A
Dissertation
On

# Energy Efficient Clustering in Wireless Sensor Networks

Submitted in Partial Fulfilment of the Requirement

For the Award of the Degree of

**Master of Technology**

**In**

**Computer Science & Engineering**

Submitted By
**Varun Kumar**
**2K12/CSE/24**

Under the Esteemed Guidance of
**Mr. R. K. Yadav**
**(Asst. Prof.)**

**DEPARTMENT OF COMPUTER ENGINEERING**

**DELHI TECHNOLOGICAL UNIVERSITY**

**JUNE, 2014**

**Department of Computer Engineering**
**Delhi Technological University**
**Delhi-110042**

# CERTIFICATE

This is to certify that the dissertation titled "**Energy Efficient Clustering in Wireless Sensor Networks**" is a bona fide record of work done at **Delhi Technological University** by **Varun Kumar, Roll No. 2K12/CSE/24** in partial fulfilment of the requirements for the degree of Master of Technology in Computer Science & Engineering. This work was carried out under my supervision and has not been submitted elsewhere, either in part or full, for the award of any other degree or diploma to the best of my knowledge and belief.

**(Mr. R. K. Yadav)**
**Assistant Professor & Project Guide**
**Department of Computer Engineering**
**Delhi Technological University**

Date: ______________

# ACKNOWLEDGEMENT

I would like to express my deepest gratitude to all the people who have supported and encouraged me during the course of this project without which this work could not have been accomplished.

First of all, I am very grateful to my project supervisor **Mr. R. K. Yadav** for providing me with the opportunity to carry out this work under his guidance. I am deeply indebted to him for the support, advice and encouragement he provided without which this work could not have proceeded smoothly. I am also thankful to all my friends specially **Ankit**, **Arpan**, **Kamal**, **Kanishka**, **Neha** and **Rahul** for being there for me all the time. Above all, I would like to express my gratitude to my parents for their encouragement and support during the completion of this work. Last but not the least I am grateful to Delhi Technological University for providing the right resources and environment for this work to be carried out.

**Varun Kumar**
**University Roll no: 2K12/CSE/24**
**M.Tech (Computer Science & Engineering)**
**Department of Computer Engineering**
**Delhi Technological University**

# ABSTRACT

Wireless sensor networks (WSNs) are comprised of hundreds to thousands of tiny battery powered sensor nodes constrained in energy and computation power. Because of limited energy source, economic utilization of energy is a critical issue in wireless sensor networks. Clustering techniques are most often used to reduce the consumption of energy by the sensor nodes due to data transmission. Partitioning the network into optimal number of clusters and selecting an optimal set of nodes as cluster heads is an NP-Hard problem.

A widely used class of clustering techniques is probabilistic clustering in which a predetermined optimal probability is used to facilitate the cluster head selection process. In this work, we devised a new technique that improves the energy efficiency of existing probabilistic clustering algorithms by optimizing the number of clusters and the distribution of cluster heads in the network. We also presented two generic approaches to integrate our technique into the existing probabilistic clustering techniques. The simulation results show a considerable improvement in energy efficiency of probabilistic clustering protocols and consequently a prolonged network life time.

The NP-Hard nature of clustering problem makes it a suitable candidate for the application of evolutionary algorithm and particle swarm optimization (PSO). In this work, we also suggest a PSO based solution to the optimal clustering problem by using residual energy and transmission distance of sensor nodes. Simulation results show a considerable improvement in network lifetime as compared to existing PSO based algorithms and other clustering protocols like LEACH and SEP.

We also present a new hybrid Clustering protocol, EEHCP for Multi-level heterogeneous wireless sensor networks. In this protocol, some nodes transmit data directly to base station while some use clustering technique to send data to base station. The proposed protocol aims to conserve energy by keeping three key design factors into consideration: (1) Finding the optimal distance up to which a node can directly send the data to base station, (2) electing an appropriate node as cluster head and (3) limiting the number of clusters in the network. The simulation results show that EEHCP enhanced the stability period by approximately 75% and almost tripled the throughput when compared to existing protocols like LEACH and SEP.

# Table of Contents

# List of Figures

# List of Abbreviations

| | |
|---|---|
| **BFS** | Breadth First Search |
| **BS** | Base Station |
| **CAG** | Clustered Aggregation |
| **CH** | Cluster Head |
| **DCRR** | Dynamic-Clustering Reactive Routing |
| **HEED** | Hybrid Energy Efficient Distributed Clustering |
| **LEACH** | Low Energy Adaptive Cluster Hierarchy |
| **EEHC** | Energy Efficient Hierarchical Clustering |
| **MANET** | Mobile Ad hoc Network |
| **PSO** | Particle Swarm Optimization |
| **SEP** | Stable Election Protocol |
| **TDMA** | Time Division Multiple Access |
| **TEEN** | Threshold sensitive Energy Efficient sensor Network protocol |
| **WSN** | Wireless Sensor Network |

# Chapter 1: Introduction

## 1.1 Wireless Sensor Network (WSN)

A wireless Sensor Network is composed of hundreds to thousands of tiny battery powered sensors that are constrained in energy and computation power. In WSN, a large number of sensor nodes are deployed randomly to monitor physical or environmental conditions such as temperature, sound, vibration, pressure, motion or pollutants at different locations [1]. Advancement in wireless communications, electronics and technological evolution has enabled the development in the field of WSNs due to their low cost and variety of applications such as health, home and military etc. Research is going on to solve different technical issues in various application areas [2], [5].

Sensor nodes consist of components capable of: sensing data, processing data and also communication components to further transmit or receive data. Nodes sense and send their reports toward a processing centre which is called “sink” or “base station” [3]. Communication in WSNs occurs in different ways which totally depends on the application. Generally, there are three main types of communication:

Clock Driven: In this type of communication sensors sense and gather data constantly but communicate periodically.

Event Driven: In event driven WSN, communication is triggered by occurrence or non occurrence of a particular event.

Query Driven: In query driven WSN, communication occurs in response to a query.

In all three types of communication, efficient use of energy is of concern while studying, designing or deploying such networks to prolong the sensing time and overall lifetime of the network.

## 1.2 Radio Energy Model

For energy dissipation inside a sensor node for transmitting the data, the first order radio energy model as described in [4] and others is used.

Transmitter inside radio hardware dissipates energy to run transmit electronics and amplifier. Similarly for receiving data energy is dissipated to run receive electronics as shown in figure 1.1.

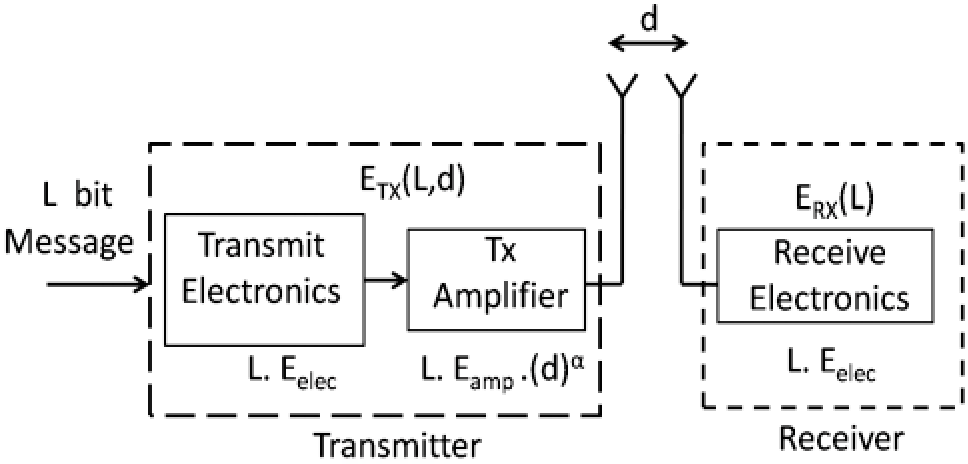


**Fig. 1.1 Radio Energy Model**

In order to achieve an acceptable SNR to transmit an L bit message to a node situated at distance d, the energy consumed by radio is given by-

$$E_{Tx}(L,d) = \begin{cases} L.E_{elec} + L.\varepsilon_{fs}.d^2 & if\ d \leq d_0 \\ L.E_{elec} + L.\varepsilon_{mp}.d^4 & if\ d > d_0 \end{cases} \tag{1.1}$$

sss

Where, $E_{elec}$ is the energy, dissipated per bit to run the transmitter or the receiver circuit; $\varepsilon_{fs}$ and $\varepsilon_{mp}$ depend on the transmitter amplifier model and d is the distance between the sender and the receiver.

By equating the two expressions at $d = d_0$, we get $d_0 = \sqrt{\frac{\varepsilon_{fs}}{\varepsilon_{mp}}}$.

To receive an L−bit message the radio expends-

$$E_{Rx} = L.E_{elec} \tag{1.2}$$

## 1.3 Research Objective

The nature of wireless sensor networks is quite different from general wireless networks because of various constraints on sensors such as limited battery power and computation power. WSNs are highly application specific in nature. In practice, cost and other application specific issues affect the communication properties of WSN system. For example, radio communication range of WSNs is short in comparison to other wireless networks. The system performance characteristics vary considerably even though the same basic principles of wireless communication are used. The size, power, cost and their tradeoffs are fundamental constraints in WSNs.

Efficient utilization of power is one of the most critical issue in design and operations of WSNs. The main objective of this work is to develop energy efficient communication techniques that can lower the energy consumption and consequently can improve the network life time. Clustering techniques are widely used to reduce the energy consumption in transmission. In this research work we tried to develop the techniques to improve the performance of existing algorithms and tried to explore biologically inspired optimization methods such as particle swarm optimization (PSO) to solve the problem of clustering that is NP-hard otherwise.

## 1.4 Thesis Organization

In this chapter we have given a brief introduction of wireless sensor networks and radio energy model that serve as basis for many popular protocols for WSNs.

In Chapter 2 we will present a detailed survey on different types of clustering algorithms available in literature and their variants that are used in practice with an insight to their classification.

In Chapter 3 we will propose two viable modifications in existing probabilistic clustering algorithms to improve their energy efficiency. We will also provide extensive simulation study to show the relevance of our proposed modifications.

In Chapter 4 we will propose a particle swarm optimization based solution to the clustering problem. Simulation results will also be presented to evaluate the performance of our proposed solution.

In chapter 5 we will propose an Energy Efficient Hybrid Clustering Protocol (EEHCP) for multilevel heterogeneous wireless sensor networks and will compare the performance of our proposed algorithm with some popular algorithms such as LEACH and SEP.

# Chapter 2: Clustering in Wireless Sensor Networks

In most wireless sensor network (WSN) applications nowadays the entire network must have the ability to operate unattended in harsh environments in which pure human access and monitoring cannot be easily scheduled or efficiently managed or it's even not feasible at all [1]. Based on this critical expectation, in many significant WSN applications the sensor nodes are often deployed randomly in the area of interest by relatively uncontrolled means and they form a network in an ad hoc manner [2], [3]. Moreover, considering the entire area that has to be covered, the short duration of the battery energy of the sensors and the possibility of having damaged nodes during deployment, large populations of sensors are expected; it's a natural possibility that hundreds or even thousands of sensor nodes will be involved. In addition, sensors in such environments are energy constrained and their batteries usually cannot be recharged. Therefore, it's obvious that specialized energy-aware routing and data gathering protocols offering high scalability should be applied in order that network lifetime is preserved acceptably high in such environments. Naturally, grouping sensor nodes into clusters has been widely adopted by the research community to satisfy the above scalability objective and generally achieve high energy efficiency and prolong network lifetime in large-scale WSN environments. The corresponding hierarchical routing and data gathering protocols imply cluster-based organization of the sensor nodes in order that data fusion and aggregation are possible, thus leading to significant energy savings. In the hierarchical network structure each cluster has a leader, which is also called the cluster head (CH) and usually performs the special tasks referred above (fusion and aggregation), and several common sensor nodes (SN) as members.

The cluster formation process eventually leads to a two-level hierarchy where the CH nodes form the higher level and the cluster-member nodes form the lower level. The sensor nodes periodically transmit their data to the corresponding CH nodes. The CH nodes aggregate the data and transmit them to the base station (BS) either directly or through the intermediate communication with other CH nodes. However, because the CH nodes send all the time data to higher distances than the common nodes, they naturally spend energy at higher rates. A common solution in order to balance the energy consumption among all the network nodes is to periodically re-elect new CHs thus rotating the CH role among all the nodes over time in each cluster. A typical example of the implied hierarchical data communication within a clustered network assuming single hop intra-cluster communication and multi-hop inter-cluster communication is illustrated in Figure 2.1.

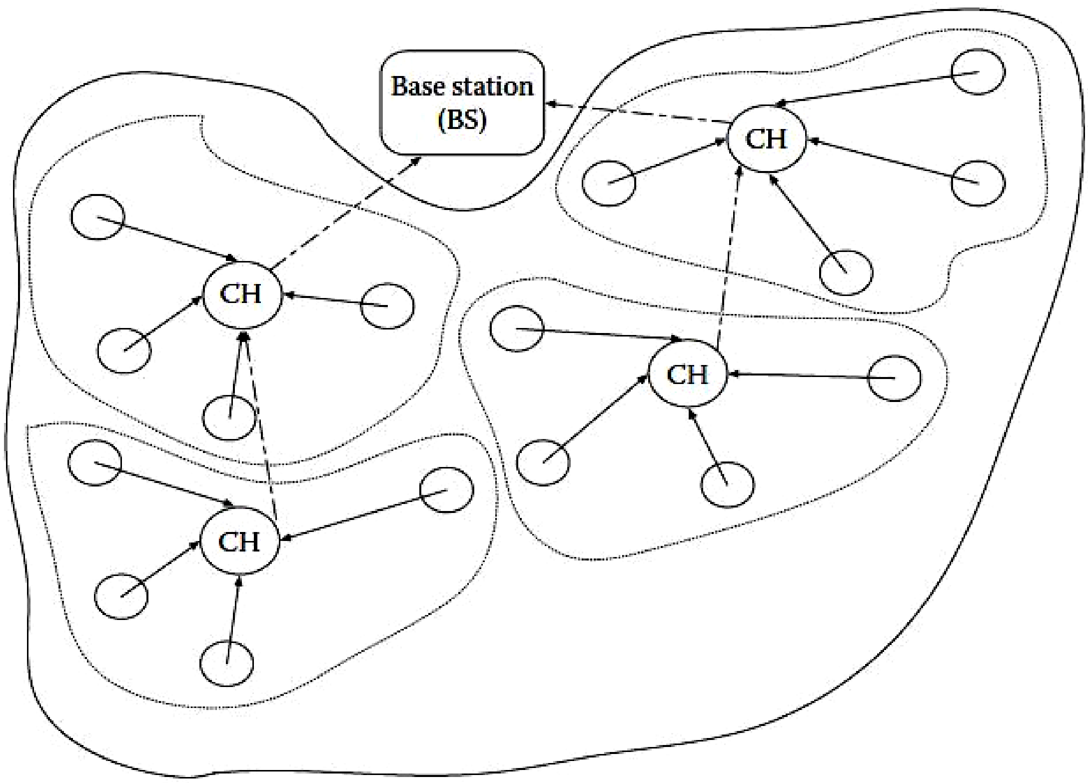


**Fig. 2.1 Data Communication in Clustered Network**

The BS is the data processing point for the data received from the sensor nodes, and where the data is accessed by the end user. It is generally considered fixed and at a far distance from the sensor nodes. The CH nodes actually act as gateways between the sensor nodes and the BS. The function of each CH, as already mentioned, is to perform common functions for all the nodes in the cluster, like aggregating the data before sending it to the BS. In some way, the CH is the sink for the cluster nodes, and the BS is the sink for the CHs. Moreover, this structure formed between the sensor nodes, the sink (CH), and the BS can be replicated as many times as it is needed, creating (if desired) multiple layers of the hierarchical WSN (multi-level cluster hierarchy).

## 2.1 Objective of clustering

As was mentioned at the beginning, hierarchical clustering in WSNs can greatly contribute to overall system scalability, lifetime, and energy efficiency. Hierarchical routing is an efficient way to lower energy consumption within a cluster, performing data aggregation and fusion in order decrease the number of transmitted messages to the BS. On the contrary, a single-tier network can cause the gateway to overload with the increase in sensors density. Such overload might cause latency in communication and inadequate tracking of events. In addition, the single-tier architecture is not scalable for a larger set of sensors covering a wider area of interest because the sensors are typically not capable of long-haul communication. Hierarchical clustering is particularly useful for applications that require scalability to hundreds or thousands of nodes. Scalability in this context implies the need for load balancing and efficient resource utilization. Applications requiring efficient data aggregation are also natural candidates for clustering. Routing protocols can also employ clustering [9], [27]. In Ref. [50], clustering was also proposed as a useful tool for efficiently pinpointing object locations.

In addition to supporting network scalability and decreasing energy consumption through data aggregation, clustering has numerous other secondary advantages and corresponding objectives [1]. It can localize the route setup within the cluster and thus reduce the size of the routing table stored at the individual node. It can also conserve communication bandwidth because it limits the scope of inter-cluster interactions to CHs and avoids redundant exchange of messages among sensor nodes. Moreover, clustering can stabilize the network topology at the level of sensors and thus cuts on topology maintenance overhead. Sensors would care only for connecting with their CHs and would not be affected by changes at the level of inter-CH tier. The CH can also implement optimized management strategies to further enhance the network operation and prolong the battery life of the individual sensors and the network lifetime. A CH can schedule activities in the cluster so that nodes can switch to the low-power sleep mode and reduce the rate of energy consumption. Furthermore, sensors can be engaged in a round-robin order and the time for their transmission and reception can be determined so that the sensors reties are avoided, redundancy in coverage can be limited, and medium access collision is prevented.

