Energy Efficient Clustering for Wireless Sensor Devices in Internet of Things

Diletta Cacciagrano, Rosario Culmone, Matteo Micheletti and Leonardo Mostarda

Department of Computer Science
University of Camerino, 62032 Camerino, Italy
Email: {diletta.cacciagrano, rosario.culmone, matteo.micheletti, leonardo.mostarda}@unicam.it

Abstract

A recent study predicted that in 2020 there will be 50 billion devices connected to the Internet. These devices are not only smartphones and tablets, but also *things* which are able to perform various operations, such as sensing data and actuating on the external environment. With this perspective, WSNs are highly needed in the Internet of Things (IoT) vision. Since WSN nodes are often equipped with batteries, energy efficient WSNs is an important goal to achieve. In this paper we review and compare different energy efficient clustering protocols for WSNs. We consider WSNs that are composed of heterogeneous wireless sensor devices (i.e., heterogeneous WSNs) but we also take into account protocols that incorporate various IoT devices such as RFID and energy harvesting. We describe our novel Rotating Energy Efficient Clustering for Heterogeneous Devices (REECHD) [17]. This is a novel clustering protocol for heterogeneous WSNs. REECHD is compared with the state of art clustering by using simulation.

1 Introduction

The Internet of Things (IoT) is composed of interrelated smart objects, wireless devices and people that can autonomously communicate data over the network. Different studies predict that the global IoT market will grow from \$157B in 2016 to \$457B by 2020. Transportation and logistics, smart homes, smart supply chain, smart cities, connected cars, smart industry, and smart retails are examples of applications that will benefit from the internet of things technology.

Wireless sensor networks (WSNs) play a very important role for implementing the vision of the IoT; they behave as a digital skin and implement a virtual layer where the information of the physical world can be read by the computational system. Wireless Sensor Networks (WSN) are composed of spatially distributed sensors that can autonomously collect environmental data.

Sensors can produce a large volume of data and can have heterogeneous features such as computational power, memory, and communication capabilities. WSNs are referred to as homogeneous when all the nodes are equal, for instance, they have the same hardware and the same transmission rate. A WSN which is not homogeneous is referred to as heterogeneous. Devices are usually battery-powered thus gathering data from a WSN in an energy efficient way is quite important.

Clustering is one of the energy efficient solutions that has been proposed by the research community in order to gather data from a WSN. This produces a set of clusters.

Each cluster has a set of member nodes and a cluster head (CH). This gathers data from its members (intra-cluster communication). CHs cooperate in order to report data to a centralised base station (BS) (inter-cluster communication).

In this paper we review and compare different energy efficient clustering protocols for heterogeneous WSNs. We also consider various protocols for homogeneous WSNs which has been adapted in the heterogeneous context. We describe our novel Rotating Energy Efficient Clustering for Heterogeneous Devices (REECHD) [17]. REECHD is a clustering protocols for heterogeneous WSNs that introduces a novel leader election protocol which considers the node residual energy and the node induced work. This is estimated by using the node transmission rate. REECHD also introduces the concept of intra-traffic limit rate (ITLR). This defines a limit on the intra-traffic communication that all WSN clusters must comply with. ITLR can be used to improve energy efficiency. We compare REECHD with various clustering protocols. Comparison is performed by simulating all protocols with same case study and the same assumptions. This ensures a fair comparison.

The rest of the article is organised as follows: Section 2 reviews the state of art of clustering for homogeneous and heterogeneous WSNs; Section 3 details the REECHD election and its novel contribution as well as the algorithm for cluster formation; Section 4 describes the network model and the simulation results; finally, Section 5 concludes the article .

2 State of Art on Clustering for WSNs

A great deal of literature and research articles are available on clustering protocols. In this section, we focus on existing prominent clustering protocols for homogeneous and heterogeneous WSNs. We consider clustering approaches having equal and unequal size clustering, rotation and non rotation, single hop and multi-hop. We conclude the section with clustering protocols that consider harvesting and IoT devices, and protocols which are based on machine learning.

2.1 Clustering protocols for homogeneous WSNs

Low Energy Adaptive Clustering Hierarchy (LEACH) [8] is one of the pioneering routing protocols that introduced the idea of clustering into the field of WSNs. Unlike most of the clustering protocols, which use the node residual energy for cluster election, LEACH uses a probabilistic function. All cluster heads can directly communicate with BS, i.e., multi-hop communication never takes place. Once a node has been elected as a CH it cannot take the same role in the next cluster election. LEACH proposes a randomised rotation of CHs and data aggregation at each CH.

HEED [25] clustering protocol produces clusters of equal size, i.e., each cluster has the same radius. The HEED algorithm is composed of the following two phases: (i) clustering, and (ii) network operation. During the clustering phase, CHs get elected based on the residual energy, and member nodes join the closest CH[1]. During the network operation phase data messages get delivered from the members to the BS. Clustering and

[1] Communication costs can be considered to elect or join a CH.

network operation phases are repeated over time. HEED generally prevents two nodes within the same transmission range from becoming CHs. As reported in [25], sensor nodes close to the BS deplete their energy faster with respect to nodes that are farther away. This problem is referred to as hot spot problem. In fact, while all CHs will have the same amount of average intra-traffic communication (i.e., the traffic inside a cluster) CHs close to the BS have a higher inter-cluster communication (i.e., relay traffic amongst CHs).

Distributed Weight-based Energy-efficient Hierarchical Clustering protocol (DWEHC) [4] is an equal size clustering based protocol for WSNs. It optimises intra-cluster communication by introducing multi-hop transmission within the clusters. All sensor nodes execute DWEHC individually to decide whether to be a cluster head or a member node. DWEHC clustering formation phase is based on HEED topology. Resultant clusters arrangement is well-balanced and leads to enhance network lifetime.

Voluminous literature have been developed on devising energy efficient unequal size clustering protocols for WSNs.

Energy Efficient Unequal Clustering (EEUC) algorithm [15] for WSNs is one of the first approach that had been conceived. EEUC is based on the idea that a larger cluster size should be used when the CH resides in zones farthest from BS whereas zones nearest to BS should be populated with a considerable amount of smaller clusters. This approach would minimise excessive overhead burden on cluster heads nearest to BS and should alleviate the energy hole or hot spot problem.

