A New Weight based Rotating Clustering Scheme for WSNs

Alsnousi Essa, Ahmed Y. Al-Dubai, Imed Romdhani and Mohamed A. Eshaftri
School of Computing, Edinburgh Napier University,
10 Colinton Road, Edinburgh EH10 5DT, U.K.
Email : {A. Essa; A. Al-Dubai; I. Romdhani; M. Eshaftri}@napier.ac.uk

Abstract—**Although energy efficiency in Wireless Sensor Networks (WSNs) has been investigated widely, it is still a challenging dilemma. Clustering mechanisms are among the most widely recommended approaches to sustaining a network throughout its lifetime. Despite a number of research activities associated with clustering in WSNs, some aspects of clustering have not yet been adequately investigated. The election of Cluster Head (CH) nodes is the key element in most clustering algorithms. The majority of the elected CHs in previous algorithms use a single metric as their principal criteria, such as remaining energy, distance to the base station (BS), and random probability. Unlike existing solutions, this paper proposes a new weighting based approach to electing CHs, considering the node's context; i.e. its transmission range, its degree, remaining energy and centrality from its neighbours. The novelty of our algorithm relates to the selection of a set of nodes with the highest weight as CH candidates, in order to compete for final CH. This will guarantee the selection of optimal cluster heads among sensor nodes. Moreover, this novel approach eliminates the re-clustering process for the entire network in each round, by rotating the CHs inside the created clusters in the first set up phase. The simulation experiments demonstrate that the new approach outperforms its counterparts (HEED, LCP and EEUC) with respect to the network's lifetime.**

Keywords— WSNs; Distributed clustering; Routing; Network lifetime.

I. Introduction

Wireless Sensor Networks (WSNs) are increasingly integral components of a wide range of applications employed to monitor environmental conditions, automate chain manufacturing systems and additional Internet of Things (IoT) applications [1][2]. In these applications, sensor nodes are distributed in a specific sensing field and have significant capabilities to communicate with each other using short-range wireless transmissions. Some applications require that their sensor nodes be scattered in harsh environments, such as forests and oil and gas fields, where replacing or recharging batteries is almost impossible or considered impractical [3]. Critical problems arise when sensor nodes are deployed with the expectation of forwarding their data constantly to the base station. This process has an enormous impact on the energy of the sensor nodes. Thus, developing an energy-efficient protocol is essential for these types of networks and heavy duty applications [5].

Minimising overall energy consumption is a further key requirement when planning wireless sensor network deployment. Most deployment approaches rely on cluster-based techniques, where small disjointed groups are formed and managed by a core node, called the Cluster Head (CH). This CH is elected to manage the Cluster Members (CMs) and aggregate their data to forward it to the Base Station (BS), as illustrated in Fig. 1. Dividing a network into clusters returns multiple advantages, such as: reducing the routing table size stored in each node, conserving the communication bandwidth by avoiding the exchange of redundant messages, and isolating routing changes from one cluster to another [8]. Traditional clustering algorithms were based on electing a CH based on the nodes' ID, or their location information, when frequent control broadcasting messages are used [6].

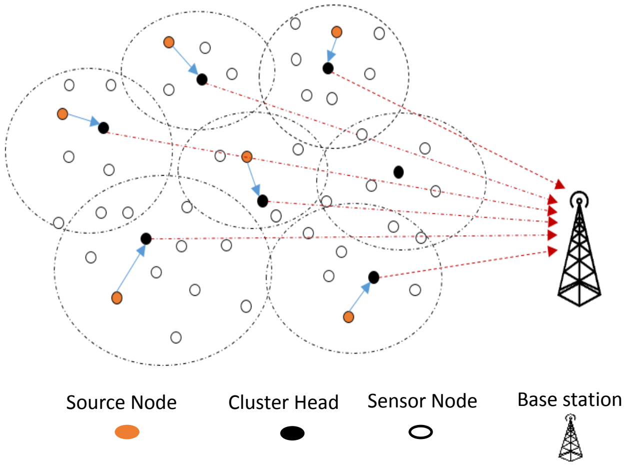


Fig 1: General system model for the cluster-based scheme

Despite these advantages, there remain many factors that affect the efficacy of clustering methods, such as: electing optimal CHs, maintaining and avoiding re-clustering of the created clusters, minimising the overheads of inter-cluster and intra-cluster communication, and prolonging network lifetimes [7]. The protocols proposed to address the above issues are not subject to these limitations. In particular, they are not influenced adversely by aspects of the nodes' characteristics such as battery life, location transmission range, etc. In order to prolong a network's lifetime, while scaling network size, we propose the introduction of a novel energy efficient protocol, named "Weight and Energy-Efficient Rotating Clustering Protocol for WSNs" (WEERC).