## 2.2 Design Challenges of Clustering in WSNs

WSNs also present several particular challenges in terms of design and implementation. Similar challenges and design goals have also been faced earlier in the field of mobile ad hoc networks (MANETs), and naturally a lot of related ideas considering clustering protocols etc. have been borrowed from that field. In WSNs, however, the limited capabilities such as battery power, transmission range, processing hardware and memory, etc. of the sensor nodes combined with the special location-based conditions such as not easily accessed in order recharge the batteries or replace the entire sensors make the energy efficiency and the scalability factors even more crucial. Moreover, the challenge of prolonging network lifetime under the above restrictions is difficult to be met by using only traditional techniques. Consequently, it becomes unavoidable to follow alternative techniques leading to more efficient protocols with a lot of differences compared to the ones designed for MANETs.

Beyond the typical challenges mentioned above some additional important considerations in the design process of clustering algorithms for WSNs should be the following:

Cluster formation: The CH selection and cluster formation procedures should generate the best possible clusters. However they should also preserve the number of exchanged messages low and the total time complexity should, if possible remain constant and independent to the growth of the network. This yields a very challenging trade-off. Application Dependency: When designing clustering and routing protocols for WSNs, application robustness must be of high priority and the designed protocols should be able to adapt to a variety of application requirements. Secure communication: As in traditional networks, the security of data is naturally of equal importance in WSNs too.

The ability of a WSN clustering scheme to preserve secure communication is ever more important when considering these networks for military applications. Synchronization: Slotted transmission schemes such as TDMA allow nodes to regularly schedule sleep intervals to minimize energy used. Such schemes require corresponding synchronization mechanisms and the effectiveness of this mechanisms must be considered. Data aggregation: Because this process makes energy optimization possible it remains a fundamental design challenge in many sensor network schemes nowadays. However its effective implementation in many applications is not a straightforward procedure and has to be further optimized according to specific application requirements.

## 2.3 Clustering Parameters

It is worth to know some important parameters with regard to the whole clustering procedure in WSNs. These parameters serve as the basic means for further comparison and categorization of the presented clustering protocols throughout this chapter.

A. **Number of clusters:** In most recent probabilistic and randomized clustering algorithms the CH election and formation process lead naturally to variable number of clusters. In some published approaches, however, the set of CHs are predetermined and thus the number of clusters is preset. The number of clusters is usually a critical parameter with regard to the efficiency of the total routing protocol.
B. **Intra-cluster communication:** In some initial clustering approaches the communication between a sensor and its designated CH is assumed to be direct (one-hop communication). However, multi-hop intra-cluster communication is often required, i.e., when the communication range of the sensor nodes is limited or the number of sensor nodes is very large and the number of CHs is bounded.
C. **Nodes and CH mobility:** If we assume stationary sensor nodes and stationary CHs we are normally led to stable clusters with facilitated intra-cluster and inter-cluster network management. On the contrary, if the CHs or the nodes themselves are assumed to be mobile, the cluster membership for each node should dynamically change forcing clusters to evolve over time and probably need to be continuously maintained.
D. **Nodes types and roles:** In heterogeneous network models the CHs are assumed to be equipped with significantly more computation and communication resources than others. In most usual homogeneous network models all nodes have the same capabilities and just a subset of the deployed sensors is designated as CHs.
E. **Cluster formation methodology:** In most recent approaches, when CHs are just regular sensors nodes and time efficiency is a primary design criterion, clustering is being performed in a distributed manner without coordination. In few earlier approaches a centralized or hybrid approach is followed; one or more coordinator nodes are used to partition the whole network off-line and control the cluster membership.
F. **Cluster-head selection:** The leader nodes of the clusters in some proposed algorithms mainly for heterogeneous environments can be preassigned. In most cases, in homogeneous environments, the CHs are picked from the deployed set of nodes either in a probabilistic or completely random way or based on other more specific criteria such as residual energy, connectivity etc.
G. **Algorithm complexity:** In most recent algorithms the fast termination of the executed protocol is one of the primary design goals. Thus, the time complexity or

convergence rate of most cluster formation procedures proposed nowadays is constant or just dependent on the number of CHs or the number of hops. In some earlier protocols, however, the complexity time has been allowed to depend on the total number of sensors in the network, focusing in other criteria first.

H. **Multiple levels:** In several published approaches the concept of a multi-level cluster hierarchy is introduced to achieve even better energy distribution and total energy consumption instead of using only one cluster level. The improvements offered by multi-level clustering are to be further studied, especially when we have very large networks and inter-CH communication efficiency is of high importance.

I. **Overlapping:** Several protocols give also high importance on the concept of node overlapping within different clusters either for better routing efficiency or for faster cluster formation protocol execution or for other reasons. Most of the known protocols, however, still try to have minimum overlap only or do not support overlapping at all.

## 2.4 Classification of Clustering Protocols

There have been several different ways based directly on the above-mentioned parameters to initially distinguish and further classify the algorithms used for WSNs clustering, [4]. Two of the most early and common classifications in the bibliography are

(i) Clustering algorithms for homogeneous or heterogeneous networks

(ii) Centralized or distributed clustering algorithms.

The first of the above classifications is based on the characteristics and functionality of the sensors in the cluster, whereas the other one is based on the method used to form the cluster. In heterogeneous sensor networks [6], [10] there are generally two types of sensors, sensors with higher processing capabilities and complex hardware, used generally to create some sort of backbone inside the WSN being preset as the CH node and also serve as data collectors and processing centres for data gathered by other sensor nodes, and common sensors, with lower capabilities, used to actually sense the desired attributes in the field. In homogeneous networks, all nodes have the same characteristics, hardware and processing capabilities. In this case which is the most usual in nowadays applications every sensor can become a CH. Moreover, the CH role can be periodically rotated among the nodes in order achieve better load balancing and more uniform energy consumption. Also, when all the nodes have the same capabilities (homogeneous environments), a distributed CH election and formation process is the most appropriate technique to gain increased flexibility and fast execution-convergence time, independent of the number of nodes of the WSN.

There are also a few approaches using centralized or hybrid techniques as described in [5], [6], [12] where one or more coordinator nodes or the BS is responsible to partition the whole network off-line and control the cluster membership, however they are naturally not suitable for practical general-purpose large-scale WSNs applications. They may be suitable only for special purpose limited-scale applications where high-quality connectivity and network partitioning is required. Here we mainly focus on distributed clustering protocols that are the most efficient, especially for large networks and are the most general purpose and widely used nowadays.

Another common classification is between static and dynamic clustering. A cluster formation procedure is regarded as dynamic when it includes regular (periodic or event driven) CH re-election or cluster reorganization procedures, either to effectively react to network topology changes and adjust appropriately the cluster topology, or simply aiming at the appropriate rotation of the CH role among the nodes to gain in energy efficiency. Dynamic cluster architectures make a better use of the sensors in a WSN and naturally lead to improved energy consumption management and network lifetime.

Most of the known clustering algorithms for WSNs can be further distinguished into two main categories, depending on cluster formation criteria and parameters used for CH election:

(i) Probabilistic clustering algorithms

(ii) Non probabilistic clustering algorithms

In the category of probabilistic selection clustering algorithms [11–24], a priori probability assigned to each sensor node is used to determine the initial CHs or some other type of random election procedure is used. The probabilities initially assigned to each node often serve as the primary criterion in order for the nodes to decide individually on their election as CHs in a flexible, uniform, fast and completely distributed way; however other secondary criteria may also be considered either during CH election process such as the residual energy or during the cluster formation process such as the proximity or the communication cost in order achieve better energy consumption and network lifetime. Beyond the high energy efficiency which is facilitated also from the periodic CH re-election scheme usually adopted, the clustering algorithms of this category usually achieve faster execution/convergence times and reduced volume of exchanged messages.

In the category of non probabilistic clustering algorithms [25–43], more specific and deterministic criteria for CH election and cluster formation are primarily considered, which are mainly based on the node's proximity, connectivity, degree etc. and on the information received from other closely located nodes. The cluster formation procedure here is mainly based on the communication of nodes with their neighbours (one or multi-hop neighbours) and generally requires more intensive exchange of messages and probably graph traversing in some extent, thus leading sometimes to worse time complexity than probabilistic clustering algorithms. On the contrary these algorithms are usually more reliable toward the direction of extracting robust and well-balanced clusters. In addition to node proximity, some algorithms [37–40] also use a combination of metrics such as the remaining energy, transmission power, mobility, etc., forming corresponding combined weights to achieve more generalized goals than single-criterion protocols. In the same category we also address a relatively new and quite challenging class of clustering algorithms for WSNs, namely, the biologically inspired protocols [41–43], based on swarm intelligence which are probably the most promising alternative approaches for clustering in WSNs nowadays.

Furthermore, there is a special-purpose class of clustering protocols, those that are suitable for Reactive Networks [44–49]. These protocols have clearly different objectives compared to the most common category of proactive clustering algorithms to which all the other above-mentioned protocols belong. They are specifically oriented to applications with timing restrictions and usually take advantage of user queries for the sensed data or of specific triggering events that occur in the WSN.

There is another class of protocols that consider mobility of sensor nodes in network. The number of applications that require mobile nodes is considerably limited; also there is not much specialized work in the literature till now. We find some relevant information and specific related work in [51] and [52].

## 2.5 Earliest Clustering Protocols

Before the detailed presentation of the main clustering categories introduced above, we want to discuss the former protocols used before the last decade for clustering in WSNs. The first clustering algorithms for WSNs were naturally inspired from or entirely based on corresponding algorithms already studied and used in the field of wired sensor networks or, later, in the field of mobile ad hoc networks. Uniformly assigned unique identifiers were usually the key parameter for selecting CHs in those algorithms.

One of the first such clustering algorithms, initially developed for wired sensor networks was the Linked Cluster Algorithm (LCA) [7]. LCA was a distributed ID based, one-hop, static clustering algorithm, trying to maximize network connectivity.

The main disadvantage of LCA was that usually led to excessive number of clusters. An improved LCA-based approach, generating smaller number of clusters was given in [8] (LCA2). Both algorithms [7] and [8] had limited scope as clustering algorithms for WSNs because they did not consider the problem of limited energy of WSNs. Additionally, both protocols construct one-hop clusters and their time complexity is O(n) which is rather unacceptable for large size WSNs. Similarly, an early example of clustering protocols initially developed for mobile ad hoc networks and then applied also to WSNs was the adaptive clustering algorithm presented in [9]. Other classical paradigms of clustering algorithms designed initially for MANETs, were the MAX-MIN [29], HC [28] and WCA [38] algorithms. Finally, some of the initial clustering schemes proposed for WSNs were based on some sort of manual formation of the clusters and are mostly applicable to heterogeneous environments. Such a representative algorithm can be found in [10]. These manual-based clustering formation schemes are not applicable to general-purpose WSNs of our days, unless specific conditions are met.

## 2.6 Probabilistic Clustering Approaches

As the need for efficient use of WSNs on large regions increased in the last decade dramatically, more specific clustering protocols were developed to meet the additional requirements such as increased network lifetime, reduced and evenly distributed energy consumption, scalability etc. The most significant and widely used representatives of this category such as LEACH, EEHC, HEED, SEP and their most valuable extensions will be described in this section. They are all probabilistic in nature and their main objective was to reduce the energy consumption and prolong the network lifetime. Some of them follow a random approach for CH election, the initially assigned probabilities serve as the basis for the random election of the CHs, whereas others like HEED follow a hybrid probabilistic methodology, secondary criteria such as residual energy etc. are also considered during CH election.

### 2.6.1 Low Energy Adaptive Clustering Hierarchy (LEACH)

One of the first and most popular clustering protocols proposed for WSNs was LEACH [11], [12]. It is probably the first dynamic clustering protocol which addressed specifically the WSNs needs, using homogeneous stationary sensor nodes randomly deployed, and it still serves as the basis for other improved clustering protocols for WSNs. It's a hierarchical, probabilistic, distributed, one-hop protocol, with main objectives (a) to improve the lifetime of WSNs by trying to evenly distribute the energy consumption among all the nodes of the network and (b) to reduce the energy consumption in the network nodes by performing data aggregation and thus reducing the number of communication messages. It forms clusters based on the received signal strength and also uses the CH nodes as routers to the BS. All the data processing such as data fusion and aggregation are local to the cluster. LEACH forms clusters by using a distributed algorithm, where nodes make autonomous decisions without any centralized control. All nodes have a chance to become CHs to balance the energy spent per round by each sensor node. Initially a node decides to be a CH with a probability "$p$" and broadcasts its decision. Specifically, after its election, each CH broadcasts an advertisement message to the other nodes and each one of the other non-CH nodes determines a cluster to belong to, by choosing the CH that can be reached using the least communication energy based on the signal strength of each CH message.

The role of being a CH is rotated periodically among the nodes of the cluster to balance the load. The rotation is performed by getting each node to choose a random number r' between 0 and 1. A node becomes a CH for the current rotation round if the number is less than the following threshold:

$$T(n) = \begin{cases} \dfrac{p}{1 - p\left(r \bmod \dfrac{1}{p}\right)} & if\ n \in G \\ 0 & otherwise \end{cases}$$

Where, $p$ is the desired percentage of CH nodes in the sensor population, $r$ is the current round number, $G$ is the set of nodes that have not been CHs in the last $1/p$ rounds. The clusters are formed dynamically in each round and the time to perform the rounds is also selected randomly. Generally, LEACH can provide a quite uniform load distribution in one-hop sensor networks. Moreover, it provides a good balancing of energy consumption by random rotation of CHs.

LEACH has also some clear drawbacks. Because the decision on CH election and rotation is probabilistic, there is still a good chance that a node with very low energy gets selected as a CH. Due to the same reason it is possible that the elected CHs will be concentrated in one part of the network and some nodes will not have any CH in their range. Also, the CHs are assumed to have a long communication range so that the data can reach the BS directly. This is not always a realistic assumption because the CHs are usually regular sensors and the BS is often not directly reachable to all nodes. Moreover, LEACH forms in general one-hop intra-cluster and inter-cluster topology where each node should transmit directly to the CHs and thereafter to the BS, thus normally it cannot be used effectively on networks deployed in large regions.

### 2.6.2 Energy Efficient Hierarchical Clustering (EEHC)

EEHC, another significant probabilistic clustering algorithm was proposed earlier in [13]. The main objective of this algorithm was to address the shortcomings of one-hop random selection algorithms such as LEACH by extending the cluster architecture to multiple hops. It is a distributed, k-hop hierarchical clustering algorithm aiming at the maximization of the network lifetime.

Initially, each sensor node is elected as a CH with probability "$p$" and announces its election to the neighbouring nodes within its communication range. The above CHs are now called the "volunteer" CHs. Next, all the nodes that are within $k$-hops distance from a "volunteer" CH, are supposed to receive the election message either directly or through intermediate forwarding. Consequently, any node that receives such CH election message and is not itself a CH, becomes a member of the closest cluster. Additionally, a number of 'forced' CHs are elected from nodes that are neither CH nor belong to a cluster. Specifically, if the election messages do not reach a node within a preset time interval $t$, the node becomes a "forced" CH assuming that it is not within k hops of all volunteer CHs.

However, the most challenging feature of the EEHC algorithm is the direct extension to a corresponding multi-level clustering structure. The initial clustering process is recursively repeated at the level of CHs making it possible to build multiple levels of cluster hierarchy. Assuming that an $h$-level cluster hierarchy has been constructed in that way with corresponding preset CH election probabilities $p_1, p_2, \cdots, p_h$ for each level, the algorithm ensures the efficient $h$-level communication between common sensor nodes and the BS, as follows : Common sensor nodes transmit their collected data to the corresponding first-level CHs, the CHs of the first-level clusters transmit the aggregated data to the second-level CHs and so on, till the top (#$h$) level of the clustering hierarchy is reached; the CHs of those h-level clusters transmit their final aggregated data reports to the BS. This multi-level protocol has a time complexity of $O(k_1 + k_2 + \cdots + k_h)$, where $k_i$ is the corresponding parameter (for each level) to the above-mentioned "$k$" parameter. That was a significant improvement over the O(n) time complexity that many of the existing algorithms till then like LCA had, and made this algorithm quite suitable for large networks.

### 2.6.3 Hybrid Energy Efficient Distributed clustering (HEED)

Another improved and very popular energy-efficient protocol is HEED [14]. HEED is a hierarchical, distributed, clustering scheme in which a single-hop communication pattern is retained within each cluster, where as multi-hop communication is allowed among CHs and the BS. The CH nodes are chosen based on two basic parameters, residual energy and intra-cluster communication cost. Residual energy of each node is used to probabilistically choose the initial set of CHs. On the other hand, intra-cluster communication cost reflects the node degree or node's proximity to the neighbour and is used by the nodes in deciding to join a cluster or not. Thus, unlike LEACH, in HEED the CH nodes are not selected randomly. Only sensors that have a high residual energy are expected to become CH nodes. Also, the probability of two nodes within the transmission range of each other becoming CHs is small. Unlike LEACH, this means that CH nodes are well distributed in the network.