Unequal clustering algorithm based on HEED (UHEED) [7] is an unequal size clustering based protocol for WSNs. UHEED incorporates the idea of EEUC protocol into HEED in order to build unequal size clusters. The size of a cluster CH depends on its distance from the BS. The farther away CH is from the BS, the larger its competition radius is. In other words, clusters that are farther away from the BS have a larger radius with respect to clusters nearer to the BS. UHEED reduces the hot spot problem and increases network lifetime when compared to HEED and LEACH.

Rotated Unequal HEED (RUHEED) [1] uses an unequal size clustering based approach that not only improves the hot spot problem but also enhances the network lifetime. RUHEED is composed of three stages that are CH election; clusters formation; and CH rotation. HEED is used to elect CHs based on its residual energy and communication cost. EEUC concept, which is based on the sensor node distance from the BS, is used in order to establish unequal sized clusters. During CH rotation phase, current CH selects the member nodes with the highest energy and directly designates it as the next cluster head. Rotation strategy avoids re-clustering of the network thus network lifetime is improved. Re-clustering of the network takes place when any of the sensor nodes drain its entire energy. RUHEED preserves energy and minimises the number of cluster election and cluster formation phases.

ER-HEED [23] is a clustering protocol that enhances performance of HEED by introducing CHs role rotation inside clusters. ER-HEED is composed of three stages that are cluster head election, cluster formation using HEED and cluster head rotation. Like RUHEED, CHs nominate the next CHs that have the highest residual energies. This concept of CH selection within the cluster member nodes reduces the number of cluster elections. HEED based cluster head election is performed only when any of the sensor nodes depletes its energy completely. ER-HEED performance in terms of first node dies measure criteria is far superior to RUHEED, HEED and UHEED.

2.2 Clustering protocols for Heterogeneous WSNs

While WSNs have homogeneous nodes, heterogeneous WSNs introduce nodes that can have differences in the following features: (i) energy level; (ii) data rate; (iii) transmission range; (iv) aggregation performance; (v) processing capabilities. Heterogeneity affects significantly the network lifetime and lessens network response time [24]. In this section we describe various clustering algorithms that have been devised for heterogeneous wireless sensor networks. Different protocols can make different assumptions about the heterogeneity of the WSNs.

DEEC (distributed energy-efficient clustering algorithm for heterogeneous WSNs) [19] is an equal size clustering protocol. DEEC cluster head election is based on a probability that is calculated by considering the ratio of the residual sensor node energy and the network average energy. The CH role is rotated among sensor nodes on the basis of their residual energies. This ensures a uniform energy consumption over the entire network. Sensor nodes that have the highest residual and highest initial energies will be more likely selected as cluster heads. BS broadcasts the network average energy information to all wireless sensor network nodes.

Distributed energy balance clustering Protocol for heterogeneous WSNs (DEBC) [5] is a clustering protocol for heterogeneous WSNs. DEBC assumes that sensor nodes have heterogeneous energy levels. The cluster head election is based on the sensor node residual energy. Sensor nodes that have the highest initial energy and the highest residual energies are highly probable to be selected as cluster heads. The simulation results shows that the performance of DEBC is superior to LEACH and SEP[22].

The authors in [10] describe a distributed clustering with load balancing (DBLC) for forming cluster efficiently and balancing load in inter-cluster communication. Size (range) is important in terms of energy efficiency and balancing load in multi-hop communication of CHs. This avoids energy inefficiency and produces balanced load of cluster. Blanced inter-traffic communication is achieved by using clusters with different sizes at each step.

The autors in [12] proposed a distributed CH election approach for heterogeneous WSNs. The election of cluster heads is based on a weighted probability. Member nodes communicate with their CH and then CHs communicate the aggregated information to the base station. Three different types of nodes are considered and all have different threshold. The weight assigned to each node will decide the selection of cluster head for each type.

Energy efficient heterogeneous clustered scheme for wireless sensor networks (EEHC) [11] is a clustering protocol for heterogeneous WSNs. In EEHC, a percentage of sensor nodes are equipped with various levels of battery capacity. EEHC aims at enhancing network efficiency and reliability. Like DEEC and DEBC, the cluster head election probability of EEHC depends on sensor node residual energies.

A stable election protocol for clustered heterogeneous wireless sensor networks (SEP) [22] is a heterogeneous protocol and intents to enhance network lifetime according to the first node dies network lifetime measure. SEP assumes two different types of nodes that are normal and advanced sensor nodes. CH election is based on sensor node initial and residual energies. Simulation results show that SEP prolongs network lifetime and average throughput.

FMUC (feedback mechanismbased unequal clustering) [16] is a feedback mechanism based unequal heterogeneous protocol. FMUC is specifically designed to avoid the energy hole problem when balancing the energy load in application-based WSNs. Initially, FMUC divides the network into layers which are computed analytically. A mathematical model is used in order to uniform the ratio of the energy consumption and the total initial energy of each layer. Each cluster will belong to the one of the layers. The size of each cluster is calculated by considering the ratio of the energy consumption of each layer. Clusters send their sizes as a feedback to the sink which broadcasts the collected values into the network. All nodes of the WSN receives the feedback values but only the cluster heads change their competition radius according to received values.

2.3 Clustering protocols with harvesting

Harvesting is the capability of sensor nodes to be able to harvest energy either from a dedicated or an opportunistic ambient source such as solar, thermal, wind and vibration.

The authors in [20] give a detailed list of various sources that can be used to harvest energy in WNSs. They can be categorised into ambient sources and external sources.