This paper is organised as follows. Section II reviews related work in the areas of WSNs. Section III, presents the problem statement. Section IV, details the WEERC algorithm precisely. Section V provides the simulation settings and gives the performance results. Future work and recommendations for further improvements to the proposed approach are discussed in Section VI.

II. RELATED WORK

Different clustering approaches have been proposed in the literature to address fault-tolerance [9], lifetime prolonging [8], and scalability [10] issues in WSN. The “Low Energy Adaptive Clustering Hierarchy” (LEACH) protocol is the most primitive clustering algorithm for WSNs [11].

The LEACH protocol consists of two phases: a setup-phase and a steady-phase. In the setup-phase, each node will declare itself to be a CH in the current round, according to the probability model, as defined in equation (1). By having a random number between 0 and 1, the only nodes likely to become the final CH are those that choose a random number less than the threshold $T_{(n)}$. In the steady-phase, all nodes join the closest CH to begin transmitting their data to their CH.

$$T(n) = \begin{cases} \frac{P}{1-P*\ (r \bmod \frac{1}{P})} & \text{if n } \in \text{ G} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where r is the round number, P is the probability that each node could become the CH, and G is a set of nodes, which have never taken the role of CHs in the previous $1/p$ rounds.

Despite its simplicity, the election of CHs in LEACH is relies on a probabilistic approach that does allow for any particular network deployment (density, nodal degree, etc.). Thus, there is always a possibility that the two elected CHs will be neighbours, and therefore, cover the same area. Moreover, the LEACH protocol cannot establish the preferred number of CHs elected in each round. In addition, the elected CHs could be located close to the network boundary, forcing other nodes to dissipate more energy to transfer the message to their associated CHs.

A Hybrid Energy-Efficient, Distributed protocol (HEED) was proposed in [12]. The election process in HEED draws on the residual energy in each node and the cost of intra-cluster communication. In HEED, the nodes chosen as CHs must have the highest average residual energy. Clusters in HEED are formed in three main phases: initialisation, iteration, and finalisation. The initialisation phase, according to equation 2, means that each node is assigned the probability of becoming a tentative CH.

$$CH_{prob} = C_{prob} \times \frac{E_{residual}}{E_{max}} \quad (2)$$

where C_{prob} is initial probability (i.e., a predefined value), $E_{residual}$ is residual energy, and E_{max} is the maximum energy generated by the sensor nodes. In the iteration phase, some nodes will qualify as tentative CHs. If there are two tentative CHs in the same communication range, the node with the lowest cost will become the considered CH. In the last finalisation phase, the nodes not covered by any CH (isolated nodes) will declare themselves CHs. While HEED addresses some of the limitations of the LEACH protocol, it still results in unbalanced energy distribution among the created clusters. In fact, the cluster closest to the BS is likely to be overwhelmed and expected to handle more traffic compared to those furthest from the base station. Furthermore, the large number of iterations required to form clusters creates a massive overhead.

An energy-efficient unequal clustering mechanism for wireless sensor networks EEUC protocol was proposed in [13]. EEUC aims to address the problems associated with the battery lifetime of critical CHs close to the BS. EEUC proposed a mechanism for partitioning nodes into clusters of unequal size, so the clusters near the BS would be smaller than those located furthest from it. As a result, the CHs closer to the base station could potentially preserve some of the energy for inter-cluster data forwarding. The selection of a CH in EEUC is based primarily on each node’s residual energy. However, the elected CHs near the BS is not guaranteed to have more residual energy than those far away from the BS.

The Life Time Sensitive Weighted Clustering on Wireless Sensor Networks LTS-WCA protocol was proposed in [15]. LTS-WCA enhanced the existing weighted clustering algorithm WCA for MANET, using a local minimum instead of a global minimum; thereby, allowing each node to make a decision independently, without considering the decisions or specifications of the other nodes. The nodes then communicate in a multi-hop manner. The weight of each node is calculated by combining several parameters, including remaining energy, the transmission range of the node, the number of nodes the CH can manage and the number of neighbouring nodes in the first hop. Each node in the network calculates its own parameters using equation 3:

$$S = (NK^2T^2r\ \pi)/A \quad (3)$$

where S is the ideal number of members in each cluster, K is the number of hops a node can support inside its cluster, N refers to the number of nodes that can receive transmitted packets, and A is the network field. The weight would then be calculated as described in equation 4:

$$w = (w_1T_r+w_2M_v) / (w_3d_v+w_4E_r+w_5S) \quad (4)$$

where, T_r is the transmission range of the node, M_v is the speed of the node, d_v is the number of first hop neighbours, and E_r is the residual energy of a node, $w_1+w_2=1$, $w_3+w_4+w_5=1$. Note that, the weightings in LTS-WCA are: $w_1 = 0.5$, $w_2 = 0.5$, $w_3 = 0.4$, $w_4 = 0.4$, and $w_5 = 0.2$