Moreover, when choosing a cluster, a node will communicate with the CH that yields the lowest intra-cluster communication cost. In HEED, each node is mapped to exactly one cluster and can directly communicate with its CH. Also, energy consumption is not assumed to be uniform for all the nodes. The algorithm is divided into three stages. At the beginning, the algorithm sets an initial percentage of CHs among all sensors. This percentage value, *C*prob, is used to limit the initial CHs announcements to the other sensors. Each sensor sets its probability of becoming a CH, *CH*prob, as follows:

$CHprob = Cprob * E_{residual}/E_{max}$, here $E_{residual}$ is the current energy in the sensor, and $E_{max}$ is the maximum energy, which corresponds to a fully charged battery. *CH*prob is not allowed to fall below a certain threshold *p*min, which is selected to be inversely proportional to $E_{max}$. The main body of the algorithm consists of a constant number of iterations. Every sensor goes through these iterations until it finds the CH that it can transmit to with the least transmission power (cost). If it hears from no CH, the sensor elects itself to be a CH and then sends an announcement message to its neighbours informing them about the change of status. Finally, each sensor doubles its *CH*prob value and goes to the next iteration of this phase. It stops executing this phase when its *CH*prob reaches 1. Therefore, there are two types of CH status that a sensor could announce to its neighbours: (a) The sensor becomes a tentative CH if its *CH*prob is less than 1; it can change its status to a regular node at a later iteration if it finds a lower cost CH (b) The sensor permanently becomes a CH if its *CH*prob has reached 1. At the end, each sensor makes a final decision on its status. It either picks the least cost CH or announces itself as CH. Note also that for a given sensor's transmission range, the probability of CH selection can be adjusted to ensure inter-CH connectivity.

Generally, HEED's mechanism to select the CHs and form the clusters produces a uniform distribution of cluster heads across the network through localized communications with little overhead. It also clearly outperforms LEACH with regard to the network lifetime and the desired distribution of energy consumption. However, synchronization is required and the energy consumed during data transmission for far away cluster heads is significant, especially in large-scale networks.

### 2.6.4 Other variants

On the basis of the probabilistic nature of LEACH, several other protocols were developed aiming at better energy consumption and overall performance. First, the LEACH-C and the LEACH-F protocols were proposed in [12], introducing slight modifications to the initial LEACH cluster formation procedure. LEACH-C is a centralized version of LEACH, in the sense that the responsibility of the cluster creation is transferred to the BS. Each node is initially obligated to perform a direct communication with the BS in order that a global view of the network is formed. As a result an improved cluster formation procedure is performed and a slightly better overall performance of the network is achieved. LEACH-F is also a centralized protocol and is based initially on the same global clustering scheme as in LEACH-C. The main difference lies on the fact that all clusters are fixed once when they are formed, thus reducing the overhead of cluster formation in the network. However, the above design directive prevents the use of the protocol in networks with any kind of mobility. A valuable extension to LEACH has been proposed in [15], a two-level LEACH, where the key idea of probabilistic CH election is extended to construct a two-level clustering scheme. The outer level consists of the "primary" CHs where as the inner level consists of the “secondary” CHs. The “primary” CHs in each outer-level cluster communicate directly with the corresponding “secondary” CHs and the “secondary” CHs in each inner level cluster communicate directly with the corresponding nodes in that sub cluster. Data fusion as well as communication within a cluster is performed like in LEACH, in TDMA schedules. The selection of the “primary” and the “secondary” CHs is performed also in the same way as in LEACH, by setting corresponding a priori probabilities for each node. The “primary” CHs are selected first and the “secondary” CHs are selected next from the remaining nodes. The probability to become a “primary” CH is normally less than the probability to become a “secondary” CH. Generally, the two-level clustering scheme of this algorithm achieves a significant reduction on the percentage of nodes that have to transmit data to the BS in each round. Thus, it is normally expected to reduce the total energy spent.

Most of the published probabilistic clustering algorithms construct disjoint clusters. On the contrary, in [16] the authors argue that allowing some degree of overlap among clusters can be quite effective for many tasks like inter-cluster routing, topology discovery and node localization, recovery from CH failure, etc. Specifically, they introduce a probabilistic, distributed Multi-hop Overlapping Clustering Algorithm (MOCA) for organizing the sensors into overlapping clusters. The goal of the clustering process is to ensure that each node is either a CH or within $k$ hops from at least one CH, where $k$ is a preset cluster radius. The algorithm initially assumes that each sensor in the network becomes a CH with probability $p$. Each CH then advertises itself to the sensors within its radio range. This advertisement is forwarded to all sensors that are no more than $k$ hops away from the CH. A node sends a request to all CHs that it heard from to join their clusters. In the join request, the node includes the ID of all CHs it heard from, which implicitly implies that it is a boundary node. The CH election probability $p$ is used to control the number of clusters in the network and the degree of overlap among them. The authors also provide extensive simulation work to validate appropriate values of $p$ to achieve particular cluster count and overlapping degree.

Beyond the pure use of a priori probabilities to elect the initial CHs, another significant parameter additionally used (like in HEED) is the residual energy of each node. Two such recent algorithms, similar also to LEACH with regard to the overall clustering process were proposed in [17] and [18]. In Time Controlled Clustering Algorithm (TCCA) [17], the whole operation is divided into rounds trying to achieve better load distribution among sensor nodes. In each round initially the CH selection procedure takes place and overall cluster formation process follows. Each node decides to elect itself as a CH or not based on the suitable combination of two basic criteria, its residual energy and a preset probability $p$. Actually in this step TCCA applies a direct combination of LEACH and HEED algorithms by having the energy fraction $E_{residual}/E_{max}$ participating directly in the computation of the (LEACH inspired) CH-election threshold $T_i$ in each round. When a CH is selected, it announces its selection to the neighbouring nodes by sending a message which includes its node id, initial time-to-live, its residual energy, and a time stamp. The time-to-live parameter is selected according to the residual energy and it is used to restrict the size of the clusters that are formed. On the other hand, in [18] Energy Efficient Clustering Scheme (EECS), a constant number of CHs are elected (i) based on their residual energy (as the main criterion) and (ii) using localized competition process without iteration to complete the cluster formation process. Specifically, the candidate CHs compete for their chance to be elected at any given round by broadcasting their residual energy to neighbouring candidates. If a given node does not find a node with more residual energy, it becomes the CH. Additionally clusters are then formed by retaining variable sizes dynamically, mainly depending on the distance of each cluster from the BS. As a result, the corresponding algorithm can effectively lead to better energy consumption and uniform load distribution having a clearly better behaviour compared to LEACH in simulated experiments, based on the fact that clusters at a greater distance from the BS require more energy for transmission than those that are closer.

Also, considering the HEED algorithm a slight however effective modification was also proposed in [20]. Specifically, the difference here is the treatment of nodes that eventually did not hear from any CH; during the finalization phase of the initial protocol all these nodes become CHs themselves. On the contrary, in [20] the authors claim that re-executing the algorithm for just those orphaned nodes could lead to significant improvements. Furthermore, this slight modification was shown to significantly decrease the CHs' count which then leads to reduced size of the routing tree needed during inter-CH communication which finally results in faster data gathering procedures.

Similarly, considering the multi-level EEHC algorithm, a valuable extension, EEMC that includes additional CH election criteria is proposed in [19], where the expected number of CHs at each level is previously determined by analytical formulas. The authors generalize the analysis given in [13] and present results about the optimal number of CHs at a certain level. Considering the formation process, they follow a top-down approach starting from the formation of level-1 clusters. The CHs at each level are randomly selected according to a certain probability. The probability of a node becoming a CH is proportional to the residual energy of the node as well as the distance of this node to the sink node or to the CH it belongs to at lower levels. The distance is taken into account as each CH should transmit the aggregated data on behalf of its member nodes to its next level CH and a large distance between these two nodes contributes to fast energy consumption in the transmitting CH. The probabilities are also normalized so that the expected number of CHs at each level is according to the optimal values determined in their analysis. Extensive simulation work is also provided, in which the EEMC protocol is shown to achieve longer network lifetime and less latency compared to LEACH and EEHC protocols.

Finally, some random selection protocols have also been developed that follow an even more clearly random CH election procedure by randomly waiting or by generating a random competition, etc. Such an early proposed algorithm was RCC [21], which was initially designed for MANETs and applies the 'First Declaration Wins' rule. In [22], another completely randomized clustering algorithm CLUBS was proposed, where each node participates in the election procedure by choosing a random number from a fixed integer range and then it counts down from that number silently. Two more recent and quite efficient (converging in constant time) completely randomized protocols were proposed in [23] and [24]. In [23] Fast Local Clustering service (FLOC), a distributed protocol that produces approximately equal sized clusters with minimum overlap is presented.

## 2.7 Non Probabilistic Clustering Approaches

Alternatively to the probabilistic algorithms described in the previous section, another basic class of clustering algorithms for WSNs primarily adopt more specific and deterministic criteria for CHs election and cluster formation, which are mainly based on the nodes' proximity (connectivity, degree, etc.) and on the information received from other closely located nodes. The cluster formation procedure here is mainly based on the communication of nodes with their neighbours and generally requires more intensive exchange of messages and probably graphs traversing in some extent. The use of additional metrics including the remaining energy, transmission power, mobility, etc. in the form of combined weighted values is also a quite promising technique followed to achieve more generalized goals than other single-metric protocols. Furthermore, an even more challenging and promising non probabilistic clustering approach is based on the use of swarm intelligence and has led to the construction of corresponding biologically inspired clustering protocols that already have been shown to extend network lifetime in WSNs.

### 2.7.1 Node Proximity and Graph Based Protocols

A proximity-traversing-based algorithm, Hierarchical Control Clustering (HCC) was earlier proposed in [27]. It is a distributed multi-hop hierarchical clustering algorithm which also efficiently extends to form a multi-level cluster hierarchy. Any node in the WSN can initiate the cluster formation process. The algorithm proceeds in two phases namely, Tree Discovery and Cluster Formation.”The tree discovery phase is basically a distributed formation of a Breadth-First-Search (BFS) tree rooted at the initiator node. Each node, u, broadcasts a signal once every *p* units of time, carrying the information about its shortest hop distance to the root, *r*. A node *v* that is neighbour of *u* will choose *u* to be its parent and will update its hop distance to the root, if the route through *u* is shorter. The broadcast signal carries the parent ID, the root ID, and the sub tree size. Every node updates its sub tree size when its children sub tree size change. The cluster formation phase starts when a sub tree on a node crosses the size parameter, *k*. The node initiates cluster formation on its sub tree. It will form a single cluster for the entire sub tree if the sub tree size is less than 2*k*, or else, it will form multiple clusters. The cluster size and the degree of overlap are also considered.

Two other early proposed algorithms of this category can be found in [28] and [29]. In Highest Connectivity (HC) [28] a connectivity-based heuristic is proposed, in which the sensor node with maximum number of one-hop neighbours is elected as a CH in its neighbourhood. The formation of one-hop clusters and the clock synchronization requirement limit the practical usage of this algorithm nowadays. On the other hand, in Max-Min D-Cluster algorithm [29], a distributed algorithm is proposed, in which the clusters consist of nodes that are no more than d-hops away from the CH. It does not require clock synchronization and it provides a better load balancing compared to LCA and HC algorithms.

Other more recent examples of proximity-connectivity and neighbour’s information based algorithms have been proposed in [30–32]. In [30] a typical centralized, graph-based clustering approach (EEDC) is presented. To minimize the number of clusters and therefore maximize the energy saving, EEDC models the cluster creation process as a clique-covering problem and uses the minimum number of cliques to cover all vertices in the graph. The sink also dynamically adjusts the clusters based on spatial correlation and the received data from the sensors. The algorithm produces robust and well-balanced clusters; however it is centralized and thus not suitable for large-scale WSNs.

In [31] Clustering Algorithm via Waiting Timer (CAWT), a distributed proximity-connectivity-based algorithm for constructing cluster hierarchy has been proposed for homogeneous sensors with the same transmission range. Once sensors are deployed, each sensor broadcasts a “hello” message to show its presence to the neighbours while listening to the others. The sensors that hear a significant number of “hello” messages (meaning that are nodes with high connectivity) organize into clusters while others are waiting to form clusters. The performance of the algorithm was evaluated using simplified simulations leading to quite good results with regard to network lifetime.

However, as it is clearly observed, the generalization of the algorithm is subject to detailed evaluation with respect to load balancing, CH re-election, and energy usage across the network.

Similarly, in [32] a distributed clustering procedure, EACLE which beyond the proximity takes also in account the residual energy of each node, is followed. It is mainly based on the information of 2-hop neighbours with a practical transmission power control scheme, and then builds a broadcast tree only by cluster heads. Initially, each sensor is in a ‘waiting’ state and waits for time $T1$ which is a monotonous decreasing function on the residual energy of the node. When the timer expires, the waiting node becomes a CH and broadcasts two packets with different transmission power each, which contain the list of the neighbour-IDs received before broadcasting. When a waiting node receives a power-low packet it becomes a member node, whereas when it receives a power-high packet, it compares its own neighbour list with the list of IDs in the receiving packet, to decide if it should continue waiting or become a CH. Also, each node executes the clustering process periodically. Once a node becomes a CH in a specific round, its timer is then set to a longer value to avoid becoming a CH again in the next round.

A quite valuable alternative Algorithm for Cluster Establishment (ACE) was given in [33].Unlike other distributed clustering schemes, ACE employs an emergent algorithm. Emergent algorithms much like artificial neural networks evolve to optimal solution through a mix of local optimization steps. Initially, a node decides to become a “candidate” CH, and then it broadcasts an invitation message. Upon getting the invitation, a neighbouring sensor joins the new cluster and becomes a follower of the new CH. At any moment, a node can be a follower of more than one cluster. Next, the migration phase takes place in order the best candidate for being CH to be selected. Each CH periodically checks the ability of its neighbours for being a CH and decides to step down if one of these neighbours has more followers than it does. A node that has the largest number of followers and the least overlap with other clusters will be considered as the best final candidate for CH.

### 2.7.2 Biologically Inspired Clustering Approaches

In the last few years some new algorithms have also been proposed based on swarm intelligence techniques which model the collective behaviour of social insects such as ants. They have shown very promising results in simulated experiments when compared to protocols like LEACH and HEED with regard to network lifetime. In [41] the authors propose such a swarm intelligence-based clustering algorithm based on the ANTCLUST method. ANTCLUST is a model of an ant colonial closure to solve clustering problems.

In colonial closure model, when two objects meet together they recognize whether they belong to the same group by exchanging and comparing information about them. In the case of a WSN, initially the sensor nodes with more residual energy become CHs independently. Then, randomly chosen nodes meet each other, exchange information, and clusters are created, merged, and discarded through these local meetings and comparison of their information. Each node with less residual energy chooses a cluster based on specific criteria, like the residual energy of the CH, its distance to the CH, and an estimation of the cluster size. Eventually, energy efficient clusters are formed that result in an extension of the lifetime of the WSN.

Another related approach that ensures the good distribution of CHs and high energy efficiency, can be found in [42]. Also, in [43], a protocol that has the objective of minimizing the intra-cluster distance and optimizing the energy consumption of the network using Particle Swarm Optimization (PSO) is presented and evaluated via simulations. Generally, biologically inspired clustering algorithms show that they can dynamically control the CH selection while achieving quite uniform distribution of CHs and energy consumption. However, they have to be studied further as it is pointed out in the literature.

### 2.7.3 Clustering Protocols for Reactive Networks

Reactive algorithms respond to specific triggering events that occur in the WSN. Nodes may react instantly to sudden and drastic changes in the value of a sensed attribute. This approach is useful for time-critical applications, but not particularly suited for applications where data retrieval is required on a regular basis.

The Threshold sensitive Energy Efficient sensor Network protocol (TEEN) [44] forms a hierarchical clustered structure, grouping nearby nodes within the same cluster. The protocol defines two thresholds: the hard threshold is a threshold value for the sensed attribute, while the soft threshold is a threshold value for change in the sensed attribute. The concept of threshold is highly significant in a variety of WSN applications, such as fire alarm, temperature monitoring etc. The nodes transmit sensor readings only when they fall above the hard threshold and change by given soft threshold.

The Adaptive Periodic-TEEN (APTEEN) [45] is a variation of TEEN which addresses the main shortcomings of TEEN. It is a hybrid routing protocol wherein the nodes still react to time-critical situations, but also give an overall picture of the network at periodic intervals in an energy efficient manner.

More recently, the Clustered Aggregation (CAG) [46], mechanism was proposed, which utilizes the spatial correlation of sensory data to further reduce the number of transmissions by providing approximate results to aggregate queries. CAG guarantees the result to be within a user-specified error-tolerant threshold.

The Updated CAG algorithm [47] extends CAG defining two operation modes, depending on the dynamics of the environment. In the interactive mode, users issue a one-shot query and the network generates a single response. On the other hand, in the streaming mode, the CHs transmit a stream of response for a query that is issued just once.

Recently, Guo and Li proposed Dynamic-Clustering Reactive Routing (DCRR) algorithm [49]. It borrows ideas from biological neuron networks, following the observation that the latter also employ a many-to-one (neurons-to-brain) communication paradigm, similarly to the nodes of a WSN. In DCRR, once an incident emerges, the CH is dynamically selected in the incident region according to the nodes’ residual energy.