Ambient sources are:

- Radio Frequency-based energy harvesting where RF-based energy harvesting received radio waves are converted to DC power after conditioning;
- Solar-based energy harvesting where solar energy is an affordable and clean energy source useful to eliminate the energy problem in WSNs. The photo-voltaic effect converts solar rays into DC power when certain semiconductor materials are exposed to sunlight. We remark that it is not possible to perform solar energy harvesting during the night. Thus, developers have to ensure the highest possible efficiency during daylight ours to guarantee the viability of solar-power;
- Thermal-based Energy Harvesting where it is possible to convert heat energy into electrical energy exploiting the seebeck effect . This requires a load to be attached across the heated and cold faces of a ThermoElectric Generator(TEG) for thermal energy harvesting. This can be done at different scales, from large to small. In WSNs, we often need to keep the scale as small as possible. In this scenario, for instance, it can be interesting to generate power from human body temperatures;
- Flow-based energy harvesting generally uses turbines and rotors to convert rotational movement into electrical energy using electromagnetic induction principal.

External sources are:

- Mechanical-based energy harvesting that is performed with sources such as vibrations, pressure,and stressstrain. To do this, a suitable Mechanical-to-Electrical Energy Generator(MEEG) is needed. A MEEG uses either electromagnetic, electrostatic, or piezoelectric mechanisms to harvest energy.

– Human-based energy harvesting that is used in Wireless Body Area Network (WBAN). In these networks, sensor nodes are deployed on or inside of the human body to monitor physiological parameters continuously. These nodes need to be operational for long periods of time or even for the lifetime of the humans being monitored. Human-based energy harvesting can be categorized as activity based harvesters and inherent physiological parameters based harvesters. More precisely, energy can be harvested from humans in several ways, such as through locomotion, changes in finger position, body heat, and blood flow. Nevertheless, the main challenge still is to miniaturise sensors to make them easier for human adoption.

In [3] the following different energy harvesting approaches are investigated: (i) energy harvesting combined with simultaneous data decoding (a trade-off between the amount of energy that can be stored for future use and the amount of energy that should be spent for signal decoding); (ii) energy efficient operation of wireless sensor networks, making use of appropriate routing schemes or scheduled operation of sensor nodes; (iii) mobile chargers which stop at optimal locations to perform charging; and (iv) energy sharing. The approaches (iii) and (iv) are chosen in a 2 step protocol combining a mobile charger which moves inside the network toward the next discharged CH to overcharge it and an energy trading between overcharged CHs and other nodes.The energy trading takes place inside each cluster: a CH is chosen getting the node which has the highest number of neighbours inside its inclusion circle. In the first stage, the mobile charger follows the optimal path to reach the next discharged CH then it stops (the mobile charger can decide to stop or move any time a time interval is passed) and it overcharges it. Then overcharged CHs sell their energy to theirs cluster members with no competition (the number of seller nodes is significantly larger than the number of buyer nodes thanks to the first stage). It is not CHs which offer the energy, but the nodes that broadcast a message (composed of an ID and the amount of energy needed) and wait for one or more CHs to give energy to them. CHs will serve near nodes first. Simulations produced with Omnet++ and Castalia shows that EH-WSN protocol works best with greater nodes inclusion circle radius than smaller ones.

The authors in [18] uses a cross-layer cooperative TDMA scheme instead of a classical one to optimise the CHs relaying performance. The CH role is alternated between the nodes using duty cycling as a function of their individual energy harvesting capabilities. This protocol define the optimal number of clusters according to the intensity of the energy source (which is solar energy in the paper). The protocol is based on LEACH. The CH choice is based on a probability function that uses duty cycle mechanism where a node cannot become CH before n duty cycles are passed. This number is computed for each CH as the ratio of the CH required energy to its allocated energy rounded to the next integer value. The protocol can include cooperation (cooperative transmission protocol) ECO-LEACH or not (ENCO-LEACH). The cooperative transmission protocol makes use of the energy unconsumed in data transmission to relay undelivered packets from cluster members to CHs and also from CHs to the sink node.

2.4 Clustering protocols with Machine learning

Machine learning (ML) is a late 1950's technique for artificial intelligence (AI) and for the definition of computationally viable and robust algorithms. During the years, ML has been applied to different fields such as bioinformatics, speech recognition, spam detection, computer vision, fraud detection and advertising networks. ML learning techniques have been used for many tasks like classification, regression and density estimation.

From the point of view of [6] and [13] machine learning can be defined as:

- The development of computer models for learning processes that provide solutions to the problem of knowledge acquisition and enhance the performance of developed systems.
- The adoption of computational methods for improving machine performance by detecting and describing consistencies and patterns in training data.

Applying ML to the field on WSNs routing protocols is a process which has both pros and cons. Some of the ML algorithms best properties are their ability to automatically calibrate according to newly acquired knowledge, their generally low complexity and their capability to uncover correlation between sensor data and improve sensor deployment for maximum data coverage. On the other hand, ML algorithms drawbacks lie in the high amount of computational power they need, which escalates when requiring more accuracy, and the large set of existing data and samples they require to achieve high generalisation capabilities.

There are several ML techniques that can be applied to WSNs to perform clustering. These techniques try to improve node clustering and data aggregation mainly in two ways:

- compress data locally at CHs by efficiently extracting similarity and dissimilarity (e.g., from faulty nodes) in different sensors readings.
- CHs election, where appropriate cluster head selection will significantly reduce energy consumption and enhance the networks lifetime.

Some classic ML approaches have been investigated in the past to check their suitability for WSNs clustering and data aggregation [2]. Clustering can be performed basing on (i) neural networks, (ii) decision trees and (iii) role-free CHs selection while data can be aggregated using (i) self-organizing map (SOM), (ii) learning vector quantization, (iii) principal component analysis, (iv) k-means algorithm and (v) decentralized learning. However, very few clustering protocols have been implemented through the approaches listed above. Moreover, most of them do not compare their results with well known adaptive clustering protocols such as HEED, ER-HEED and LEACH. One ML based protocol which makes a comparison with other existing protocols is LEACH GA[14]. LEACH GA is a genetic algorithm based on LEACH [9]. LEACH GA modifies the LEACH algorithm, adding a preparation phase only once before the set-up phase of the first round. Initially, nodes perform cluster head selection, then they send their messages stating if they candidate to become cluster head, their node IDs, and their geographical positions to the base station. At that point, the base station uses data received from nodes to determine the optimal probability p_{opt} by performing GA operations, then it broadcasts this value to all nodes. The following set-up and steady-state phases are performed in every round and are the same as LEACH. Recently, [21] has proposed a comparison of LEACH GA performance over LEACH and LEACH-C using MATLAB simulation tool. In the simulations, nodes are randomly distributed in an area of 100m x 100m with the base station located at the centre point (50, 50). According to the simulation results, LEACH-GA performs better when compared to LEACH and LEACH-C under different initial energy and probability thresholds. In particular, LEACH-GA increases the network lifetime on the average of 54% and 47% over LEACH and LEACH-C. However, simulation results do not take into account novel clustering protocols proposed after LEACH and its variations [25][23][17], which are proved to perform better under various situations.