Unlike our algorithm, the LTS-WCA, does not consider the location of the nodes or the location of the base station (BS) when electing the cluster head. Consequently, it is possible to select A CH that is located far away from the BS. Moreover, the protocol suffers from massive overheads, as the CH has to acknowledge membership requests from each node within its cluster.

Recently, the Load-balancing Clustering Based protocol (LCP) was proposed [14]. This protocol elects the CH node with the highest energy in the first round, and then rotates the election process within the cluster itself. The LCP protocol is similar to the HEED protocol in the setup phase; however, it differs in the steady phase. Due to the new rotation phase, the re-election of the new CH occurs inside the cluster, removing the need to re-cluster the entire network. Once all the nodes inside the cluster have performed the CH task, the final remaining CH sends a message to the BS to re-cluster the network. Despite its advantages, LCP incurs a delay to network re-clustering; by preventing small clusters from reporting to the BS to re-cluster the network before the largest clusters complete their rotation cycle. Differing from other protocols, LCP also relies on the residual energy to elect the CHs. Table 1 compares and contrasts the different election processes used in the above-described protocols.

Table 1: CH selection criteria in different algorithms

Protocols	*CH election criteria*					
	Energy	Degree	BS distance	Transmission range Quality	Centrality	Multi-hop
WEERC	✓	✓	✓	✓	✓	✓
LCP	✓	×	×	×	×	×
EEUC	✓	×	✓	×	×	✓
HEED	✓	×	×	×	×	✓
DLCP	✓	×	×	×	×	✓
LTS-WCA	✓	✓	×	×	×	✓

III. PROBLEM STATEMENT

As aforementioned in the literature, existing clustering offer different methods for clustering sensor nodes [4, 9, 10, 12, 13, and 14]. However, these approaches are subject to multiple limitations when electing appropriate nodes to act as CHs and to maintain the created clusters. A number of these algorithms adopted a probabilistic selection for CH nodes, ignoring the nodes' characteristics [11].

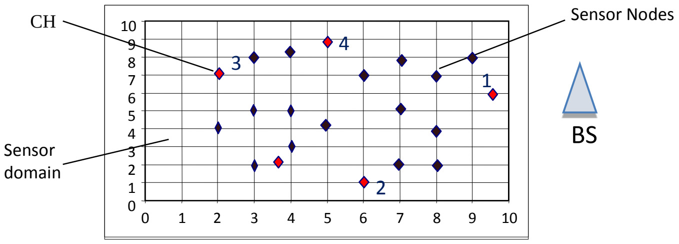


Fig 2: demonstration of the election issue

Consequently, this can lead to the election of a CH that has less energy, is heavily loaded, or has been elected previously. Giving priority to nodes with the highest energy level only is an inadequate solution to selecting a CH, particularly in random deployment [16]. The scenario demonstrated by Fig. 2 explains the issue when considering the high energy level of the node aiming to become a CH. If we assume nodes with a high energy level are red nodes, these nodes will become the CHs in this network. However, node 1 is off-centre from its neighbours and is at the edge of the sensing field. In weight-based algorithms, we found, the majority of approaches do not consider the centrality of the selected node as a CH. Therefore, in our approaches we select the final CH based on nodes weight and its centrality from its neighbours.

IV PROPOSED ALGORITHM

The proposed algorithm namely, Weight and Energy-Efficient Rotating Clustering Protocol (WEERC) aims to elect the most suitable nodes as CHs, while enhancing the network lifetime overall. The election process of CHs is mainly based on the node's weight. The weight of each node is determined using a combination of metrics, including the average number of neighbours (*degree* d_i), remaining energy (E_i) and transmission quality (*TQ*).

Once the weight is calculated, each node will broadcast a weight packet (*w-pkt*) to its neighbours. Thereafter, the nodes perform a weight comparison to select those nodes with the highest weight in the network. If the node's weight is among the highest weight in the network, it will become a CH candidate. All CH candidates will broadcast (CH *candidate-pkt*) packets; that contain node IDs and node Centrality metrics, *CN*. The nodes, that do not qualify for CH candidacy enter the sleep mode. The main aim is to select a sub-group of nodes as CH candidates, preventing others from competing to conserve their energy.