# PROPOSED WORK

# Chapter 3: A Technique to Improve the Energy Efficiency of Probabilistic Clustering Algorithms

## 3.1 Objective

Probabilistic clustering algorithms are widely used in wireless sensor networks to reduce the energy consumption by the sensors nodes in data transmission. The main objective of this study is to devise a new technique to improve the energy efficiency of existing probabilistic clustering algorithms by optimizing the number of clusters and the distribution of cluster heads in the network. Our main focus here is to suggest a widely applicable technique that can be seamlessly integrated into the existing algorithms in the class of probabilistic clustering algorithms rather than proposing a new clustering algorithm. Some of the existing probabilistic clustering algorithms operate in centralized manner while some operate in distributed manner so we shall propose two generic frameworks, one for distributed algorithms and one for centralized algorithms to seamlessly integrate our proposed modifications into existing algorithms.

## 3.2 Motivation

In this section we will explain in detail the reasoning behind our proposed improvements in cluster head selection process of probabilistic clustering algorithms. We are using the same radio energy model for power consumption in radio transmission as used in most of the probabilistic clustering algorithms as in [11] [12 ] [13] etc.

### 3.2.1 Requirement of Clusters of Almost same size

Probabilistic clustering techniques usually do not take into account the relative positions of selected cluster heads; as a result there is a fair chance that in several rounds a considerable number of the cluster heads will be either in close proximity or very far from each other. In case, when a considerable number of cluster heads are in close proximity, the number of nodes in the clusters associated with these cluster heads will be considerably low in comparison to other clusters which will give rise to uneven clustering. While a well balanced clustering i.e. nearly same sized clusters is crucial for reasonable performance of probabilistic clustering techniques such as LEACH [11], TEEN [44] etc. On the contrary if selected cluster heads are very far from each other, there will be a significant wastage of energy in inter-cluster communication in protocols such as HEED [14], EEHC [13] etc.

Therefore a uniform distribution of cluster heads over entire network region is desirable to achieve a well balanced clustering.

### 3.2.2 Maintaining the Optimal Number of Clusters in Each Round

Apart from the distribution of cluster heads in the network there is one more issue with probabilistic clustering algorithms which is the number of clusters formed in each round. For a reasonable performance all the probabilistic clustering algorithms require that the number of clusters formed in each round should be as close to an analytically predetermined optimal value as possible. There are high chances that the number of clusters formed in each round, vary considerably from the optimal value because of the probabilistic nature of these algorithms.

Therefore maintaining the optimal number of clusters in each round is unavoidable for the better performance of clustering algorithms.

## 3.3 Proposed Modifications

Based on above discussion we proposed two modifications in the existing clustering algorithms.

### 3.3.1 The Closeness Parameter ξ

As our first improvement, we want to introduce a parameter ξ to denote the closeness of two cluster heads and its value will depend on size, node density of network and number of cluster heads to be selected. If in a particular round the distance between any two selected cluster heads will be smaller than ξ, they will be considered too close to each other and one of them has to drop its decision of becoming cluster head. This way we can ensure that cluster heads will not be very close to each other and will be well distributed across the network.

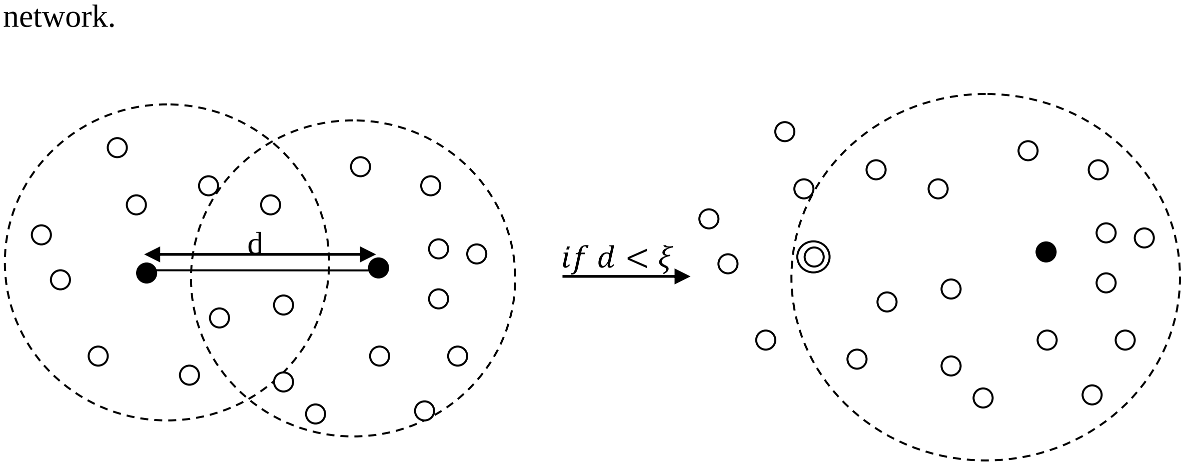


**Fig. 3.1 Effect of ξ-Closeness on Cluster Formation**

### 3.3.2 The Threshold Increment Factor σ

Though the above proposed improvement in cluster head selection process causes a significant improvement in the performance of probabilistic clustering algorithms, it has a small drawback. Consider the case when a large number of potential cluster heads (the nodes for which the value of generated random number is less than the threshold value) in a round of a particular epoch are in close proximity then many of them will drop their decision to become cluster head. In such a situation the number of selected cluster heads would be significantly lesser than the optimal number of cluster heads. This will lead to bigger size clusters and consequently more energy consumption in intra-cluster communication. Also in the last round of the epoch all the nodes that have not become cluster head so far in that particular epoch will have to become cluster head and so chances will be higher for the selection of more cluster heads than optimal number of cluster heads. This will increase the long distance transmission to the sink. To deal with this situation we want to increase the number of potential cluster heads in each round so that the number of selected cluster heads after dropping some of potential CHs because of ξ-closeness be as near to optimal value as possible. For this, we need to raise the threshold value in equation (2.1) so that more nodes than usual will be eligible to become cluster head.

So as our second improvement we want to introduce another parameter σ, denoting the threshold increment factor. The increment in the threshold value given by equation (2.1) will not be same for each round in an epoch as original threshold value itself increase in each subsequent round of an epoch and becomes equal to 1 in last round but we want to keep σ constant in each round so we can calculate new increased threshold using

$$T_{NEW}(n) = T(n) + \sigma(1 - T(n)) \tag{3.1}$$

In our assumption σ will be constant for a particular configuration of network and a particular choice of closeness factor ξ.

### 3.3.3 Determination of Closeness Factor ξ and Threshold Increment Factor σ

The value of ξ and σ is very crucial for better performance of clustering algorithms. The value of ξ depends on the network configuration, i.e. size of the network, number of nodes deployed in the network, density of nodes, optimal number of cluster heads etc. The value of σ depends on the value of ξ but it also depends on network configuration. We will calculate the optimal values of ξ and σ experimentally by running multiple instances of the algorithm for different combinations of ξ and σ while keeping all other network parameters constant.

## 3.4 Framework for Integrating into Existing Probabilistic Algorithms

We can easily integrate these two proposed improvements into existing probabilistic clustering techniques to improve their energy efficiency. We will present two generic approaches of integration, one that requires little intervention of centralized authority such as base station and other follows a completely distributed approach.

### 3.4.1 Centralized Approach

In Fig. 3.2 we are giving a centralized approach of integration.

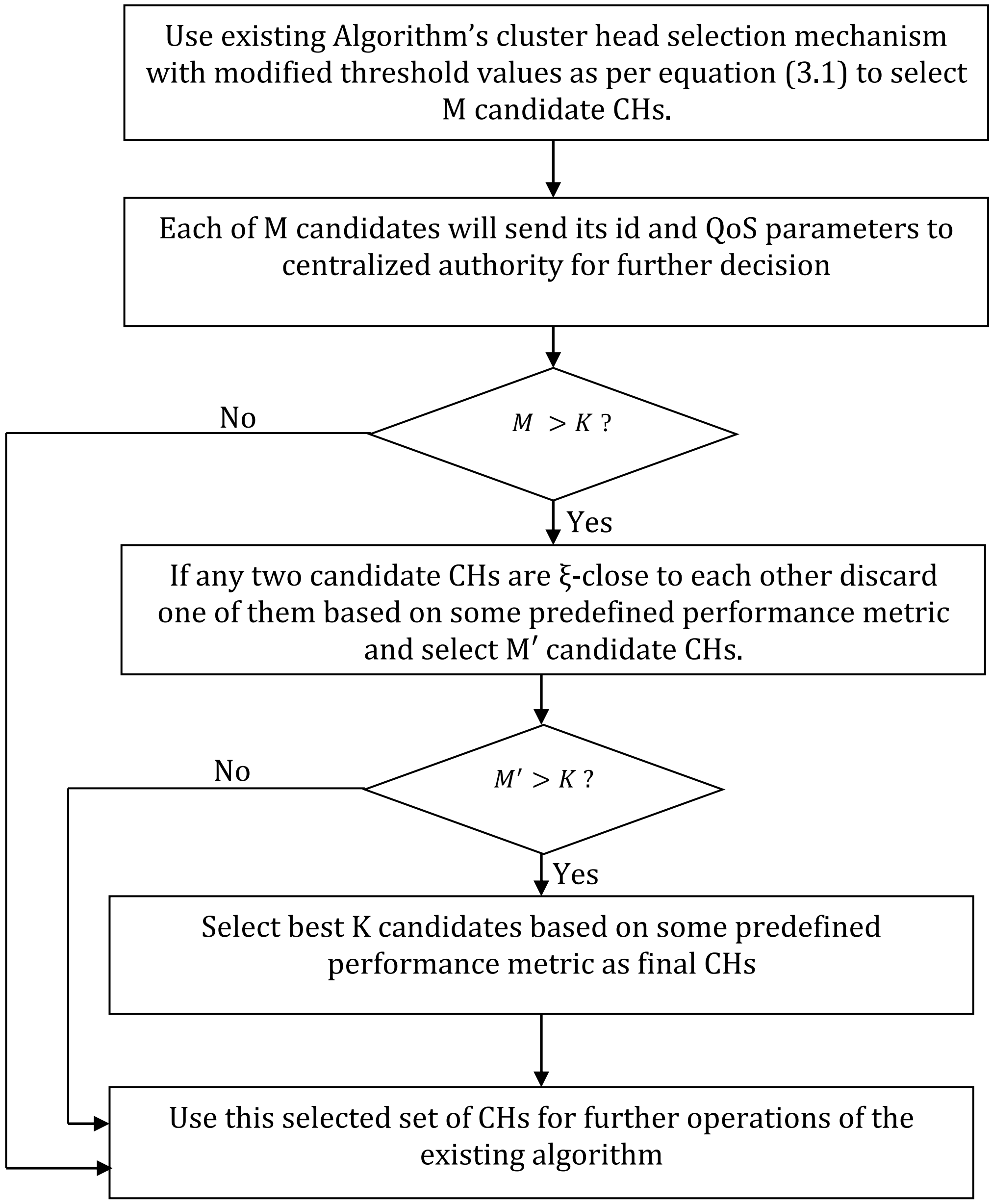


**Fig. 3.2 A centralized approach for integrating proposed improvements to existing algorithms**

Here K is the optimal number of CHs that is precalculated using analytical methods as described in [5]. The predefined performance metric can include QoS factors such as quality of link between candidate node and base station, congestion, node density around the candidate node, its residual energy etc.

### 3.4.2 Distributed Approach

In distributed approach, each node autonomously takes its decision of becoming a cluster head or not without intervention of any central authority. The following figure explains a distributed approach for integration.

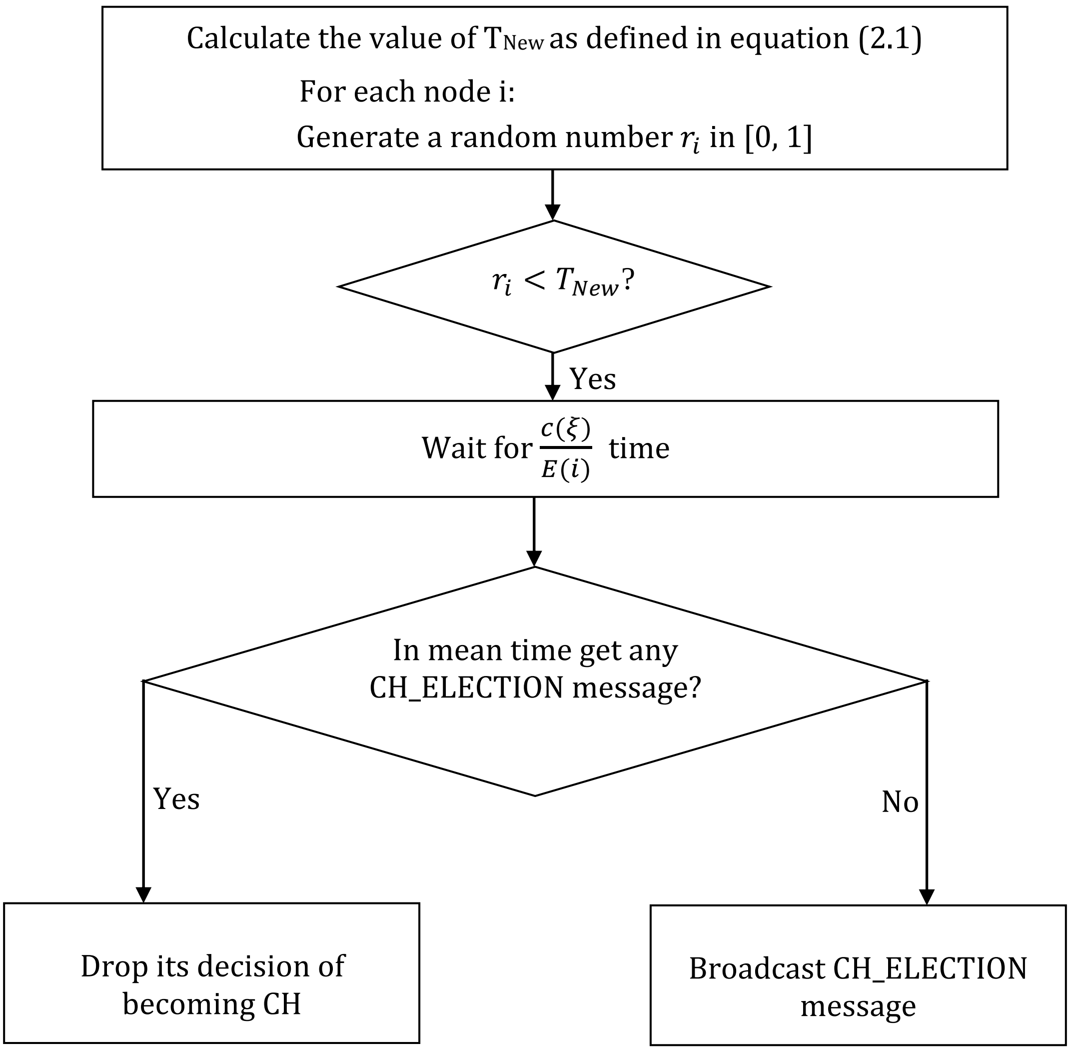


**Fig. 3.3 A distributed approach for integrating proposed improvements to existing algorithms**

Here E(i) is the residual energy of node i and c(ξ) is a function whose value depends only on closeness factor ξ. Waiting for $\frac{c(\xi)}{E(i)}$ time ensures that among all the candidate CHs that are in ξ-closeness range of each other only the node with highest residual energy will be chosen as cluster head.

Both centralized and distributed approaches have their own advantages. Centralized approach offers better QoS implementation mechanism and more tolerant to variance of values of σ and ξ from optimal value. On the other hand distributed approach requires minimal message exchange among nodes for cluster head election. Here we consider c as a function of ξ only but for QoS implementation we can make c as a function of other QoS parameters with ξ as well.

## 3.5 Simulation Results

We conducted a rigorous simulation study to evaluate the performance of our proposed modifications in probabilistic clustering. We evaluate the performance of some representative probabilistic clustering algorithms such as LEACH [11], SEP [52], TEEN [44], HEED [14] in their original version and with integration of our proposed modifications.

