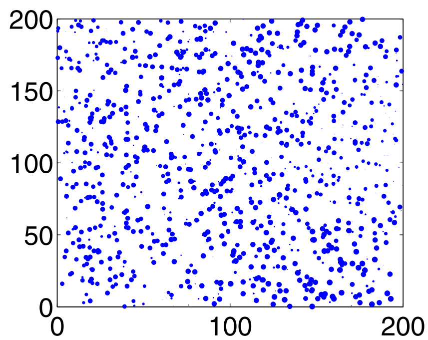


(c) Variation of node energy levels

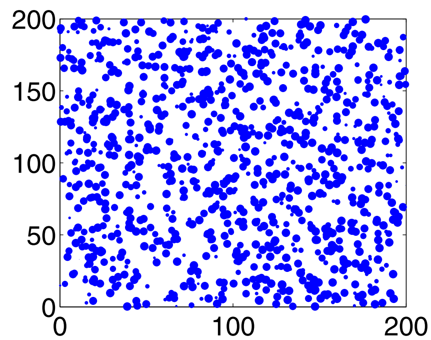
Fig. 4: Comparison of algorithm performances



(a) EEHC



(b) EECS



(c) DERC

Fig. 5: Node energy equalization: residual energy maps

III. Performance Evaluation

In this section, the performance of the DERC algorithm is evaluated by comparing it with EEHC [6] and EECS [7]. EEHC is similar to LEACH [5] but is proposed for heterogeneous networks with different initial node energy levels. Hence, EEHC is picked rather than LEACH in this study. On the other hand, EECS additionally targets at equalizing node lifetimes. A general comparison of the algorithms is listed in Table 1.

TABLE I: Comparison of the algorithms

	EEHC	**EECS**	**DERC**
Heterogeneous network	Yes	Yes	Yes
Node lifetime equalization	No	Yes	Yes
Require network wide CH announcement	Yes	Yes	No
Require re-clustering in every DCR	Yes	Yes	No

In simulations, the sink node is placed at the edge of a network with dimensions of 200m×200m at coordinates (200m,100m). Nodes have initial energy levels of $0.5(1 \pm 0.25)J$ with an average initial energy of 0.5J. 1000 nodes are deployed resulting in a node density of $\sigma = 0.025$. The CH-candidate probability $T = 0.1$ is selected according to [7]. The other parameters are selected as: $e_t = e_r = 50$ nJ, $\varepsilon_{mpf} = 0.0013\ pJ/bit/m^4$, $\varepsilon_{fs} = 10\ pJ/bit/m^2$, $E_{bit} = 5$ nJ/bit/signal, $d_0 = 87$ m, $l = 2000$ bits, $l_0 = 200$ bits.

The number of 'alive' nodes over simulation time is shown in Fig.4(a), which illustrates that DERC has significant performance improvement in extending network lifetime compared with EEHC and EECS. This is because DERC conserves node energy by reducing clustering control overhead and balances energy consumption among clusters by having suitable cluster sizes. Both DERC and EECS show a sharp decrease in the number of alive nodes demonstrating equalized energy consumption levels. Note that, in a scenario with perfectly balanced energy consumption values, all nodes should deplete their energy stocks at the same time. On the other hand, EEHC shows a gradual decrease in the number of alive nodes, indicating that some nodes run out of energy much more quickly than others.

DERC outperforms EECS and EEHC in energy conservation, as it is observed in Fig.4(b), where DERC has the highest average node residual energy levels, since it has better energy conservation features at its re-clustering mechanism.

DERC shows minimal variation in residual node energy levels, as seen in Fig.4(c), while EECS and EEHC have larger fluctuations. This shows that DERC has the best performance in equalizing node energy consumption throughout the simulation. To better illustrate this, we also provide maps of residual node energy levels for each of the algorithms at the instant when they start to lose nodes due to energy depletion. This event occurs for EEHC the earliest and is tagged with 'Energy map sampling instant' in Fig.4(a). The relevant energy maps are shown in Fig.5. In there plots, dots in indicate node energy levels (a larger dot represents a higher residual energy level). A significant variation in dot sizes means a poor balance of energy consumption. Fig.5 clears shows that DERC has the best performance in terms of energy conservation and equalization.

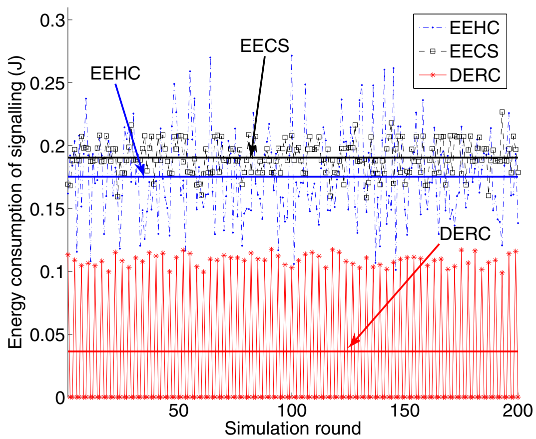


Fig. 6: Results for energy consumption of control overhead

Fig.6 shows that DERC has the least energy consumption in control overhead, which stems from two main factors: (1) DERC selects an appropriate CH-role announcement range, whereas both EECS and EEHC assume network wide CH broadcasts resulting in a significant energy consumption due to long-range transmissions and too much overhearing. (2) DERC reduces control overhead by performing necessary re-clustering operations as seen in Fig.6. In contrast, the other two algorithms perform a re-clustering operation at every DCR.