Table 1 shows a categorisation of the clustering protocols that are described in this section. Clutering protocols are categorised by considering several attributes.

Table 1. Comparison of well known Clustering Protocols for WSNs

Protocols	**Node Deployment**	**Heterogeneous**	**Homogeneous**	**Clustering Method**	Distributed(D)	Centralized(C)	**Equal Sized Clusters**	**Unequal Sized Clusters**	**Rotation**	**Location Awareness**	**Harvesting**	**Machine Learning**	**Probability**	**No Probability**
LEACH[9]	Random		✓		D		✓			N			✓	
SEP[22]	Random	✓			D		✓			N			✓	
HEED[25]	Random		✓		D		✓			N			✓	
ERHEED[23]	Random		✓		D		✓		✓	N			✓	
UHEED[7]	Random		✓		D			✓		N			✓	
LEACH-GA[14]	Random		✓		D					N		✓	✓	
RUHEED[1]	Random		✓		D			✓	✓	N			✓	
DEEC[19]	Random	✓			D		✓			N			✓	
DWEHC[4]	Random		✓		D		✓			N			✓	
IEEUC[15]	Random		✓		D			✓		N			✓	
DEBC[5]	Random	✓			D		✓			N			✓	
DCLB[10]	Random		✓		C		✓			Y				✓
EEHC[11]	Random		✓		D		✓			N			✓	
DCHE[12]	Uniform	✓			D					N			✓	
FMUC[16]	Random	✓			D			✓		N			✓	
REECHD[17]	Random	✓			D		✓		✓	N			✓	
EHWSN[3]	Random				D		✓			N	✓		✓	

3 REECHD clustering protocol

In this section we describe the leader election novelty introduced by REECHD as well as the REECHD cluster formation and rotation algorithms.

3.1 REECHD leader election probability

REECHD is a clustering algorithm for heterogeneous WSNs that produces clusters of equal size. REECHD reduces the amount of leader election phases by using rotation. This decreases the amount of overhead messages thus prolonging the WSN lifetime. The novelty of REECHD is in its probabilistic election process and the use of the intra-traffic limit.

$$CH_{prob} = max(\frac{C_{prob}}{K}(\frac{E_{residual}}{E_{max}} + IW^{-1}), P_{min}) \quad (1)$$

The equation (1) defines the leader election probability CH_{prob}. This is the probability a node has of becoming CH when a new leader election phase takes place. In the following we summarise the components of the probability CH_{prob}:

- C_{prob} is a predefined initial probability (e.g., 5%) that sets the initial percentage of cluster heads among all WSN nodes. This is used to limit the initial CH announcements, and does not impact on the final clustering.
- P_{min} defines a minimum probability value that CH_{prob} must have. This is selected to be inversely proportional to E_{max} (e.g., 10^{-4}) so that the algorithm terminates in $N_{iter} = O(1)$ iterations [25].
- $E_{residual}$ defines the residual energy of the node while E_{max} defines the maximum energy of the node (it defines a fully charged battery)
- The constant k is chosen in order to ensure the probability CH_{prob} is always between 0 and 1. In our case k is equal to two.
- The positive quantity IW is the node induced work rate. This estimates the energy the node spends and induces on other nodes when it plays the CH role. Thus, a node with higher induced work should have less probability to be elected.

$$IW = \frac{D_{Rmax}}{D_R} \quad (2)$$

In this paper we estimate the node induced work IW by using the equation (2) where D_R is the average transmission rate of the node and D_{Rmax} is rate of the node with the highest transmission rate of the WSN. This equation assigns a lower induced work to nodes with higher transmission rate, that is, nodes with higher rate should have a higher probability of becoming cluster head. In fact, when a node with a high rate is not selected as CH (it is a member node), more intra-traffic communication is generated. On the other hand, when a node n with high rate is selected as as CH, the cluster will be not overloaded with messages from n. Nodes with lower transmission rate should have less probability of becoming cluster head since they generate little intra-traffic communication inside the cluster. It is worth mentioning that the node induced work could be further refined by considering further sources of energy consumption such as the energy the node spends to run the sensor hardware or the inter-traffic the node can potentially generates.

We emphasise that election probability of equation (1) combines together energy and induced work together. More precisely, nodes with higher energy and higher transmission rate should have more probability of becoming cluster head.

3.2 REECHD intra-traffic rate limit

The intra-traffic rate limit (ITRL) defines a rate that each CH must use during cluster formation. More precisely, each CH must ensure that the sum of transmission rates of its member nodes never exceed ITRL. This is defined by the following equation:

$$\sum_{i=1}^{|member\_set|} sending\_rate(n_i) < ITRL$$

where $member\_set$ contains all member nodes that compose the cluster, $|member\_set|$ is the cardinality of $member\_set$, n_i is a node that belongs to $member\_set$ and $sending\_rate(n_i)$ is the transmission rate of the node n_i. We can define a lower and upper bound for the ITLR:

$$\left[0, \sum_{i=1}^{|WSN\_nodes|} sending\_rate(n_i)\right]$$

where $|WSN\_nodes|$ is the number of WSN nodes. We have a flat routing (i.e., each node of the WSN is cluster head and has no member nodes) when the ITLR is equal to zero. We can have a single cluster that contains all nodes when the ITRL is the sum of all node sending rates.