We assume a square network field of dimension 100m X 100m with 100 sensors deployed in it and Matlab is used for the purpose of simulation. Simulation parameters are shown in Table 1.

| **Description** | **Parameter** | **Value** |
|---|---|---|
| Initial energy | $E_0$ | 0.5J |
| Electronic circuitry energy | $E_{elec}$ | 50nJ/bit |
| Multi-path co-efficient | $\varepsilon_{mp}$ | 10 pJ/bit/m2 |
| Free space co-efficient | $\varepsilon_{fs}$ | 0.0013 pJ/bit/m4 |
| Data aggregation energy | $E_{DA}$ | 5 nJ/bit/signal |
| Data packet size | $P_{pkt}$ | 30 bytes |
| Total no. of nodes | $N$ | 100 |
| Optimal percentage of CHs | $P_{opt}$ | 0.1 |

**Table 3.1: Simulation Parameters**

**Determination of ξ and σ:** To determine optimal values of ξ and σ we run our algorithm multiple times for different combinations of ξ and σ for a given configuration of network and choose best values. Table 3.2 shows energy consumption for different combinations of ξ and σ for two different network configurations.

| ξ | σ | Energy Consumption in Configuration 1 (Joule) | Energy Consumption in Configuration 2 (Joule) |
|---|---|---|---|
| 0 | 0.00 | 14.69 | 119.65 |
| 5 | 0.05 | 14.56 | 117.74 |
| 5 | 0.09 | 14.66 | 117.80 |
| 5 | 0.17 | 14.27 | 119.46 |
| 10 | 0.05 | 13.62 | 119.43 |
| 10 | 0.15 | 13.73 | 116.77 |
| 10 | 0.20 | 13.91 | 115.99 |
| 11 | 0.05 | 13.54 | 115.54 |
| 11 | 0.15 | 13.46 | 115.54 |
| 11 | 0.19 | 13.27 | 115.54 |
| 12 | 0.05 | 13.22 | 115.59 |
| 12 | 0.15 | 13.47 | 114.91 |
| 12 | 0.20 | 13.67 | 115.67 |
| 13 | 0.05 | 13.22 | 115.27 |
| 13 | 0.15 | 13.08 | 115.13 |
| 13 | 0.20 | 13.17 | 115.17 |
| 14 | 0.05 | 13.10 | 115.76 |
| 14 | 0.15 | 13.01 | 115. 58 |
| 14 | 0.20 | 12.93 | 115.58 |
| 15 | 0.00 | 13.23 | 116.44 |
| 15 | 0.05 | 13.01 | 116.01 |
| 15 | 0.10 | 12.66 | 116.62 |
| 15 | 0.11 | 12.61 | 116.49 |
| 15 | 0.13 | 12.63 | 116.49 |
| 15 | 0.15 | 12.54 | 116.32 |

| | | | |
|---|---|---|---|
| 15 | 0.16 | 12.51 | 117.51 |
| 15 | 0.17 | 12.84 | 117.96 |
| 15 | 0.19 | 12.73 | 116.59 |
| 16 | 0.05 | 12.87 | 116.87 |
| 16 | 0.15 | 12.69 | 116.76 |
| 16 | 0.20 | 12.67 | 116.14 |
| 17 | 0.05 | 12.95 | 116.97 |
| 17 | 0.15 | 12.83 | 116.52 |
| 17 | 0.25 | 12.83 | 117.01 |
| 18 | 0.05 | 12.87 | 116.66 |
| 18 | 0.10 | 12.67 | 117.67 |
| 18 | 0.20 | 12.68 | 116.19 |
| 19 | 0.05 | 12.93 | 114.93 |
| 19 | 0.15 | 13.11 | 114.27 |
| 19 | 0.20 | 13.09 | 114.81 |
| 20 | 0.05 | 13.83 | 115.36 |
| 20 | 0.20 | 13.59 | 116.94 |
| 20 | 0.25 | 13.77 | 116.27 |
| 22 | 0.05 | 13.69 | 115.07 |
| 22 | 0.10 | 13.62 | 115.21 |
| 22 | 0.15 | 13.71 | 114.71 |
| 25 | 0.05 | 14.20 | 112.52 |
| 25 | 0.15 | 14.25 | 112.65 |
| 25 | 0.20 | 14.19 | 111.19 |
| 27 | 0.05 | 14.30 | 111.69 |
| 27 | 0.15 | 14.67 | 112.74 |
| 27 | 0.25 | 14.83 | 111.73 |
| 29 | 0.02 | 14.55 | 110.88 |
| 29 | 0.04 | 14.07 | 110.86 |
| 29 | 0.05 | 14.34 | 110.47 |
| 29 | 0.07 | 14.39 | 110.39 |
| 29 | 0.09 | 14.94 | 110.43 |

| | | | |
|---|---|---|---|
| 29 | 0.10 | 14.99 | 110.59 |
| 29 | 0.13 | 14.21 | 110.04 |
| 29 | 0.14 | 14.69 | 110.54 |
| 29 | 0.15 | 15.12 | 110.87 |
| 29 | 0.16 | 14.99 | 110.51 |
| 29 | 0.20 | 14.45 | 112.68 |
| 29 | 0.21 | 14.56 | 112.37 |
| 29 | 0.25 | 14.74 | 114.90 |
| 30 | 0.05 | 14.69 | 114.12 |
| 30 | 0.10 | 15.67 | 115.07 |
| 30 | 0.20 | 15.19 | 115.43 |

**Table 3.2 Determination of Optimal Values of ξ and σ using simulation**

We calculated the optimal value of ξ and σ for two Network configurations. In first configuration we assumed a 50 X 50 network field with 30 sensors deployed in it. In second configuration we assumed a 100 X 100 network field with 100 sensors deployed in it. We found 15m and 0.16 to be optimal values for ξ and σ respectively for first configuration. Similarly for second configuration optimal values are 29m and 0.13 respectively.

In Figure 3.4 to 3.9 we will show the performance of a probabilistic clustering algorithm in its original version and after the application of our proposed modifications. We can see in figure 3.4 that after applying our suggested modification there is an increase of 43% in stability period of HEED and an overall 66 % increment in total lifetime of network in simulated environment.

Similarly in case of SEP, in Figure 3.6 and 3.7 we can see a considerable increase in the energy efficiency and network life time. We can also observe that after application of our proposed modification there is significant improvement in the no. of clusters formed in each round.

Similarly in LEACH, there is a significant improvement of in overall lifetime of the network. There is also an increase of around 37 % in stability period of Network.