In the following section, we describe the general steps determining how the weights of the nodes are calculated.

Step 1: Neighbours Discovery

All nodes in the network broadcast a *HELLO* packet including node ID and remaining energy $E_{i.}$ The number of neighbours d_i is determined by equation 5:

$$d_i = \{v'| \, distance\ (v,v') \leq T_v\} \tag{5}$$

where d_v is the number of neighbours within its distance, and within its transmission range T_v.

Step 2: Node degree

The degree ND_i of the node is used to calculate the average number of neighbouring nodes of node *i*. Equation 6 determines the average degree d_i for each node:

$$ND_i = \frac{d_i}{Total_n} \tag{6}$$

where d_i is the number of neighbours of node *i* within its distance, $Total_n$ is the total nodes in the network.

Step 3: Transmission Quality

Transmission quality is measured to determine the connection quality between the BS and the node. Equation 7 is used:

$$TQ = \frac{BS_d}{T_v} \tag{7}$$

where *TQ* is the node's Transmission Quality, BS_d is the distance to the BS of the nodes.

Step 4: Remaining Energy

The remaining energy of node E_i is calculated, and the node with the higher remaining energy when compared with its neighbours will have more chance to become the CH. The remaining energy E_i is calculated using equation 8:

$$E_i = \frac{Ren_i}{Max_{en}} \tag{8}$$

Where *Ren$_i$* is the remaining energy of the node, and *Max$_{en}$* is maximum remaining energy among all the neighbours of node *i*.

Step 5: Node Weight

Each node in the network will employ the above criteria to define its weight. The weight is calculated using equation 9:

$$w_i = w_1 \times ND_v + w_2 \times TQ + w_3 \times E_i \qquad (9)$$

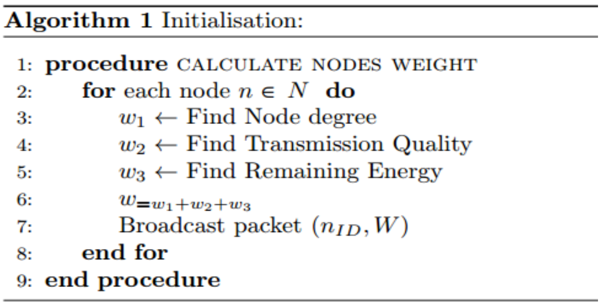
w_1, w_2 and w_3, are the weight coefficients that correspond to the system criteria, so that the sum of $w_1 + w_2 + w_3 = 1$, the coefficient values considered in this simulation are 0.2, 0.3 and 0.5 for w_1, w_2 and w_3, respectively.

Step 6: Cluster Head Candidates

Once the weight of each node is calculated, the nodes have highest weight will be selected as CH candidates. The aim of this step is to select a set of highest weight nodes to compete for final CHs and cease low weight nodes for competition process in order to preserve their energy. The election phase in section IV describes this process in more details.

IV. Implementation

A. **The initialisation phase***:* every node in the network will broadcast a *HELLO* message, to compute how many neighbours it has and their positions. Then it will broadcast its weight back along the network using a single packet



Algorithm 1 Initialisation:

```
procedure CALCULATE NODES WEIGHT
    for each node $n \in N$ do
        $w_1 \leftarrow$ Find Node degree
        $w_2 \leftarrow$ Find Transmission Quality
        $w_3 \leftarrow$ Find Remaining Energy
        $w_{=w_1+w_2+w_3}$
        Broadcast packet $(n_{ID}, W)$
    end for
end procedure
```

B. **The election phase**: Once all the nodes have broadcast their weight to their neighbouring nodes, each node will then perform a comparison with its own weight. If the node has the highest weight, it will declare itself a CH candidate. The number of CH candidates depends on the size of the network. We defined a threshold to control the number of candidates competing for in the final CH stage. The first 20 highest weights nodes will be selected to compete for final CHs in a network of 100 nodes. The main intention behind determining this threshold is to elicit all the nodes required to compete as final CHs. All the CH candidates will broadcast a compete-pkt. The compete-pkt will contain node ID, R$_{comp}$ and node degree centrality DC. We used the *R$_{comp}$* function, for *Chengfa Li* [13] with some changes in to the represented values of the d_{max} and d_{min} to the base station values. The *R$_{comp}$* is calculated according to equation 10.

$$S_i.R_{comp} = \left(1 - c\,\frac{d_{max} - d\,(S_{i,}BS)}{d_{max} - d_{min}}\right) R^0_{comp} \qquad (10)$$

where d_{max}, d_{min} represents the maximum and minimum distance to the base station, *d* (s_i, *BS)* denotes the distance between s_i and the BS, *R$_{comp}$* the maximum value of the cluster radius, and *c* is the constant coefficient between (0- 1). The Degree Centrality (*DC*) of node (*v*) is determined as in equation 11.

$$DC = \frac{d_v}{n-1} \qquad (11)$$

where d_v is the degree of the node and $_n$ is the total number of the nodes in the network. The value of *DC* ranges between 0 and 1, where 0 represent the lowest possible centrality and 1 the highest possible centrality.