The ITLR is a quite useful means to control the number of clusters inside the WSN. Low ITLR values can generate more clusters than high ITLR values. More clusters can lead to lower intra-traffic communication at the cost of higher inter-traffic communication. As we see in Section 4.2 the choice of the ITLR depends on the aggregation rate. We emphases that the use of the ITLR is also useful when nodes are not uniformly deployed since denser area can get a higher number of clusters. This allows the balance the intra-cluster communication thus balancing the energy consumption and prolonging the WSN lifetime.

3.3 REECHD algorithm

REECHD is a clustering algorithm for heterogeneous WSNs that produces clusters of equal size and uses rotation in order to prolong the WSN lifetime. Member nodes of a cluster can directly communicate with their CH. This is referred to as 1-hop communication [25]. REECHD includes the following four main phases: (i) cluster head election; (ii) cluster formation and iteration; (iii) rotation; and (iv) network operation. Cluster head election, formation and iteration are performed at the beginning and anytime a node dies. When no node dies the rotation and network operation phases are performed in alternation. All REECHD phases are described in details in the following.

3.4 REECHD cluster head election

This phase takes place at the beginning and anytime a node dies. In this phase each node can become cluster head according to the probability that is defined by equation 1 of Section 3.1.

```
1  Initialisation ()
2    iterations = 0
3    max_iterations = n
4    set_parameter(ITLR)
5
6  Cluster_head_election()
7    cluster_head_set = tentative_CH_set = ∅
8    neighbours = all neighbour nodes which are alive
9    CH_prob = max(0.5 * C_prob * (E_residual/E_max + D_R/DR_max), P_min)
10   iterations = iterations + 1
11
12   Repeat
13     if(tentative_CH_set ≠ ∅)
14       CH = least_cost(neighbours)
15       if(CH = myself)
16         if(CH_prob = 1)
17           broadcast_election_msg(neighbours)
18           add_to(final_CH_set)
19         else
20           broadcast_tentative_msg(neighbours)
21           add_to(tentative_CH_set)
22     else if(CH_prob = 1)
23       broadcast_election_msg(neighbours)
24       add_to(final_CH_set)
25     else if(CH_prob >= random(0,1))f
26       broadcast_tentative_msg(neighbours)
27       add_to(tentative_CH_set)
28     previous_prob = CH_prob
29     CH_prob = min(CH_prob * 2, 1)
30   Until previous_prob = 1
```

Fig. 1. REECHD CH election at node B

Figure 1 outlines the cluster head election algorithm that a node B executes. An $initialisation$ procedure is used to set some variables and is executed only once before the $cluster\_head\_election$ procedure. The $initialisation$ procedure sets the following variables: (i) $iterations$; (ii) $cluster\_head\_election$; and (iii) $max\_iterations$. The variable $iterations$ stores the number of times the member node B tried to perform the $cluster\_head\_election$ procedure. B needs to redo the cluster election when it is unable to join any CHs. This happens when CHs in the neighbourhood of B reached the intra-traffic limit. In this case B repeats the $cluster\_head\_election$ procedure. The variable $max\_iterations = n$ defines the maximum amount of times B repeats the $cluster\_head\_election$ procedure before it elects itself as cluster head. The procedure $set\_parameter(ITLR)$ sets the intra-traffic limit ITLR to a predefined constant value.

The $cluster\_head\_election$ procedure initialises the set $cluster\_head\_set$ and $tentative\_CH\_set$ to empty. The set $cluster\_head\_set$ contains all nodes in the neighbourhood of B that proposed as CH. The set $tentative\_CH\_set$ contains all nodes that are attempting of becoming CH but their election is not finalised yet. The variable $neighbours$ contains all nodes that are within the radius range of B and are alive. The $cluster\_head\_election$ procedure sets its probability of becoming CH (line 9 of the algorithm in Figure 1), increases the $iterations$ counter and start the $repeat$ loop (line $12 - 30$).

B selects the least cost CH [2] from $tentative\_CH\_set$ when this set is not empty (line 13) . When the selected CH is the node itself it can broadcast either an election message or a tentative message. The election message is broadcast when CH_{prob} has reached 1 while the tentative message when CH_{prob} is less than 1. When no nodes proposed as CH and CH_{prob} is equal to one (lines $28-30$) B proposes itself as cluster head. When no nodes proposed as CH and CH_{prob} is less than one (lines $25-27$) B decides whether or not to become tentative CH by considering its probability CH_{prob}; at the end of each repeat cycle the probability is doubled (line 29). This ensures that REECHD terminates in $\mathcal{O}(1)$ number of steps.

We emphasise that the $least\_cost$ function is used to break the tie and select a cluster head when two tentative nodes lies within the same communication range. This behaviour prevents two nodes within the same transmission range from becoming CHs that is REECHD creates a set of disjoint clusters.

```
1  Cluster_formation()
2
3    if(myself ∈ final_CH_set)
4      broadcast_election_msg(neighbours)
5      member_set = member_selection(join_set, ITRL)
6      send(member_set, join)
7      send(join_set − member_set, unjoin)
8    else
9      while(final_CH_set ≠ ∅)
10       CH = least_cost(final_CH_set)
11       final_CH_set = final_CH_set − CH
12       join(CH)
13       if(join_msg_received)
14         return
15     end while
16
17     if(iterations ≤ max_iterations)
18       Cluster_head_election()
19     else
20       broadcast_election_msg(neighbours)
```

Fig. 2. REECHD cluster formation at node B

Cluster formation and iteration In Figure 2 we detail the cluster formation algorithm that is executed by the node B. A node playing the CH role executes the $then$ branch of the if control structure (lines $3-8$) while a member node executes the $else$ branch (lines $9-30$):

– B **playing the CH role.** B sends a broadcast election message. A member node can reply with a join message when B is the least cost CH it can reach. The node B keeps all join requests in the set $join\_set$. This is used together with the $ITRL$ in order to call the $member\_selection$ procedure. This returns a node set

[2] We have experimented various cost functions such as selecting the closest CH or selecting the CH which has the largest member-set. In this paper a node select the closest CH

$member\_set$ that contains all nodes B selected as cluster members. We recall that the intra-traffic communication generated by the $member\_set$ nodes must be less than the $ITRL$ (see section3.2 for details). Various member selection strategies can be adopted. For instance a random pick can be performed until $ITLR$ is reached (this is the strategy we used in the presented simulation results). The member nodes can be selected starting from the highest rate node until the $ITLR$ is reached. The member nodes can be selected so that the total rate is less than or equal to $ITLR$ and is as large as possible. B sends an $unjoin$ message to all nodes that are not included in the $member\_set$ (line 8).