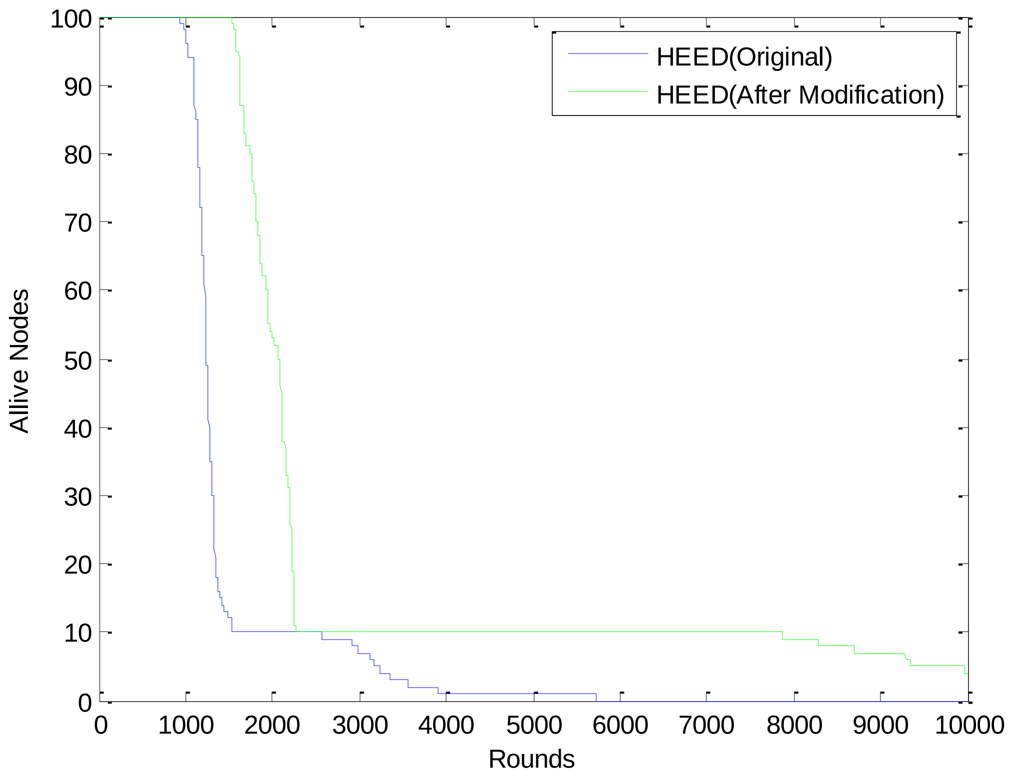


**Fig. 3.4 No. of Alive Nodes per Round in HEED**

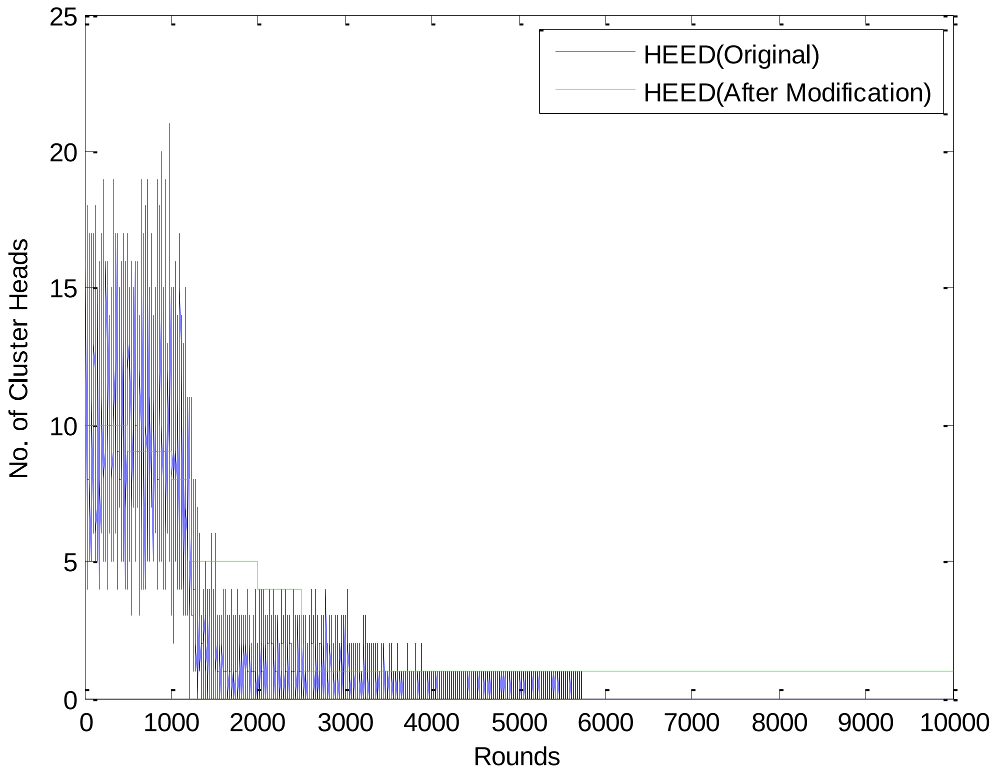


**Fig. 3.5 No. of CHs per Round in HEED**

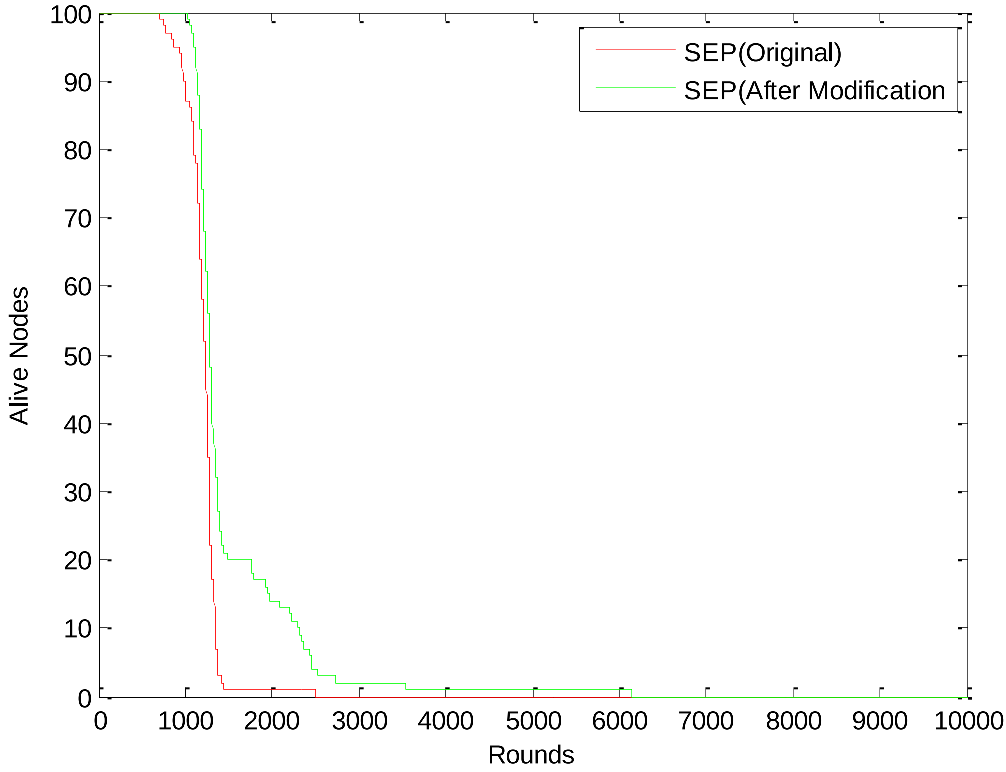


**Fig. 3.6 No. of Alive Nodes per Round in SEP**

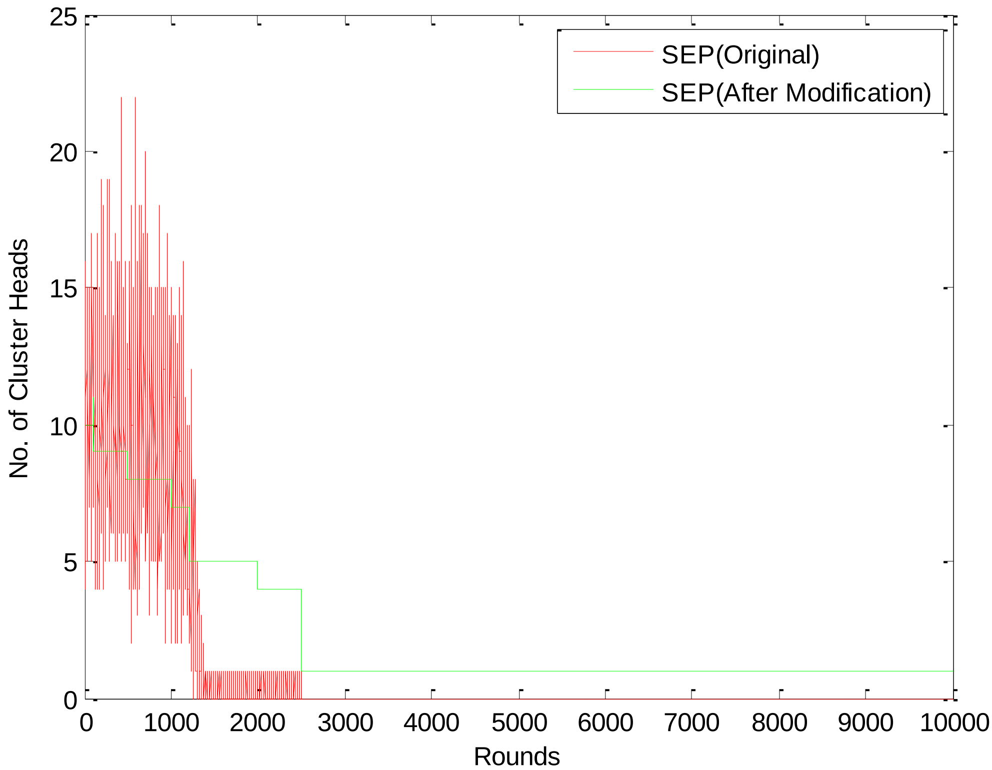


**Fig. 3.6 No. of CHs per Round in SEP**

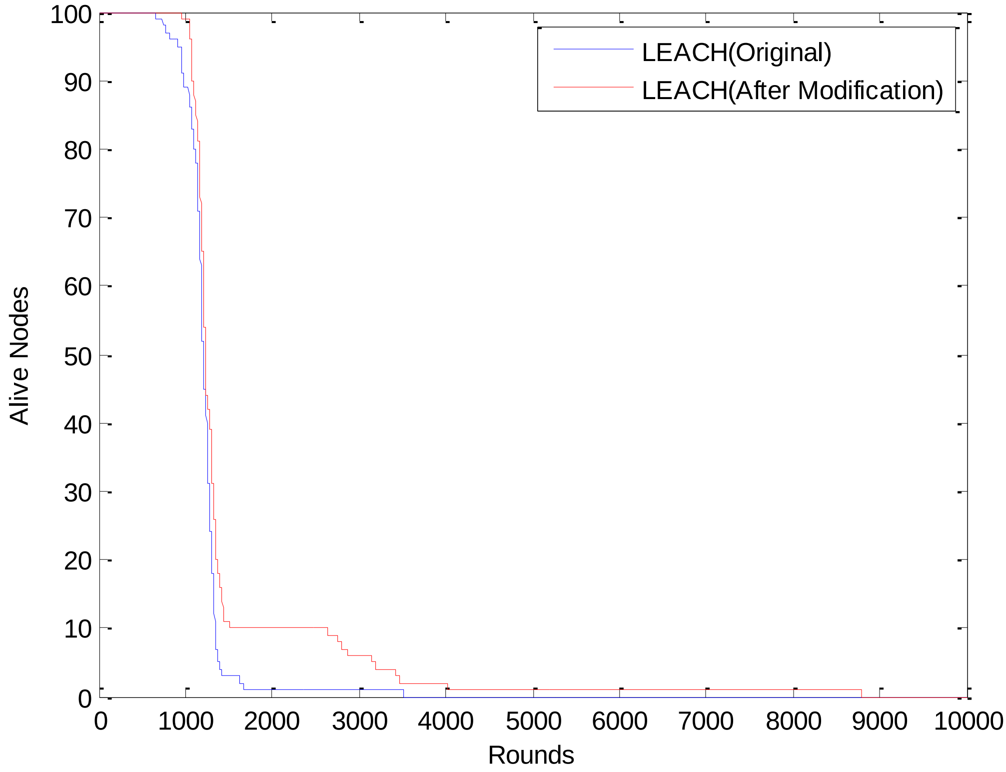


**Fig. 3.6 No. of Alive Nodes per Round in LEACH**

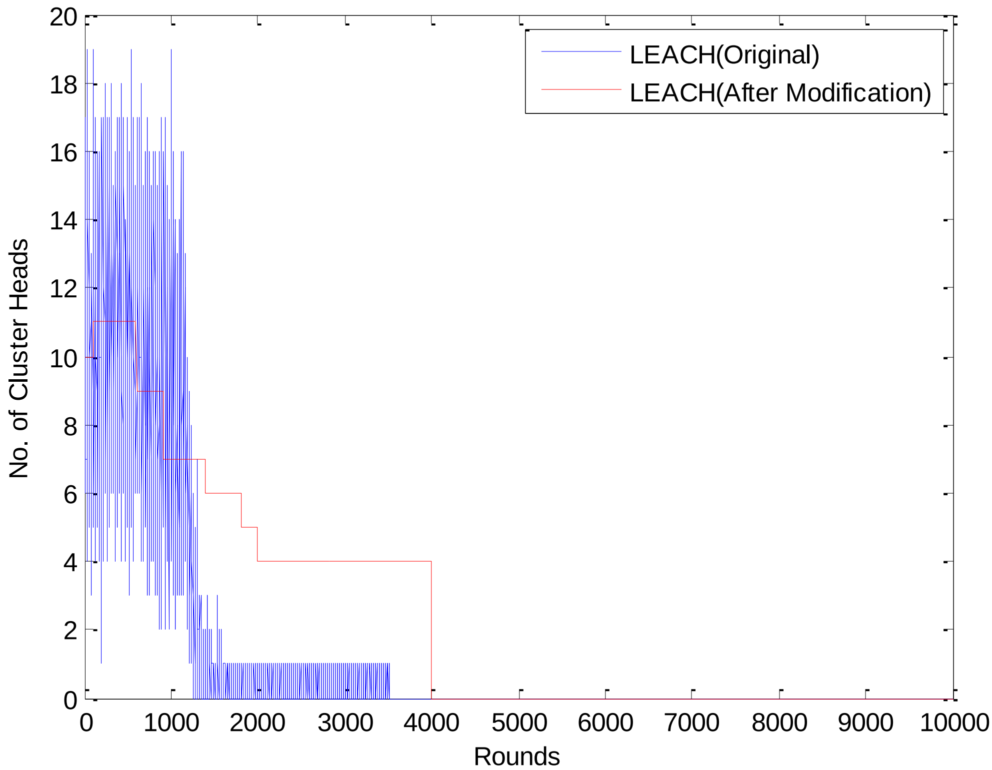


**Fig. 3.6 No. of CHs per Round in LEACH**

## 3.6 Conclusion

In this work we proposed two changes in existing probabilistic clustering algorithms to ensure more or less optimal number of clusters and well distributed CHs across the network in each round. We also proposed two ways to apply suggested changes to existing clustering algorithm. Simulation results show a considerable improvement in the performance of existing algorithms after applying the suggested changes. In this work we focused on probabilistic clustering algorithms but in future we can extend this to make it applicable to other classes of clustering Algorithms.

.

# Chapter 4: Discrete PSO Based Clustering Algorithm

## 4.1 Objective

Clustering is a widely used mechanism in wireless sensor networks to reduce the energy consumption by sensor nodes in data transmission. Partitioning the network into optimal number of clusters and selecting an optimal set of nodes as cluster heads is an NP-Hard problem. The NP-Hard nature of clustering problem makes it a suitable candidate for the application of evolutionary algorithms and particle swarm optimization (PSO). In this work, we shall suggest a PSO based solution to the optimal clustering problem by using residual energy and transmission distance of sensor nodes to define a fitness function. We will modify the existing PSO that assumes a continuous search space to create a new PSO that can work with discrete search space.

## 4.2 Particle Swarm Optimization

Particle swarm optimization (PSO), developed by Dr. Eberhart and Dr. Kennedy in 1995 and inspired by social behaviour of bird flocking or fish schooling is a population based stochastic technique to solve continuous and discreet optimization problems,.

PSO learned from the scenario and used it to solve the optimization problems. In PSO, each single solution is a "bird" in the search space. We call it "particle". All of particles have fitness values which are evaluated by the fitness function to be optimized, and have velocities which direct the flying of the particles. The particles fly through the problem space by following the current optimum particles [55].

PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generations. In every iteration, each particle is updated by following two "best" values. The first one is the best solution (fitness) it has achieved so far. (The fitness value is also stored.) This value is called pbest. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the population. This best value is a global best and called gbest. When a particle takes part of the population as its topological neighbours, the best value is a local best and is called lbest [56].

Suppose, there is a group of K random particles in an n-dimension searching space, the position of the $i^{th}$ particle is $X_i = (x_{i_1}, x_{i_2}, \ldots \ldots, x_{i_n})$, the personal best value of the particle is $pbest_i = (p_{i_1}, p_{i_2}, \ldots \ldots, p_{i_n})$, and the velocity of the particle is $V_i =$

$(v_{i_1}, v_{i_2}, \dots \dots, v_{i_n})$. The best value obtained so far by any particle in the population is $gbest = (g_1, g_2, \dots \dots, g_n)$. After finding the two best values, $pbest$ and $gbest$ the particle updates its velocity and positions as follows

$$v_{i_j} = w.v_{i_j} + c_1.r_1\left(p_{i_j} - x_{i_j}\right) + c_2.r_2\left(g_j - x_{i_j}\right) \tag{4.1}$$

$$x_{i_j} = x_{i_j} + v_{i_j} \tag{4.2}$$

Where w is inertia and used to control the trade-off between the global and the local exploration ability of the swarm, $c_1$ and $c_2$ are learning factors, $r_1$ and r2 random numbers between 0 and 1.

## 4.3 Proposed Algorithm

In this section we describe in detail the working of our proposed algorithm. We assume a wireless sensor network with sensor nodes uniformly distribute across the network. We also assume that location of Base station is fixed inside or outside the sensor network and location of sensor nodes is also known to base station.

### 4.3.1 Fitness Function

Success of our proposed algorithm will depend greatly on the formulation of fitness function. So we are defining a fitness function that includes all optimization criteria. Our aim is to minimize the intra-cluster communication energy and energy loss due to cluster head and base station communication, so we can define the fitness of a particle i as

$$F(P_i) = E_1(P_i) + \mu\, E_2(P_i) \tag{4.3}$$

$$E_1(P_i) = \sum_{k=1}^{K} \sum_{\forall n_{k_j} \in C_k} \frac{f\left(n_{k_j}, CH_k\right) - E_{min}}{E_{max} - E_{min}} \tag{4.4}$$

$$E_2(P_i) = \sum_{k=1}^{K} \frac{g(CH_k, BS) - E_{min}}{E_{max} - E_{min}} \tag{4.5}$$

$$f\left(n_{k_j}, CH_k\right) = \begin{cases} s^2\left(n_{k_j}, CH_k\right) & if\ s\left(n_{k_j}, CH_k\right) \le d_0 \\ s^4\left(n_{k_j}, CH_k\right) & if\ s\left(n_{k_j}, CH_k\right) > d_0 \end{cases} \tag{4.6}$$

$$g(CH_k, BS) = \begin{cases} d^2_{CH_k,BS} & if\ d_{CH_k,BS} \le d_0 \\ d^4_{CH_k,BS} & if\ d_{CH_k,BS} > d_0 \end{cases} \tag{4.7}$$

$$s(n_i, CH_k) = \begin{matrix} \min(s_{n_i,CH_k}) \\ \forall k = 1,2,\dots,K \end{matrix} \tag{4.8}$$

Where, $d_{i,j}$ is the distance between node $i$ and node $j$; $s$ is a function that find the minimum distance cluster head for a given node; $f$ is a function whose value for a given node is proportional to the energy consumed in communication between the node and its cluster head; similarly g signifies the energy loss due to cluster head and base station communication; $E_{max}$ and $E_{min}$ are the maximum and minimum energy loss in the network. $C_k$ is $k^{th}$ cluster in a solution or particle.

$E_1$ and $E_2$ are two normalized functions that represent the energy dissipated in intra-cluster communication and due to communication between sink and CHs respectively. F is fitness function and our aim is to minimize this function.

$\mu$ is a controlling parameter that control the distance between base station and cluster heads. The higher the value of $\mu$ will be the closer will be the CHs from BS. $K$ is the optimal number of cluster heads.

For each particle or solution we will choose k random nodes as cluster heads and remaining nodes will join the cluster whose CH is at minimum distance from it. Then we will evaluate the value of fitness function for each particle and will calculate $pbest$ and $gbest$. Then we will update the velocity vector and position vector according to equation (4.1) and (4.2).

### 4.3.2 A new operator $\bigoplus_{NW}$

We will define a new operator $\bigoplus_{NW}$ that when applied on a location with respect to a network, will return a valid sensor node location in the network. In each iteration of our algorithm we will update the location of CHs in each particle or solution. Keeping this into consideration we define $\bigoplus_{NW}$ as follows:

Suppose $\dot{a} = (a_1, a_2)$ is any location with respect to a sensor network $NW$ then $\bigoplus_{NW} \dot{a}$ will return a valid location in network $NW$. The operator $\bigoplus_{NW}$ will first check if $\dot{a}$ is a valid location in network is. If $\dot{a}$ is a valid location than it return $\dot{a}$ as it is; if not then it will return nearest valid location in the network $NW$ toward base station with highest residual energy. After calculating new velocity and position using equation (4.1) and (4.2) we will apply our operator to the calculated positions to get valid new positions.

### 4.3.3 Working of proposed PSO Algorithm

1. Create and initialize a K-dimensional swarm of P particles by choosing K CHs with residual energy higher than average energy of network for each particle.
2. **repeat**
3.     **for** each particle $i = 1,2,\dots,P$ **do**
4.         **if** $F(X_i) < lbest_i$ **then**
5.             $lbest_i = X_i$
6.         **end**
7.         **if** $F(X_i) < gbest$ **then**
8.             $gbest = X_i$
9.         **end**
10.     **end**
11.     **for** each particle $i = 1,2,\dots,P$ **do**
12.         update velocity $V_i$ using equation (4.1)
13.         update position vector $X_i$ using equation (4.2)
14.         apply $\oplus_{NW}$ operator to updated position
15.     **end**
16. **until** the maximum number of iteration reached

**Fig. 4.1 Pseudo Code for PSO Based Clustering**

## 4.5 Simulation Results

For simulation we assume a square network field of size 100m X 100m with 100 sensor nodes deployed uniformly in it. We assume that sink is at the centre of the field. We will compare the performance of our proposed algorithm with LEACH and its popular variant LEACH-C. We are using same simulation parameters are described in Table 3.1 in chapter 3. Figure 4.2 shows the no. of alive nodes in each round of LEACH, LEACH-C and our proposed protocol PSOBC. Simulation results show a considerable improvement in network lifetime. We compare all protocols for different positions of base station.

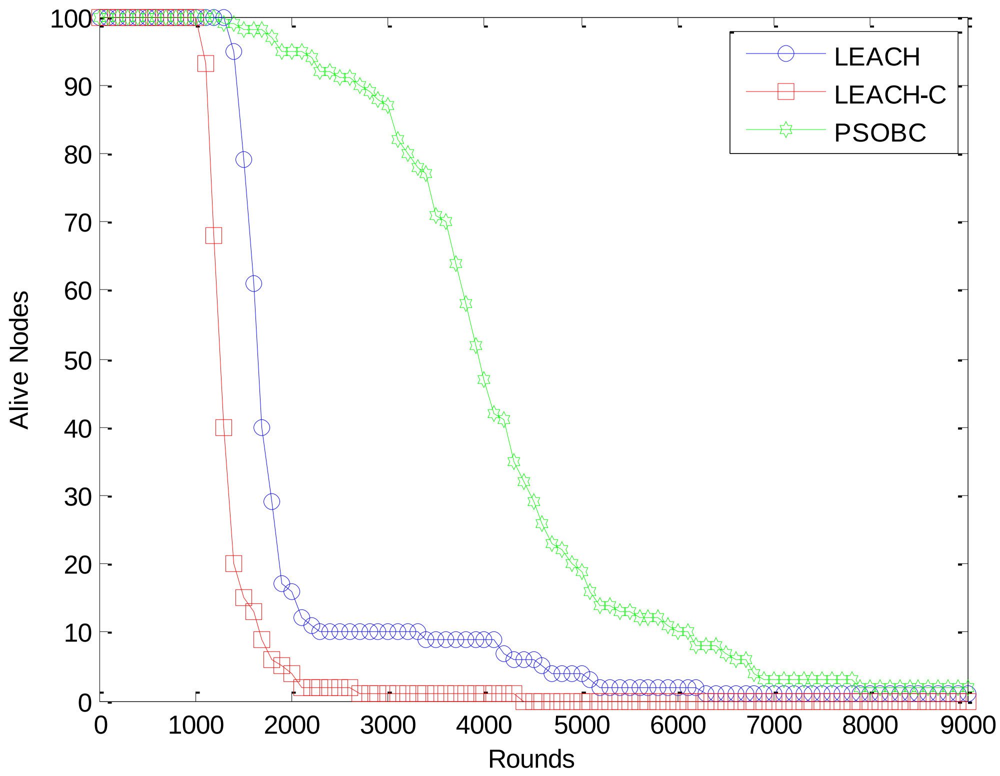


**Fig. 4.2 Alive Nodes per Round for BS position (50, 0)**

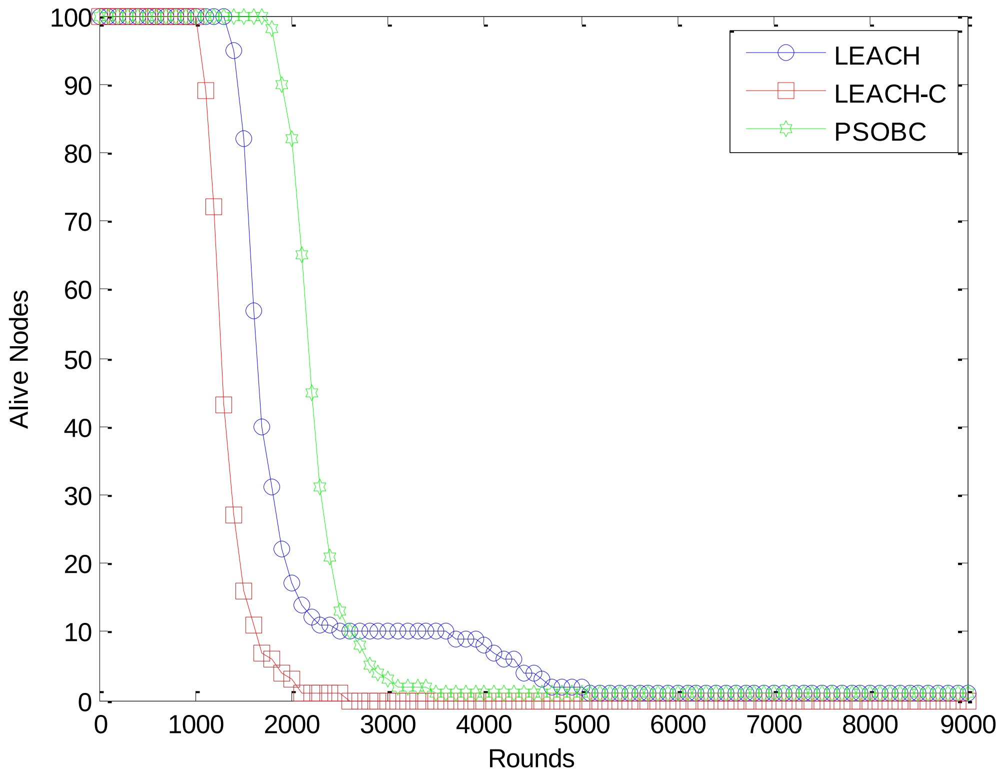


**Fig. 4.3 Alive Nodes per Round for BS position (0, 50)**

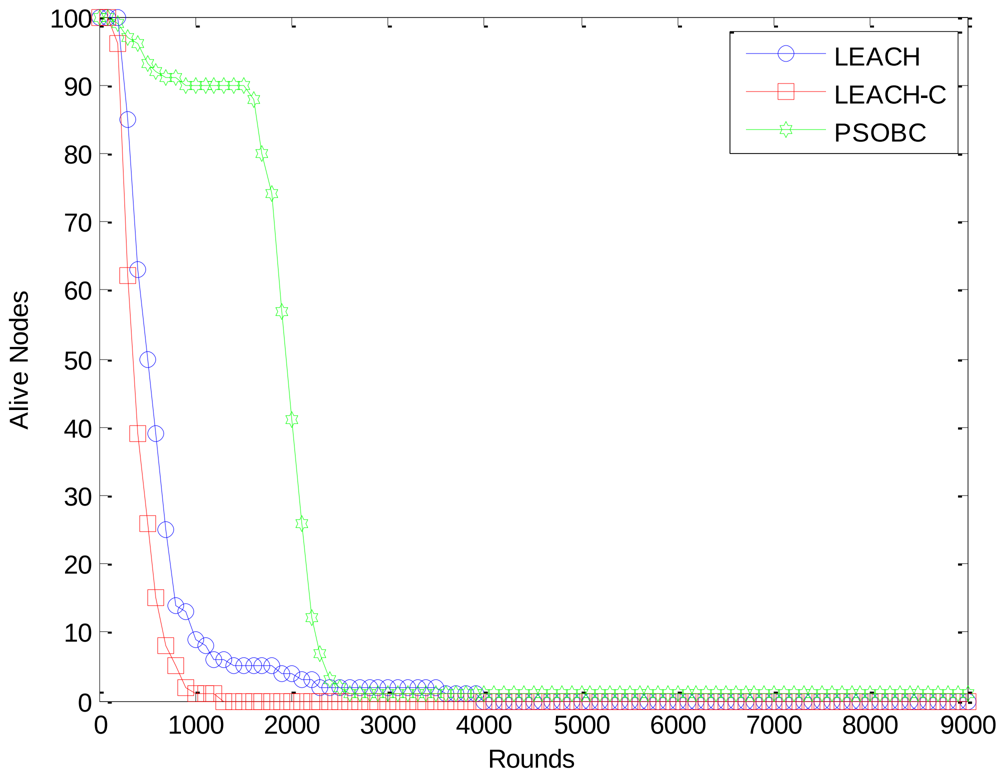


**Fig. 4.3 Alive Nodes per Round for BS position (200, 200)**

## 4.6 Conclusion

In this work we proposed a PSO based solution to clustering problem. We used same PSO algorithm that is used for continuous search space with little modification. We defined a new operator and used it with original PSO algorithm to make it work with discrete search space. Simulation results show a considerable increment in Network lifetime as compared to LEACH and LEACH-C. The main drawback of this easy and efficient solution is that it requires the presence of a central authority for cluster setup but it is not always possible in practical applications. We can use base station as central authority if it is not power constrained. The basic idea here was to optimize intra cluster communication energy and energy loss due to communication between CHs and base station by using PSO and by using base station as centralized authority for cluster set up in the network.