The following table demonstrates an example of centrality values for a set of nodes in a network of 100 nodes, when applying equation 11. As can be seen node 3 has highest centrality value. In our algorithm if node 1, 2, 3 and 4 are CH candidates, the node 3 will become the final CH.

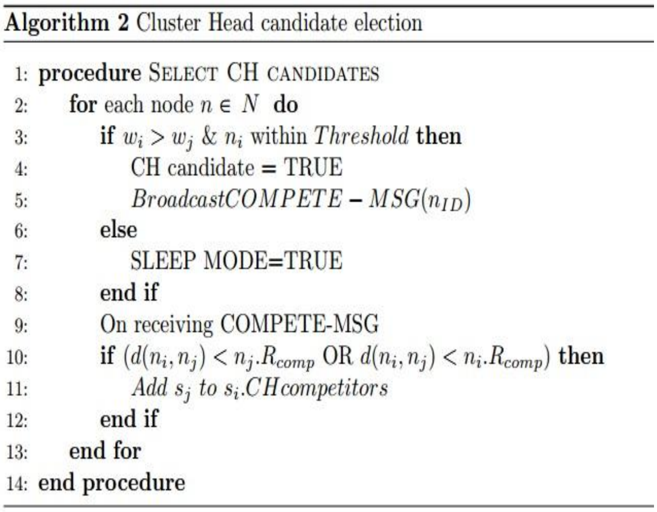
TABLE 2. DEGREE CENTERALITY

Node	Degree /n-1	Centrality
1	5/99	0.051
2	8/99	0.081
3	12/99	0.121
4	7/99	0.071

We have used Freeman's general formula for degree centrality for UCINET Software for Social Network Analysis (SNA) [17] represented in equation 12 to prove the accuracy of our formula by measuring the degree centrality of the nodes 1, 2, 3 and 4, the outputs are 0.051, 0.081, 0.121 and 0.071 respectively.

$$C_D = \frac{\sum_{i=1}^{g}[C_D(n^*)-C_D(i)]}{[(N-1)(N-2)]} \qquad (12)$$

where C_D(n*) is the degree of the node, *N* is the total number of nodes in the network.



Algorithm 2 Cluster Head candidate election

```
1:  procedure SELECT CH CANDIDATES
2:      for each node $n \in N$ do
3:          if $w_i > w_j$ & $n_i$ within $Threshold$ then
4:              CH candidate = TRUE
5:              $BroadcastCOMPETE - MSG(n_{ID})$
6:          else
7:              SLEEP MODE=TRUE
8:          end if
9:          On receiving COMPETE-MSG
10:         if $(d(n_i, n_j) < n_j.R_{comp}$ OR $d(n_i, n_j) < n_i.R_{comp})$ then
11:             $Add\ s_j\ to\ s_i.CHcompetitors$
12:         end if
13:     end for
14: end procedure
```

C. **The finalisation phase**: Once the nodes receive the compete packet (compete-pkt) from the other CH candidates, each node will use the R$_{comp}$ function as described in line 10 of algorithm 2 to check its diameter comparative to the other candidates. If there are no other candidates, the node will become the final CH, otherwise the node with the greatest degree centrality will become the final CH, and the others nodes will stop competing, even if they share the same diameter.

```
Algorithm 3 Final Cluster Head election
procedure SELECT FINAL CH
    while $CHcand = TRUE$ & $CHcompetitors! = 0$ do
        if $Centrality_i > Centrality_j$ then
            IsCH =TRUE
            Broadcast CLUSTER HEAD-MSG ($n_{ID}$)
        else
            CHcand = FALSE
            EXIT
        end if
    end while
end procedure
```

D. ***Rotation phase***: we have used the rotation mechanism of the CH inside each cluster to mitigate heavy procedures and to create clusters in the set-up phase. Once the clusters have been created in the first round, the rotation of the next CH will take place inside each cluster, by selecting the node with the highest weight from among the cluster members. If the energy of the current CH has drained below the threshold. The threshold would then be defined in this stage as shown in equation 13.