– B **playing the member role.** B tries to join one after another all reachable CHs (from the $least\_cost$ to the $worst\_cost$). B will join the first CH that replies with a join message (lines $10-16$). After the join the nodes terminate the cluster formation procedure.

B interates the cluster head election when it is not CH and was not able to join any cluster. The cluster head election can be repeated a maximum number of times (i.e., $max\_iterations$).

Cluster rotation The current CH designates the next CH directly by using the equation (1) [3]. More precisely the current CH calculates the probability CH_{prob} of each member node and chooses the one with the highest CH_{prob} as the next CH. The new CH is elected without the need of performing any election protocol. We refer to as *operation phase* the one where member nodes send data to their CHs. These cooperate in order to report data to the base station.

4 Comparing the state of art clustering protocols

In this section we compare REECHD with various clustering protocols. Comparison is performed by simulating all protocols by using the same WSN features, and the same network and communication models.

4.1 Network Model

In our network model nodes are not mobile and are uniformly distributed in a two dimensional area. We have energy heterogeneity since nodes can have different initial energy. Nodes have different data transmission rates within a defined maximum and minimum rate. Nodes have the same processing and aggregation capabilities. Nodes have a unique IDs and nodes can transmit at various power levels depending on the distance of the receiver.

The BS is not mobile, has no energy constraints and is located outside the WSN area. The BS has more communication and processing capabilities with respect to normal sensor nodes. Each CH can aggregate the intra-traffic data in order to reduce the

[3] It is assumed that each data packet received by the CH contains energy information of its member nodes. This is needed in order to calculate CH_{prob}.

amount of bits that are forwarded to the BS. Inter-traffic is not aggregated that is a CH forwards (towards the BS) messages received from other CHs with no aggregation.

We use a network operation model that has been adopted in quite a few papers such as LEACH, HEED, RUHEED, FMUC and so on. We recall that a clustering protocol usually includes the following phases: (i) cluster election and formation; (ii) network operation phase; (iii) rotation (if any); (iv) re-election and formation. During the data network operation phase a TDMA is composed of the following two activities: (i) each member node sends one variable size message to its cluster head; (ii) all CHs data reaches the BS. In other words, a TDMA starts from the collection of data from the member nodes and ends when all the data reaches the base station. A round is composed of multiple TDMAs.

We define two types of WSN nodes that are homogeneous and heterogeneous. Homogeneous nodes have an initial energy of 0.5 joules and send messages of 1000 bits. Heterogeneous nodes have an initial energy that falls within the interval $[0.2, 0.8]$ joules and send messages of a size that falls within the interval $[100, 1900]$ bits. We define the heterogeneity level as the ratio between the number of the heterogeneous nodes and all WSN nodes. For instance an heterogeneous level of 20% means that 20% of the WSN nodes are heterogeneous. Table 2 summarises all network parameters.

For simulation purposes we define the aggregation rate (AR) which is a number between 0 and 1. This is is used to calculate the inter-traffic message size that is generated by the CH as follows:

$$MIN(\sum_{i=1}^{|cluster\_set|} sending\_rate(n_i) * (1 - AR), min\_msg\_size) \tag{3}$$

where $cluster\_set$ is the set of nodes that compose a cluster (including the CH), $|cluster\_set|$ is its cardinality, n_i is a node that belongs to $cluster\_set$, $sending\_rate(n_i)$ is the transmission rate of the node n_i, AR is the aggregation rate and min_msg_size is a constant that denotes the minimum size of message that is forwarded by a CH. When the aggregation rate (AR) is zero a CH packs all messages received by the members (during a TDMA) and forwards them to the next hope. In this case no aggregation takes place. When the aggregation rate is 1 the CH aggregates all messages received by the members in a TDMA by producing a message with a minimum size. In this paper we set this minimum size to 100, that is the minimum rate of a node. A more refined min_msg_size value could consider the node with the smaller (greater) rate inside the cluster or the average rate of the cluster.

The adopted radio model utilises free space and multi path channel model. The assumed network grid size is 100 by 100 meters and BS is placed at position (175, 50). The simulation parameters are outlined in Table 2. Transceiver circuitry of a sensor node consumes $E_{elec} = 50nJ/bit$. Sensor node amplification energy E_a depends on the distance between sender and receiver. When $d < d_0 = 75m$, E_a becomes $E_{fs} = 10pJ/bit/m^2$ and when $d \geq d_0 = 75m$, E_a reduces to $E_{mf} = 0.0013pJ/bit/m^4$. The transmission and reception energy consumed in sending and receiving a data packet k (bits) over distance d, can be be computed [8] as:

$$E_{Tx} = k(E_{elec} + E_a d^n) \tag{4}$$

$$E_{Rx} = k(E_{elec}) \tag{5}$$

Table 2 summarises all network parameters.

Table 2. Simulation parameters

Simulation parameters	
Parameters	Values
Network grid	$From(0,0)to(100,100)$
BS	$(175, 50)$
E_{elec}	$50nJ/bit$
E_{fs}	$10pJ/bit/m^2$
E_{mp}	$0.0013pJ/bit/m^4$
R_0	$30m, 35m, 40m, 45m, 50m$
Control parameter UHEED	$c = 0.5$
Number of nodes	200

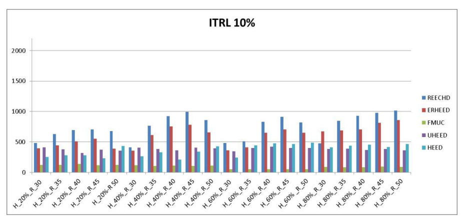

Fig. 3. Lifetime measure=FND; aggregation=50%; Heterogeneity level= 20%,40%,60%,80%; radiuses= 30m,35m,40m,45m,50m; Intra Traffic Limit Rate= 10%

4.2 Simulation Results and Analysis

We simulated REECHD, UHEED, HEED, ERHEED and FMUC on a WSN composed of 200 nodes and with a grid size of 100 by 100 meters. The heterogeneity level varied from 20% to 80% with a step of 20, the node competition radius R_0 from 30 to 50 meters, and we set the aggregation to 50%. For REECHD we also set ITLR percentage by multiplying the maximum ITLR value by a number between zero and one. We have used an ITLR percentage of 0.1,0.2,0.5 and 0.8. Each simulation is an average of hundred runs.