# Chapter 5: Energy Efficient Hybrid Clustering Protocol for Heterogeneous Wireless Sensor Networks

## 5.1 Objective

In this work, we will present a new hybrid Clustering protocol for Multi-level heterogeneous wireless sensor networks. In this protocol, some nodes transmit data directly to base station while some use clustering technique to send data to base station. The proposed protocol aims to conserve energy by keeping three key design factors into consideration: (1) the optimal distance up to which a node can directly send the data to base station, (2) electing an appropriate node as cluster head and (3) limiting the number of clusters in the network. The term heterogeneity here refers to the difference in the energy levels of the sensor nodes. In this protocol we will use the best available solutions to the problems like clustering and routing in the literature to create a new protocol that can perform reasonably well in comparison to existing protocols. We will also perform a rigorous simulation study to evaluate the performance of our proposed protocol. We will also compare the performance of our protocol with homogeneous protocol LEACH and heterogeneous protocol SEP.

## 5.2 Motivation and Related Work

Most of the early routing protocols proposed for wireless sensor networks do not consider heterogeneity in the network and therefore are not able to take advantage of the heterogeneity present in the network. This heterogeneity may present in network either from very start of the network or may occur as a result of network operations as network evolves in time. First Georgios Smaragdakis et al. studied the impact of heterogeneity in wireless sensor networks and proposed stable election protocol [52]. SEP judiciously consumes extra energy from the nodes having high energy and increase the stability period and life time of the network.

### 5.2.1 Optimal Number of Clusters

Optimal number of clusters $K_{opt}$ can be found using simple analysis as in [11]. Let us assume an area of M X M square meters with base station situated at the centre and N sensor nodes are distributed over this area.

The energy dissipated in the cluster head node assuming its distance from BS less than $d_0$, during a round can be given by the following formula:

$$E_{CH} = \left(\frac{N}{k} - 1\right) L.E_{elec} + \frac{N}{k}.L.E_{DA} + L.E_{elec} + L.\varepsilon_{fs}.d_{toBS}^2 \tag{5.1}$$

Where k is the number of clusters, $E_{DA}$ is the data aggregation cost of a bit per report to the base station, and $d_{toBS}$ is the average distance between the cluster head and the base station. The energy used in a non-cluster head node is equal to:

$$E_{nonCH} = L.E_{elec} + L.\varepsilon_{fs}.d_{toCH}^2 \tag{5.2}$$

Here $d_{toCH}$ is the average distance between a cluster member and its cluster head.
Thus energy dissipated in a cluster per round:

$$E_{cluster} \approx E_{CH} + \frac{N}{k}.E_{nonCH} \tag{5.3}$$

The total energy dissipated in the network is equal to:

$$E_{tot} = L(2.N.E_{elec} + N.E_{DA} + \varepsilon_{fs}(k.d_{toBS}^2 + N.d_{toCH}^2) \tag{5.4}$$

According to [7]:

$$\mathrm{d}_{\mathrm{toCH}}^2 = \int_{x=0}^{x=x_{max}} \int_{y=0}^{y=y_{max}} (x^2 + y^2)\rho(x,y)dxdy = \frac{M^2}{2\pi k} \tag{5.5}$$

And

$$d^2_{toBS} = \oint_A \sqrt{x^2 + y^2}\frac{1}{A}.dA = 0.765\frac{M}{2} \tag{5.6}$$

Differentiating $E_{tot}$ with respect to k and equating to zero, the optimal number of constructed clusters can be found:

$$K_{opt} = \sqrt{\frac{N}{2\pi}}\frac{M}{d_{toBS}^2}\sqrt{\frac{\epsilon_{fs}}{\epsilon_{mp}}} \tag{5.7}$$

## 5.3 Proposed Protocol

In this section we will discuss in detail the functioning of our proposed protocol.

### 5.3.1 Network Deployment Model

Most of the routing protocols for heterogeneous WSNs consider the random deployment of sensor nodes in the network space but to take advantage of heterogeneity present in the network in term of energy level we partitioned the network space into two zones: α- zone and β – zone.

This protocol assumes three levels of heterogeneity with three types of nodes:

a) α-nodes
b) β-nodes
c) ω-nodes

α-nodes have the highest energy among all the nodes, β-nodes have energy less than α-nodes but greater than ω-nodes and ω-nodes are normal nodes. We assume that a small fraction of total nodes are α- or β-nodes while most of the nodes are ω-node. As in SEP, we consider that the base station is not mobile and coordinates of the base station are known. The area in network space near the base station is β-zone and β-nodes are deployed in this region while the α-nodes are deployed in the farthest region, i.e. α-zone and ω-nodes are randomly deployed in the whole network space.

The main reason behind this type of deployment is that nodes at the farthest positions from base station in network space require more energy to transmit the data to base station.

### 5.3.2 Cluster Head Selection

Let us assume that A is fraction of total nodes that are α-nodes and B is the fraction of total nodes that are β-nodes. The energy of a α-node is α times higher than an ω node while the energy of a β-node is β time higher than an ω node. If initial energy of an ω-node is $E_0$ then total energy of all the nodes will be:

$$\begin{aligned} E_{total} &= N(1 - A - B)E_0 + N.A.E_0(1 + \alpha) + N.B.E_0(1 + \beta) \\ &= N.E_0(1 + A.\alpha + B.\beta) \end{aligned} \tag{5.8}$$

Optimal probability of cluster head selection in case of homogeneous network is given by [4]:

$$P_{opt} = \frac{k_{opt}}{N} \tag{5.9}$$

Where $K_{opt}$ is optimal number of cluster heads as calculated above and N is total number of nodes in network.

Now optimal probability of a node to be cluster head on the basis of residual energy can be calculated as:

$$(P_\omega)_i = \frac{P_{opt}\, E_i(r)}{(1+A\alpha+B\beta)\bar{E}(r)} \tag{5.10}$$

$$(P_\beta)_i = \frac{(1+\beta)P_{opt}\, E_i(r)}{(1+A\alpha+B\beta)\bar{E}(r)} \tag{5.11}$$

$$(P_\beta)_i = \frac{(1+\beta)P_{opt}\, E_i(r)}{(1+A\alpha+B\beta)\bar{E}(r)} \tag{5.12}$$

Here $E_i(r)$ is residual energy of $i^{th}$ node in $r^{th}$ round and $\bar{E}(r)$ is the average energy in the $r^{th}$ round.

Depending on the weighted probabilities the threshold values can be calculated as follows

$$T_\omega = \begin{cases} \frac{P_\omega}{1-P_\omega\left(1-r \bmod \frac{1}{P_\omega}\right)} & if\ \omega \in G'' \\ 0 & otherwise \end{cases} \tag{5.13}$$

$$T_\beta = \begin{cases} \frac{P_\beta}{1-P_\beta\left(1-r \bmod \frac{1}{P_\beta}\right)} & if\ \beta \in G' \\ 0 & otherwise \end{cases} \tag{5.14}$$

$$T_\alpha = \begin{cases} \frac{P_\alpha}{1-P_\alpha\left(1-r \bmod \frac{1}{P_\alpha}\right)} & if\ \omega \in G \\ 0 & otherwise \end{cases} \tag{5.15}$$

Where G, G' and G'' are the sets of α, β, ω-nodes that have not been the cluster head in last epoch respectively.

Each node generates a number in interval [0, 1] randomly. If this random number is less then corresponding threshold the node will become cluster head.

Once the cluster head is selected, the cluster head broadcasts an advertisement message to all the nodes. A node that receives such message decides on the basis of received signal strength that to which cluster head it will associate for the current round.

### 5.3.3 Data Transmission

In this protocol we use two techniques for data transmission:

a) Single-hop direct transmission

b) Multi-hop transmission through cluster heads

In single hop transmission a node n near the base station will directly send the data to base station if

$$d_{n\ to\ BS} < d_0/_{k'}$$

and

Residual energy $E_n(r) \geq \bar{E}\ (r)$

Here $d_{n\ to\ BS}$ is distance between nth node and base station; $k'$ is a parameter which is used to control the single hop transmissions; $E_n(r)$ is residual energy of nth node in rth round; $\bar{E}\ (r)$ is average energy of whole network in rth round.

If the above two conditions do not satisfy simultaneously, the node will send data to cluster head for further processing. Each cluster head create a schedule based on TDMA and nodes send data to their cluster head at their respective time slots as described in the schedule.

We used Dijkastra's shortest path algorithm to find the shortest route from a cluster head to base station through other cluster heads. For this we used the distance of nodes from each other and base station as weights. We can also use any predefined QoS metric as weight with distance.

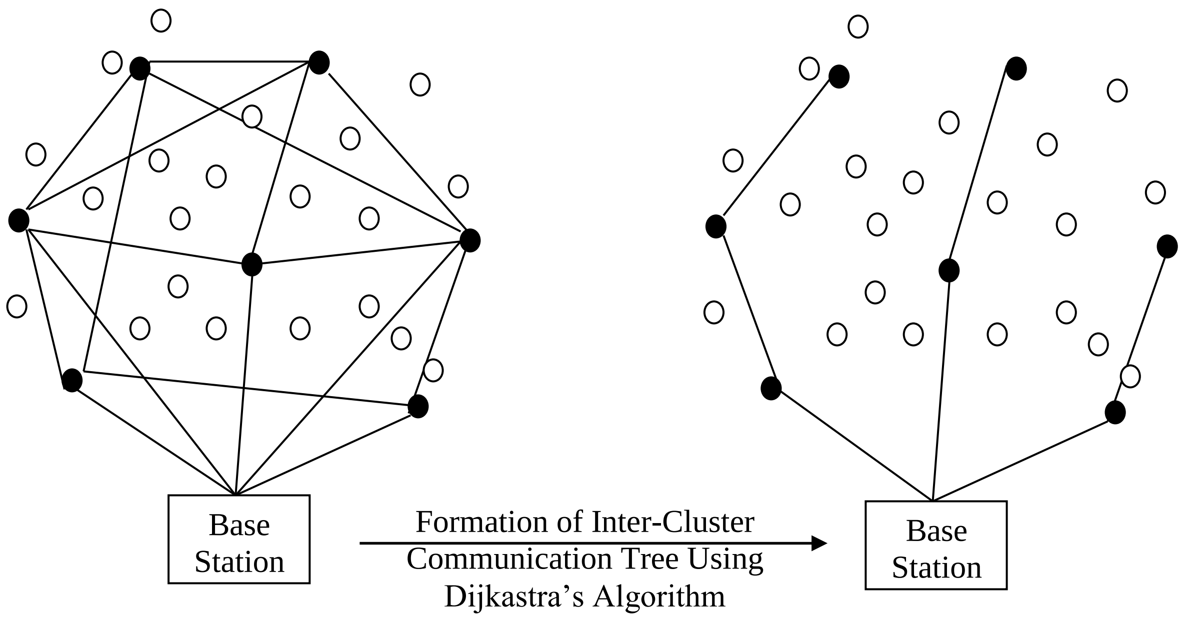


**Fig. 5.1 Formation of Inter-Cluster Communication Tree Using Dijkastra's Algorithm**

## 5.4 Simulation Results

For simulation we assume a square network field of dimension 100m X 100m with 100 sensors deployed in it. 20m X 20m area centred at (50, 50) is normal-zone and the area of width 20m surrounding the normal zone is β-zone and β-sensors have been deployed in this zone randomly. Similarly the reaming area of width 20m that surrounds the β-zone is α-zone and α-sensors are deployed in the α-zone remaining ω-sensors are deployed randomly in whole network field. Matlab is used for the purpose of simulation.

Following metrics is used to evaluate the performance of the protocol relative to LEACH and SEP:

a) *Stability Period:* It is the time span between network boot time and first dead node.
b) Number of nodes alive per round.
c) Number of Cluster heads formed in each round.
d) Throughput of the network.

Simulation parameters are shown in the Table 1.

| **DESCRIPTION** | **PARAMETER** | **VALUE** |
|---|---|---|
| Initial energy of normal nodes | $E_0$ | 0.5 *J* |
| No. of $\alpha$ nodes as fraction of total nodes | $A$ | 0.1, 0.2 |
| No. of $\beta$ nodes as fraction of total nodes | $B$ | 0.1, 0.2 |
| Energy coefficient for $\alpha$ nodes | $\alpha$ | 2 |
| Energy coefficient for $\beta$ nodes | $\beta$ | 1, 1.5 |
| Data aggregation energy | $E_{DA}$ | 5 *nJ/bit/signal* |
| Electronic circuitry energy | $E_{elec}$ | 50 *nJ/bit* |
| Free space co-efficient | $\varepsilon_{fs}$ | 10 *Pj/bit/m2* |
| Multi-path co-efficient | $\varepsilon_{mp}$ | 0.013 *pJ/bit/m4* |
| Optimal percentage of CHs | $P_{opt}$ | 0.1 |
| Initial Energy of $\alpha$ nodes | $E_\alpha$ | $E_0$ (1+α) |
| Initial Energy of $\beta$ nodes | $E_\beta$ | $E_0$ (1+ β ) |
| Total no. of nodes | $N$ | 100 |

**Table 5.1: Simulation Parameters**

We performed simulation using different values of α, β, A and B.