$$T_{CH} = \frac{\sum E_{Residual}}{\sum N} - 1 \qquad (13)$$

where $E_{Residual}$ is the sum of the residual energy in the cluster, and the *n* is the sum of nodes in the cluster. Then the CH will notify its member to trigger a further election. The node with the highest weight will then be declared the next CH, until all the nodes have been drained of most of their energy. In this case, the BS will then be notified to re-cluster the entire network.

```
Algorithm 4 Rotate phase
procedure SELECT CH NEXT ROUND
    Nodes Wait for interval of time
    if ($n_j En < T_{CH}$) then
        Broadcast Re-cluster message
    else
        Select Next node as CH
        Broadcast Final CH message
    end if
end procedure
```

V. PERFORMANCE EVALUATION

Simulation experiments have been conducted to examine the performance of the proposed algorithm when compared with HEED, LCP and EEUC. We contrasted the energy consumption efficiency of the network as the main performance parameter, using the Castalia simulator to implement and conduct the performance evaluation. Table 3 shows the parameters and provides the values for our experiments.

TABLE 3. SIMULATION PARAMETERS

Parameters	Values
Area	250 x 250 meters
Number of nodes	100 - 500
Base station	275 x 125
Initial energy	25 Joules
Packet size	200 bytes
Deployment	non-uniform
Radio Model	CC2420

We consider the following assumptions for evaluating WEERC against the most know clustering algorithm:

- Each node has limited energy and equal initial energy.
- Each node is capable of exchanging messages with other nodes in its transmission range.
- The network topology is static throughout the network lifetime.
- The BS is located away from deployment field.
- All the nodes are homogeneous.
- The distance between the node and the base station is measure based on the strength of the received signal.
- Maximum number of hops each node is capable to support, which is two in our assumption

Based on above network assumptions, the following three metrics are employed to evaluate network lifetime:

- First Node to Die (FND): when the first node in the network has drained its battery.
- Half Nodes Die (HND): when half of the nodes have drained their batteries.
- Last Node Dies (LND): when all the nodes have had their batteries drained.
  Moreover, we examined the average number of control packets.

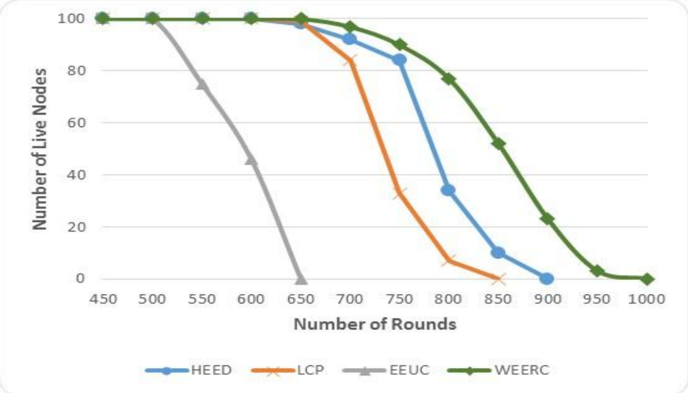


Fig 3: Number of alive nodes VS number of rounds for HEED. LCP, EEUC and WEERC

Fig. 3 represents a comparison between four protocols in a network of 100 nodes and their lifetime. It's apparent that WEERC outperforms the most efficient protocol of the other protocols by 10%. The results demonstrate that the EEUC is less efficient in terms of energy conservation, as all its nodes die rapidly than the other protocols. HEED and LCP perform better than EEUC, but their nodes died before those of WEERC protocol. The findings demonstrate that the nodes of the EEUC did not remain live after 650 rounds. The LCP did a maximum of 850 rounds, and all the nodes in HEED died within 890 rounds, whereas WEERC lost all its nodes after 990 rounds. Hence, our protocol improves network lifetime by around 100 rounds. We believe that considering multi parameters to elect the CH and selecting a set of the highest weight nodes to compete for the final CH as well as using a rotation technique inside each cluster had a great impact on prolonging the network lifetime.