Figure 3,4,5 and 6 shows the lifetime of the network for different heterogeneity levels until first node dies (FND)) for REECHD, ERHEED, FMUC, UHEED and HEED protocols. These are run for an increasing heterogeneity level, and radius from 30m to 50m.

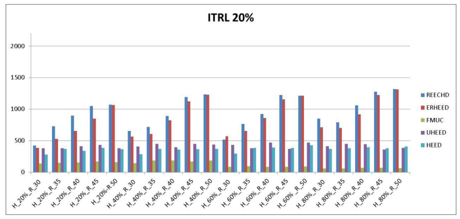

Fig. 4. Lifetime measure=FND; aggregation=50%; Heterogeneity level= 20%,40%,60%,80%; radiuses= 30m,35m,40m,45m,50m; Intra Traffic Limit Rate= 20%

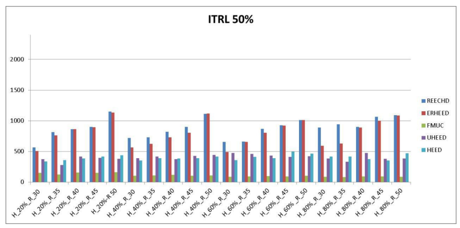

Fig. 5. Lifetime measure=FND; aggregation=50%; Heterogeneity level= 20%,40%,60%,80%; radiuses= 30m,35m,40m,45m,50m; Intra Traffic Limit Rate= 50%

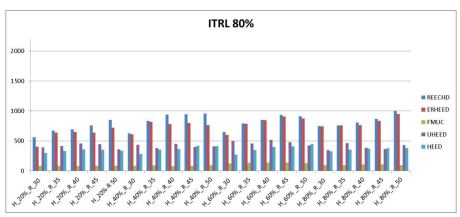

Fig. 6. Lifetime measure=FND; aggregation=50%; Heterogeneity level= 20%,40%,60%,80%; radiuses= 30m,35m,40m,45m,50m; Intra Traffic Limit Rate= 80%

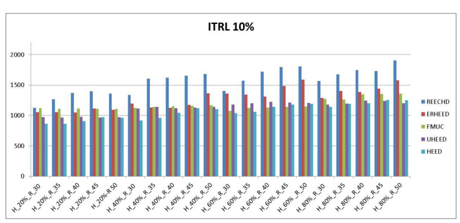

Fig. 7. Lifetime measure=HND; aggregation=50%; Heterogeneity level= 20%,40%,60%,80%; radiuses= 30m,35m,40m,45m,50m; Intra Traffic Limit Rate= 10%

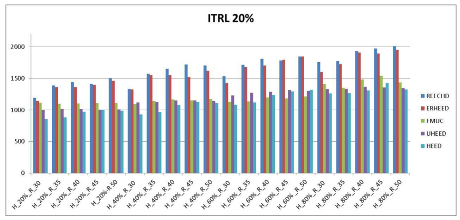

Fig. 8. Lifetime measure=HND; aggregation=50%; Heterogeneity level= 20%,40%,60%,80%; radiuses= 30m,35m,40m,45m,50m; Intra Traffic Limit Rate= 20%

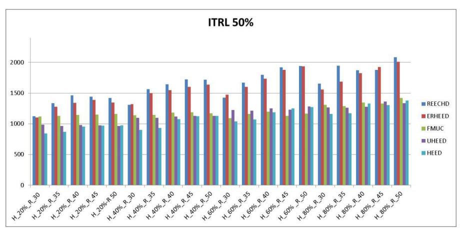

Fig. 9. Lifetime measure=HND; aggregation=50%; Heterogeneity level= 20%,40%,60%,80%; radiuses= 30m,35m,40m,45m,50m; Intra Traffic Limit Rate= 50%

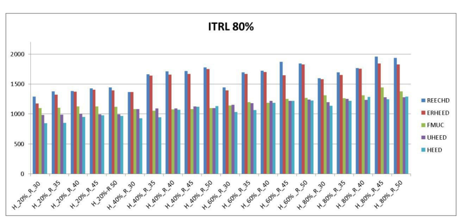

Fig. 10. Lifetime measure=HND; aggregation=50%; Heterogeneity level= 20%,40%,60%,80%; radiuses= 30m,35m,40m,45m,50m; Intra Traffic Limit Rate= 80%

Figure 7,8,9 and 10 shows the lifetime of the network for different heterogeneity levels until half of the nodes die (HND)) for REECHD, ERHEED, FMUC, UHEED and HEED protocols. These are run for an increasing heterogeneity level, and radius from 30m to 50m.

The network lifetime for all the protocols increase as the heterogeneity level approaches 80%. For each heterogeneity level, we show the network lifetime for different ITRL percentages. The most energy efficient results are achieved when the ITLR percentage is equal to 0.5 for HND and 0.2 for FND.

By looking at the Figures and we can observe that REECHD outperforms all other clustering protocols for HND lifetime measure.

Is it worth mentioning that REECHD outperforms FMUC [16], a protocol that have been conceived in the heterogeneous WSN context. We used the same simulation settings of FMUC [16] which outperforms the EEUC and DEBUC protocols. Thus REECHD outperforms both EEUC and DEBUC.

5 Conclusions

In this paper we reviewed the state of art of the most prominent energy efficient clustering for WSNs. We reviewed algorithms for heterogeneous and homogeneous WSNs, clustering protocols that consider the introduction of harvesting into WSNs and clustering protocols that are based on machine learning techniques.