IV. Conclusion

A well self-managed clustered WSN should have a sufficiently long lifetime by conserving individual node energy levels as well as balancing the traffic loads over all the nodes. Appropriate re-clustering of the network can help make better decisions on workload balancing. On the other hand, having too frequent and unnecessary re-clustering operations can lead to significant energy waste due to an excessive number of control messages. In this paper, we built an analytical framework that models the energy consumption of both control message overhead of re-clustering and actual sensory data transmission. The DERC algorithm provides efficient rates of cluster reformation and offers balanced energy usage among all nodes. Our results demonstrate the effectiveness of the DERC algorithm in extending the network lifetime. The proposed re-clustering framework is also applicable to other energy consumption models.

Acknowledgment

This work has been partially supported by the European Union FP7 research projects SENSEI (IST-7-215923), and EXALTED (ICT-5-258512). The views expressed are those of the authors and do not necessarily represent the projects.

References

[1] C. Jiang, D. Yuan, and Y. Zhao, "Towards clustering algorithms in wireless sensor networks-a survey," in *Wireless Communications and Networking Conference, 2009. WCNC 2009. IEEE*, 2009, pp. 1–6.

[2] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communications*, vol. 30, pp. 2826–2841, October 2007.

[3] S. Bandyopadhyay and E. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, 2003.

[4] D. Wei, Y. Jin, A. Gluhak, R. Tafazolli, and K. Moessner, "Hot-spot issue aware clustering for wsns to extend stable operation period," in *Future Network and Mobile Summit, 2010*, june 2010, pp. 1–9.

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

[6] D. Kumar, T. C. Aseri, and R. Patel, "EEHC: Energy efficient heterogeneous clustered scheme for wireless sensor networks," *Computer Communications*, vol. 32, no. 4, pp. 662 – 667, 2009.

[7] M. Ye, C. Li, G. Chen, and J. Wu, "EECS: an energy efficient clustering scheme in wireless sensor networks," in *24th IEEE International Performance, Computing, and Communications Conference, IPCCC 2005*, 2005, pp. 535 – 540.

[8] O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366 – 379, 2004.

[9] G. Smaragdakis, I. Matta, and A. Bestavros, "SEP: A stable election protocol for clustered heterogeneous wireless sensor networks," in *In: Proc. of the Intl Workshop on SANPA*, 2004.

[10] L. Qing, Q. Zhu, and M. Wang, "Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks," *Computer Communications*, vol. 29, pp. 2230–2237, August 2006.

[11] Y. Jin, L. Wang, Y. Kim, and X. Yang, "EEMC: An energy-efficient multi-level clustering algorithm for large-scale wireless sensor networks," *Computer Network*, vol. 52, pp. 542–562, February 2008.

[12] D. Wei and H. Chan, "Equalizing cluster lifetime for sensor networks with directional data traffic to improve energy efficiency," in *CCNC 2008, 5th IEEE Consumer Communications and Networking Conference, 2008.*, 2008, pp. 714 –718.

[13] C. Li, M. Ye, G. Chen, and J. Wu, "An energy-efficient unequal clustering mechanism for wireless sensor networks," in *Mobile Adhoc and Sensor Systems Conference, 2005. IEEE International Conference on*, 2005, pp. 8pp.–604.

[14] F. Tashtarian, A. Haghighat, M. Honary, and H. Shokrzadeh, "A new energy-efficient clustering algorithm for wireless sensor networks," in *Software, Telecommunications and Computer Networks, 2007. SoftCOM 2007. 15th International Conference on*, 2007, pp. 1–6.

[15] Y. Jin, J. Jin, A. Gluhak, K. Moessner, and M. Palaniswami, "An intelligent task allocation scheme for multi-hop wireless networks," *IEEE Transactions on Parallel and Distributed Systems*, pp. 1–8, 2011.

[16] Y. Jin, D. Wei, A. Gluhak, and K. Moessner, "Latency and energy-consumption optimized task allocation in wireless sensor networks," in *Wireless Communications and Networking Conference (WCNC), 2010 IEEE*, 2010, pp. 1 –6.

[17] I. Dietrich and F. Dressler, "On the lifetime of wireless sensor networks," *ACM Transactions Sensor Networks*, vol. 5, pp. 5:1–5:39, February 2009.