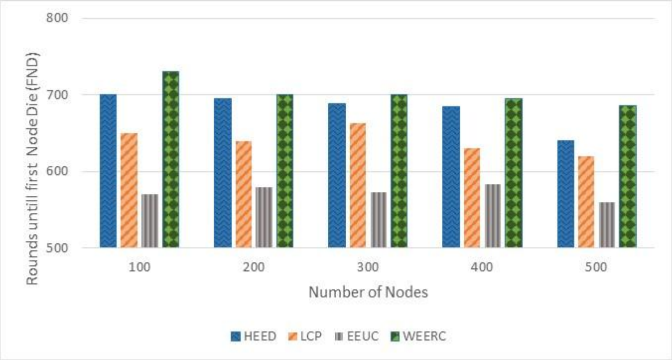


Fig 4: First node dies in HEED. LCP, EEUC and WEERC

Fig. 4 shows rounds until the first node dies FND in the four protocols tested by different network size range from 100 to 500 nodes. As the result shows the WEERC last longer than the other protocol in FND in all sets of networks. In WEERC the 100 nodes network the FND is the best compared to the other sets, where it decreases in 200, 300 and 400 sets at the approximately the same level, in the 500 network decreased farther than the other network. As the result showed WEERC last longer than the other protocol FND in all sets of networks. In comparison with the other three protocols in their best performance in networks, WEERC is more efficient by 19% than EEUC in 400 network. And more efficient by 5.6% and 4.3%, than the LCP in 300 nodes and HEED in 100 nodes respectively.

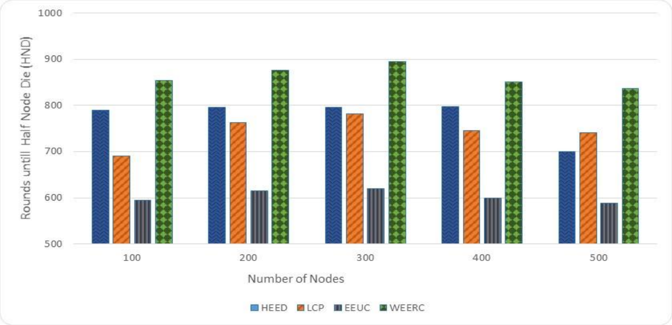


Fig 5: Half node dies in HEED. LCP, EEUC and WEERC

Fig. 5 shows the HND for the four protocols for different size of networks. As can be seen, the WEERC shows better performance over other protocols which reaches 894 rounds as its maximum in the 300 nodes network. In comparison with the other three protocols in their best performance network, WEERC is more efficient by 44% than EEUC in 300 network. And more efficient by 14% and 6.5%, than the LCP in 300 nodes and HEED in 400 nodes respectively

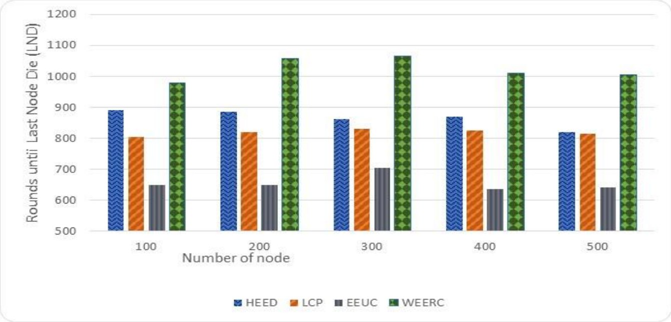


Fig 6: Last node dies in HEED. LCP, EEUC and WEERC

Fig. 6 shows round until the last node dies LND in the four protocols tested by different network size range from 100 to 500 nodes. According to the results obtained, the WEERC performs better than the other protocols in all network sizes. In WEERC the 300 and 200 nodes network has the highest performance in LND at 1058 and 1066 rounds, where it decreases in 400 and 500 node networks to just above 1000 rounds. In comparison with the other three protocols in their best performance network, WEERC is more efficient by 51.3% than EEUC in 300 network. And more efficient by 28.5% and 10.2%, than the LCP in 300 nodes and HEED in 100 nodes respectively.

.

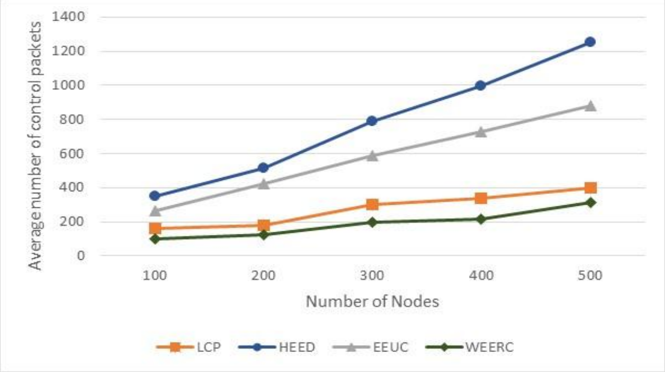


Fig 7: Average number of control packets in LCP, HEED, EEUC and WEERC

Fig.7 shows the relation between the average number of control packets and the number of nodes in different networks range from 100 to 500 nodes between LCP, HEED, EEUC and WEERC. The WEERC uses less the number of control packet against three other protocols in all networks. In comparison to the HEED in 100 nodes the average number of control packets decreased by 71% and decreased 60% in 500 nodes network.