We proposed the REECHD protocol for heterogenous WSNs. When selecting new CHs, REECHD considers not only the residual energy of the devices but also their induced work. This is estimated by using the node transmission rate. REECHD also introduces the concept of intra-traffic limit rate (ITLR). This defines a limit on the intra-traffic communication that all WSN clusters must comply with. REECHD is more suitable to cluster heterogeneous networks in which the communication rates of the devices are heterogeneous.

REECHD is more energy efficient when compared with well-known clustering protocols for homogeneous WSNs that are HEED, UHEED, ERHEED. REECHD also outperforms various clustering protocols that have been conceived in the heterogeneous WSN context that are FMUC, EEUC and DEBUC. In future work, we plan to implement a variation of REECHD which uses unequal size clustering. We plan to experiment various member selection strategies for cluster formation such as Knapsack. We plan to study heuristics to find the best ITLR under various WSN settings.

References

1. N. Aierken, R. Gagliardi, L. Mostarda, and Z. Ullah. Ruheed-rotated unequal clustering algorithm for wireless sensor networks. In *29th IEEE International Conference on Advanced Information Networking and Applications Workshops, AINA 2015 Workshops, Gwangju, South Korea, March 24-27, 2015*, pages 170–174, 2015.
2. M. A. Alsheikh, S. Lin, D. Niyato, and H. P. Tan. Machine learning in wireless sensor networks: Algorithms, strategies, and applications. *IEEE Communications Surveys Tutorials*, 16(4):1996–2018, Fourthquarter 2014.

3. M. S. Bahbahani and E. Alsusa. A cooperative clustering protocol with duty cycling for energy harvesting enabled wireless sensor networks. *IEEE Transactions on Wireless Communications*, 17(1):101–111, Jan 2018.
4. P. Ding, J. Holliday, and A. Celik. Distributed energy-efficient hierarchical clustering for wireless sensor networks. In *Proceedings of the First IEEE International Conference on Distributed Computing in Sensor Systems*, DCOSS'05, pages 322–339, Berlin, Heidelberg, 2005. Springer-Verlag.
5. C. Duan and H. Fan. A distributed energy balance clustering protocol for heterogeneous wireless sensor networks. In *2007 International Conference on Wireless Communications, Networking and Mobile Computing*, pages 2469–2473, Sept 2007.
6. A. H. Duffy. The "what" and "how" of learning in design. *IEEE Intelligent Systems*, 12:71–76, 05 1997.
7. E. Ever, R. Luchmun, L. Mostarda, A. Navarra, and P. Shah. Uheed - an unequal clustering algorithm for wireless sensor networks. In *SENSORNETS*, 2012.
8. W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8 - Volume 8*, HICSS '00, pages 8020–, Washington, DC, USA, 2000. IEEE Computer Society.
9. W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, pages 10 pp. vol.2–, Jan 2000.
10. F. Ishmanov and S. W. Kim. Distributed clustering algorithm with load balancing in wireless sensor network. In *2009 WRI World Congress on Computer Science and Information Engineering*, volume 1, pages 19–23, March 2009.
11. D. Kumar, T. C. Aseri, and R. Patel. Eehc: Energy efficient heterogeneous clustered scheme for wireless sensor networks. *Computer Communications*, 32(4):662 – 667, 2009.
12. D. Kumar, T. C Aseri, and R. Patel. Distributed cluster head election (dche) scheme for improving lifetime of heterogeneous sensor networks. 13, 09 2010.
13. P. Langley and H. A. Simon. Applications of machine learning and rule induction. *Commun. ACM*, 38(11):54–64, Nov. 1995.
14. J.-L. Liu and C. Ravishankar. Leach-ga: Genetic algorithm-basedenergy-efficient adaptive clustering protocolfor wireless sensor networks. 1:79–85, 01 2011.
15. P. Liu, T. l. Huang, X. y. Zhou, and G. x. Wu. An improved energy efficient unequal clustering algorithm of wireless sensor network. In *2010 International Conference on Intelligent Computing and Integrated Systems*, pages 930–933, Oct 2010.
16. T. Liu, J. Peng, J. Yang, G. Chen, and W. Xu. Avoidance of energy hole problem based on feedback mechanism for heterogeneous sensor networks. *International Journal of Distributed Sensor Networks*, 13(6):1550147717713625, 2017.
17. M. Micheletti, L. Mostarda, and A. Piermarteri. Rotating energy efficient clustering for heterogeneous devices (reechd). In *32nd IEEE International Conference on. Advanced Information Networking and Applications (IEEE AINA 2018), Pedagogical University of Cracow, Poland, May 16-18, 2018*, 2018.
18. C. Moraes and D. Har. Charging distributed sensor nodes exploiting clustering and energy trading. *IEEE Sensors Journal*, 17(2):546–555, Jan 2017.
19. L. Qing, Q. Zhu, and M. Wang. Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks. *Computer Communications*, 29(12):2230 – 2237, 2006.
20. F. K. Shaikh and S. Zeadally. Energy harvesting in wireless sensor networks: A comprehensive review. *Renewable and Sustainable Energy Reviews*, 55:1041 – 1054, 2016.

21. P. Sivakumar and M. Radhika. Performance analysis of leach-ga over leach and leach-c in wsn. *Procedia Computer Science*, 125:248 – 256, 2018. The 6th International Conference on Smart Computing and Communications.
22. G. Smaragdakis, I. Matta, and A. Bestavros. SEP: A Stable Election Protocol for clustered heterogeneous wireless sensor networks. In *Second International Workshop on Sensor and Actor Network Protocols and Applications (SANPA 2004)*, Boston, MA, August 2004.
23. Z. Ullah, L. Mostarda, R. Gagliardi, D. Cacciagrano, and F. Corradini. A comparison of heed based clustering algorithms – introducing er-heed. In *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*, pages 339–345, March 2016.
24. M. Yarvis, N. Kushalnagar, H. Singh, A. Rangarajan, Y. Liu, and S. Singh. Exploiting heterogeneity in sensor networks. In *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, volume 2, pages 878–890 vol. 2, March 2005.
25. O. Younis and S. Fahmy. Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Transactions on Mobile Computing*, 3(4):366–379, Oct. 2004.