VI. CONCLUSIONS

In this paper, we introduce a new energy efficient clustering scheme for WSNs. In our scheme, key parameters have been used for each node to guarantee the best selection of final CHs, such as, the remaining energy, transmission range and the number of neighbouring nodes as well as the degree centrality, to select the optimal CHs in WSNs. We have used a new selection of set of nodes as CH candidates in order to select the final CH and prevent the nodes with low weight to become CH and preserve their resources for transmitting tasks. Moreover, we eliminated the frequent re-clustering process in traditional clustering algorithms, which has a significant impact in energy consumption. The results have shown that the WEERC algorithm outperforms its counterparts in terms of network lifetime and scalability. WEERC used a composite metric to select a CH and routing data in a multi-hop manner. Moreover, we eliminated the re-clustering process for the entire network and used re-clustering inside formed clusters. The WEERC lowered the control packets compared to its counterparts by selecting a set of nodes as CH candidates and re-clustering inside cluster.

VII. Reference

[1] T. Arampatzis, J. Lygeros, and S. Manesis, "A Survey of Applications of Wireless Sensors and Wireless Sensor Networks," in *Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation Intelligent Control, 2005.*, 2005, no. 2158–9860, pp. 719–724.

[2] D. Christin, A. Reinhardt, P. S. Mogre, and R. Steinmetz, "Wireless Sensor Networks and the Internet of Things : Selected Challenges," *Struct. Heal. Monit.*, vol. 5970, pp. 31–33, 2009.

[3] J. Z. Abbas, *Wireless Sensor Networks*. John Wiley & Sons, Inc., Hoboken, New Jersey, 2009.

[4] I. F. Akyildiz and M. C. Vuran, *Wireless Sensor Networks*. John Wiley & Sons, Ltd, 2010.

[5] J. Al-Karaki and A. Kamal, "Routing techniques in wireless sensor networks: a survey," *Wirel. Commun. IEEE*, pp. 1–37, 2004.

[6] O. Boyinbode, H. Le, A. Mbogho, M. Takizawa, and R. Poliah, "A Survey on Clustering Algorithms for Wireless Sensor Networks," *2010 13th Int. Conf. Network-Based Inf. Syst.*, pp. 358–364, Sep. 2010.

[7] M. Abdullah and A. Ehsan, "Routing Protocols for Wireless Sensor Networks : Classifications and Challenges," *Electron. Commun. Eng. Res.*, vol. 2, no. 2014, pp. 5–15, 2014.

[8] N. Sharma and A. Nayyar, "A Comprehensive Review of Cluster Based Energy Efficient Routing Protocols for," *IJAIME 2014*, vol. 3, no. 1, pp. 441–453, 2014.

[9] J. Singh and A. K. Mishra, "Clustering Algorithms for Wireless Sensor Networks : A Review," *Comput. Sustain. Glob. Dev. (INDIACom), 2015 2nd Int. Conf.*, vol. 2, pp. 637–642, 2015.

[10] M. H. Anisi, A. H. Abdullah, S. A. Razak, and M. A. Ngadi, "An overview of data routing approaches for wireless sensor networks.," *Sensors (Basel).*, vol. 12, no. 4, pp. 3964–96, Jan. 2012.

[11] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," *Syst. Sci. 2000. Proc. 33rd Annu. Hawaii Int. Conf.*, p. 10, 2000.

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

[13] C. Li, M. Ye, G. Chen, and J. Wu, "An energy-efficient unequal clustering mechanism for wireless sensor networks," *2nd IEEE Int. Conf. Mob. Ad-hoc Sens. Syst. MASS 2005*, vol. 2005, pp. 597–604, 2005.

[14] M. Eshaftri, A. Al-Dubai, I. Romdhani, and M. B. Yassien, "A new energy efficient cluster based protocol for wireless sensor networks.," vol. 5, pp. 1209–1214, 2015.

[15] E. A. Jarchlo, "Life Time Sensitive Weighted Clustering on Wireless Sensor Networks," *Proc. 3rd Int. Conf. Sens. Networks*, pp. 41–51, 2014.

[16] W. Chang and T. Lin, "A Novel Cluster Head Reselection and Edge Sub-Clustering Lifetime Prolongation Scheme for Modern Sensor Networks," *Veh. Technol. Conf. (VTC Spring), 2015 IEEE 81st*, vol. 81, pp. 1–5, 2015.

[17] S. P. Borgatti, M. G. Everett, and L. C. Freeman, "UCINET 6 for Windows: Software for social network analysis (Version 6.102)," 2002.