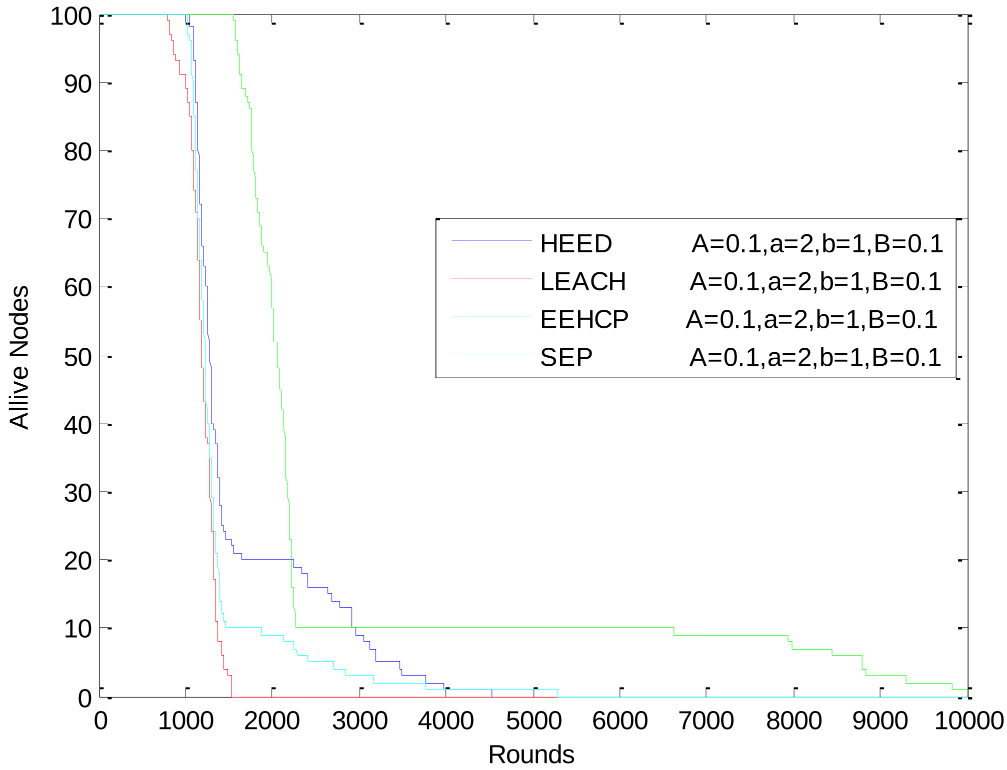


**Fig. 5.2 No. of Alive Nodes per Round for A = 0.1, α = 2, β = 1, B = 0.1**

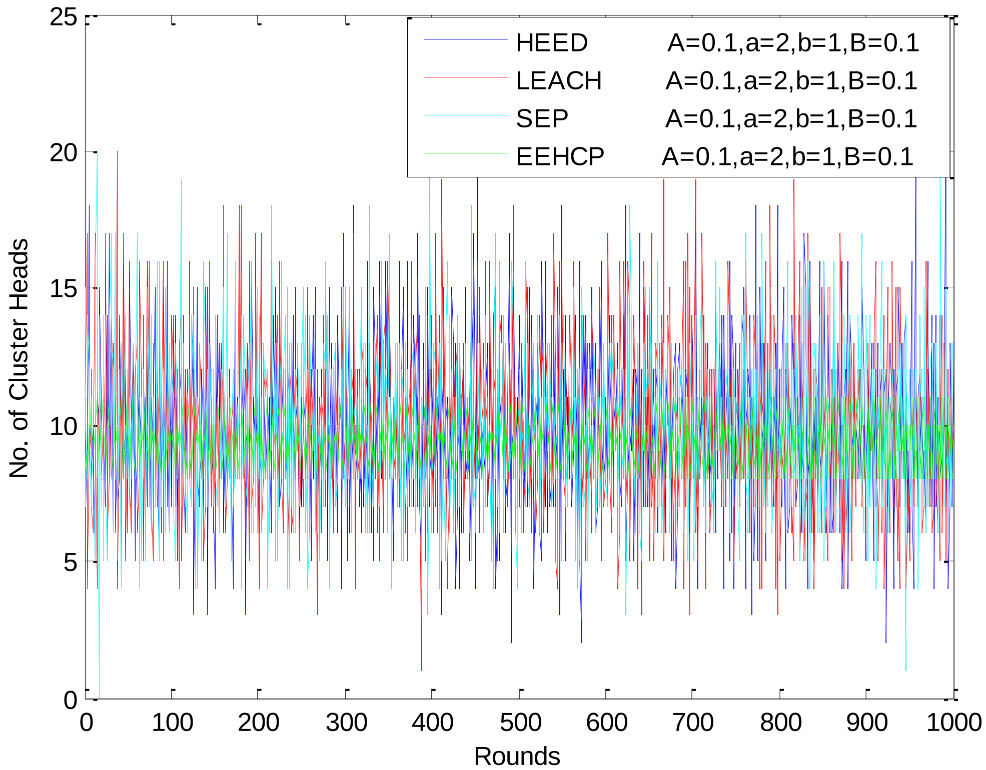


**Fig. 5.3 No. of CHs per Round for A = 0.1, α = 2, β = 1, B = 0.1**

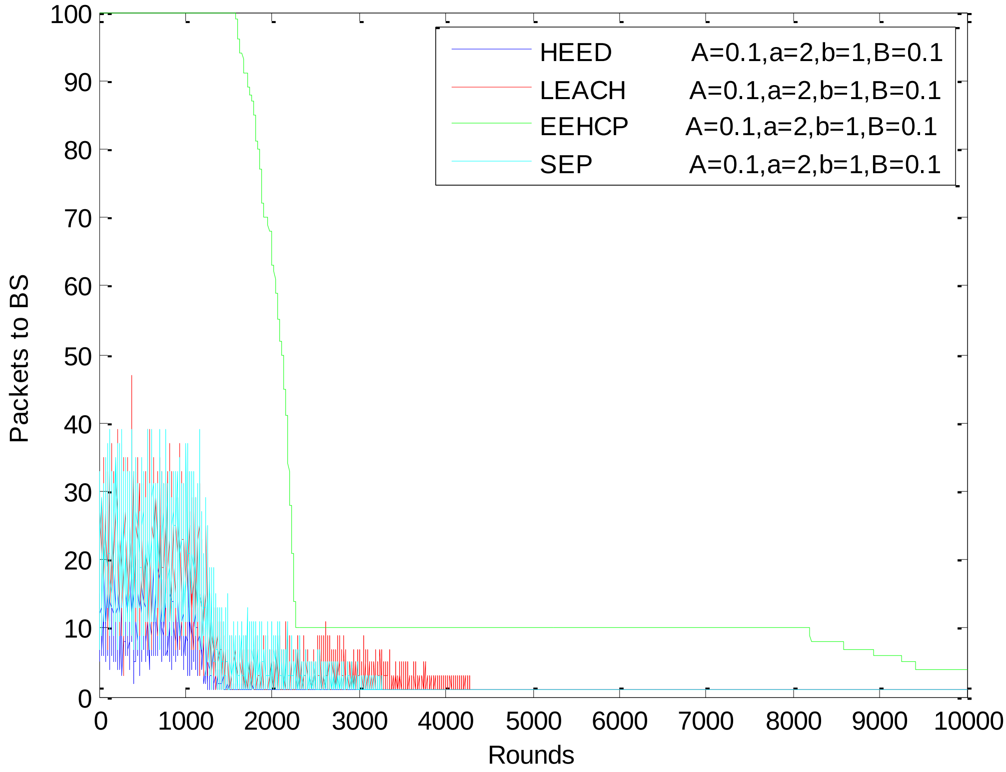


**Fig. 5.3 Throughput for A = 0.1, α = 2, β = 1, B = 0.1**

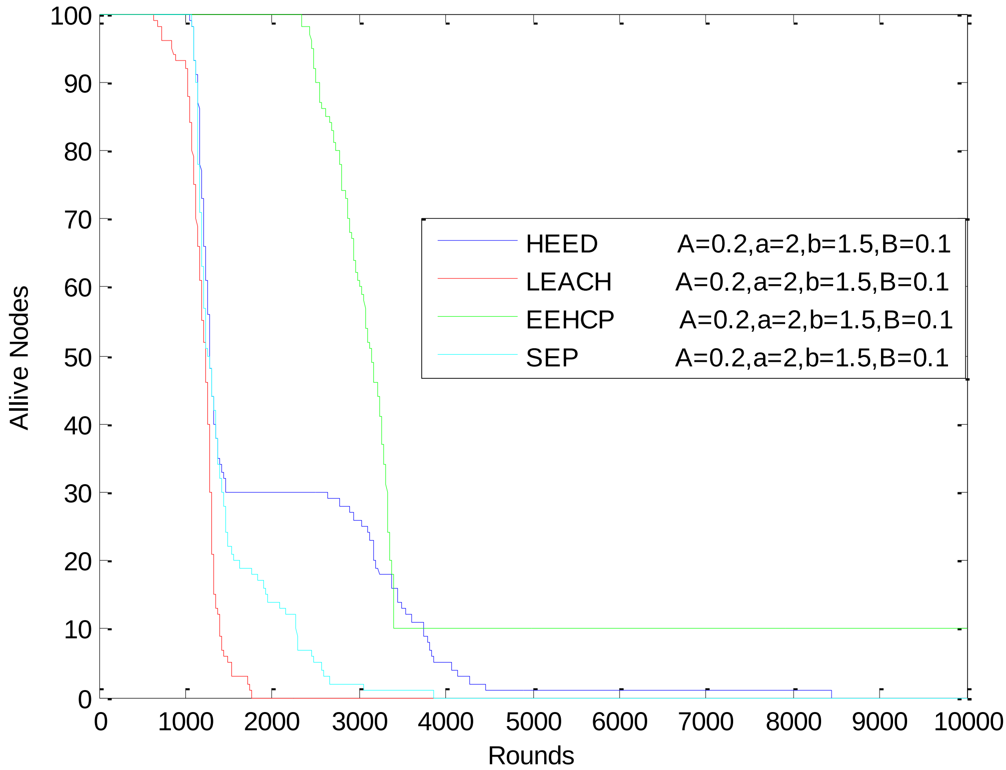


**Fig. 5.4 No. of Alive Nodes per Round for A = 0.2, α = 2, β = 1.5, B = 0.1**

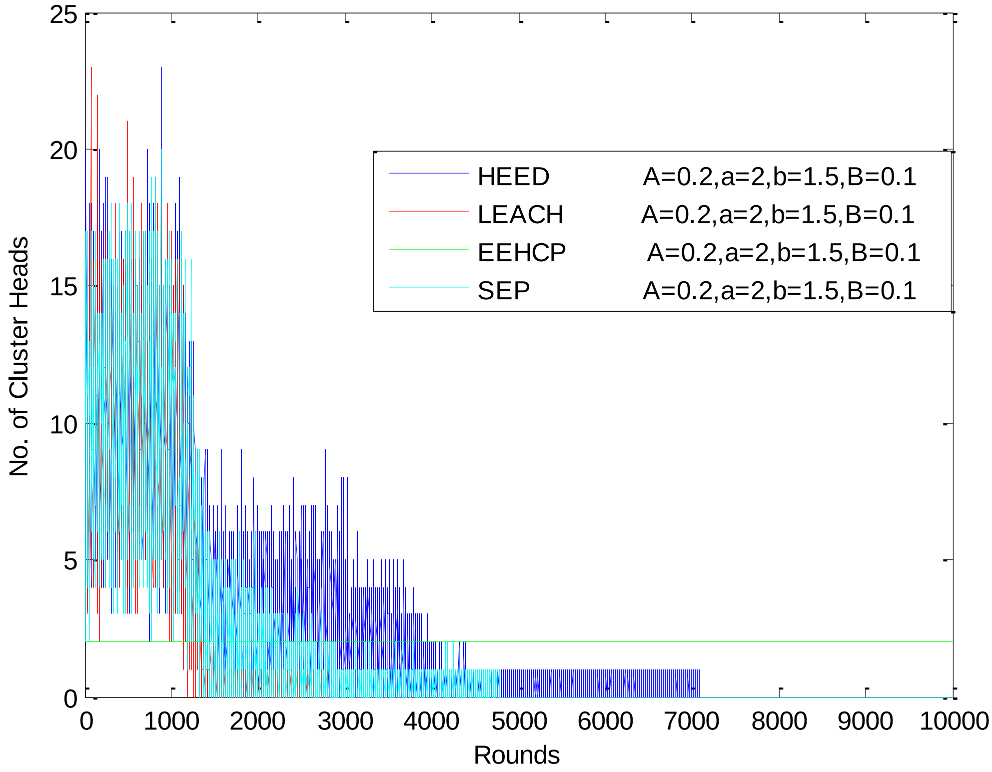


**Fig. 5.5 No. of CHs per Round for A = 0.2, α = 2, β = 1.5, B = 0.1**

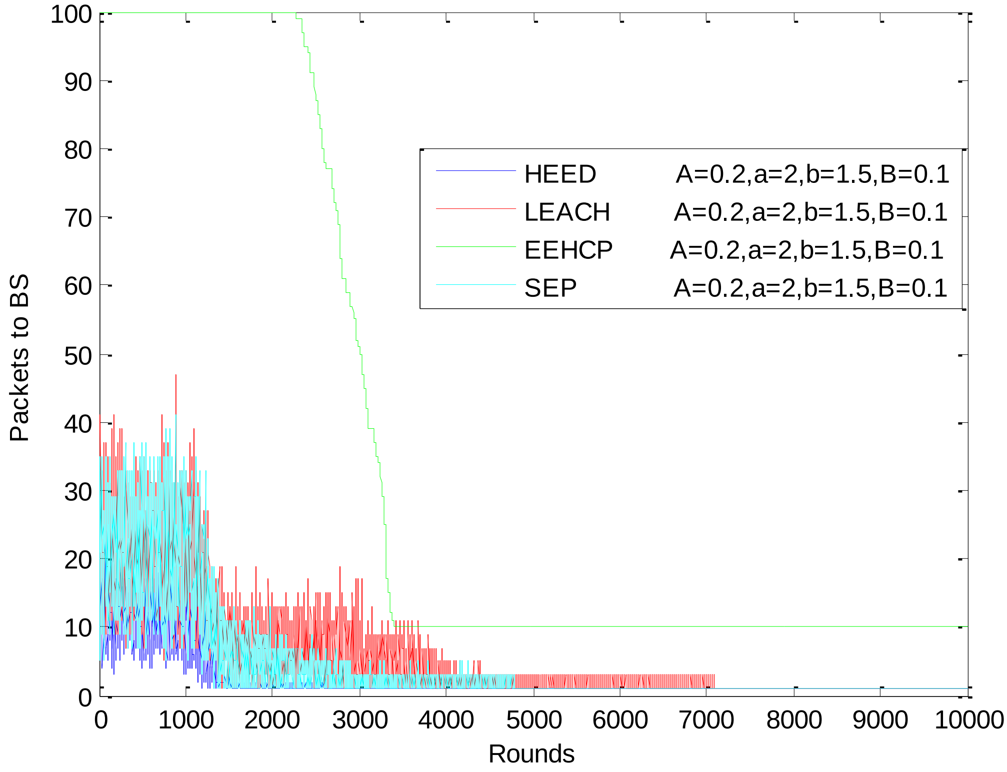


**Fig. 5.6 Throughput for A = 0.2, α = 2, β = 1.5, B = 0.1**

The above figure shows a comparison of HEED, LEACH, SEP and EEHCP for different numbers and energy levels of α, β and ω nodes. In each case EEHCP outperform LEACH and SEP. We can see in above figure that EEHCP prolongs the stability period and network life time. EEHCP prolongs the stability period of network by approximately 75% and almost doubles the lifetime of network in comparison to stable election protocol.

## 5.5 Conclusion

In this work we describe a hybrid cluster head selection protocol in detail that use the heterogeneity in sensor node for an intelligent deployment of nodes in the network and use the residual energy of nodes in particular data transmission round to weight the optimal probability of cluster head selection. The simulation results show that proposed EEHCP performs better than protocols like SEP, LEACH and HEED and prolong the stability period and Network throughput. In future we can extend the protocol to work with a mobile base station. Also we can extend the protocol to consider the optimal probability wait change when residual energy of a node with higher initial energy become equal to the residual energy of a normal node.

# REFERENCES


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

[2] K. Sohrabi et al., "Protocols for self-organization of a wireless sensor network, *IEEE Personal Communications"*, 7(5), 16–27, 2000.

[3] R. Min et al., "Low power wireless sensor networks", in *Proceedings of International Conference on VLSI Design*, Bangalore, India, January 2001.

[4] M. Liliana, C. Arboleda, N. Nidal, "Comparison of clustering algorithms and protocols for wireless sensor networks", in *Proceedings of IEEE CCECE/CCGEI Conference*, Ottawa, Ontario, Canada, pp. 1787–1792, May 2006.

[5] G. Gupta and M. Younis, "Load-balanced clustering in wireless sensor networks", in *Proceedings of the International Conference on Communication (ICC 2003)*, Anchorage, AK, May 2003.

[6] E. I. Oyman and C. Ersoy, "Multiple sink network design problem in large scale wireless sensor networks", in *Proceedings of the IEEE International Conference on Communications (ICC 2004)*, Paris, June 2004.

[7] D. J. Baker and A. Ephremides, "The architectural organization of a mobile radio network via a distributed algorithm", *IEEE Transactions on Communications*, 29(11), 1694–1701, 1981.

[8] A. Ephremides, J.E. Wieselthier, and D.J. Baker, "A design concept for reliable mobile radio networks with frequency hopping signalling", *Proceedings of IEEE*, 75(1), 56–73, 1987.

[9] C. R. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks", *IEEE Journal on Selected Areas Communications*, 15(7), 1265–1275, 1997.

[10] W. P. Chen, J. C. Hou, and L. Sha, "Dynamic clustering for acoustic target tracking in wireless sensor networks", in *11th IEEE International Conference on Network Protocols (ICNP'03)*, pp. 284–294, 2003.

[11] W. R. Heinzelman, A.P. Chandrakasan, and H. Balakrishnan, "Energy efficient communication protocol for wireless micro sensor networks", in *Proceedings of the 33rd Hawaaian Interantional Conference on System Sciences*, January 2000.

[12] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application specific protocol architecture for wireless microsensor networks, *IEEE Transactions on Wireless Communications"*, 1(4), 660–670, 2002.

[13] S. Bandyopadhyay and E. Coyle, “An energy efficient hierarchical clustering algorithm for wireless sensor networks”, in *22nd Annual Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM 2003)*, San Francisco, CA, April 2003.

[14] 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, 2004.

[15] V. Loscri, G. Morabito, and S. Marano, “A two-level hierarchy for low-energy adaptive clustering hierarchy”, *Proceedings of IEEE VTC Conference 2005*, Vol. 3, pp. 1809–1813, 2005.

[16] A. Youssef, M. Younis, M. Youssef, and A. Agrawala, “Distributed formation of overlapping multi-hop clusters in wireless sensor networks”, in *Proceedings of the 49th Annual IEEE Global Communication Conference (Globecom06)*, San Francisco, CA, Nov. 2006.

[17] S. Selvakennedy and S. Sinnappan, “An adaptive data dissemination strategy for wireless sensor networks”, *International Journal of Distributed Sensor Networks*, 3(1), 2007.

[18] M. Ye, C. Li, G. Chen, and J. Wu, “EECS: An energy efficient clustering scheme in wireless sensor networks”, in *Proceedings of IEEE Intl. Performance Computing and Communications Conference (IPCCC)*, pp. 535–540, 2005.

[19] Y. Jin, L. Wang, Y. Kim, and X. Yang, “EEMC: An energy-efficient multi-level clustering algorithm for large-scale wireless sensor networks”, Elsevier, *Computer Networks Journal*, 52, 542–562, 2008.

[20] H. Huang and J. Wu, “A probabilistic clustering algorithm in wireless sensor networks”, in *Proceedings of IEEE 62nd VTC Conference*, Dallas, TX, September 2005.

[21] K. Xu andM.Gerla, “A heterogeneous routing protocol based on a new stable clustering scheme”, in *Proceeding of IEEE Military Communications Conference* (MILCOM 2002), Anaheim, CA, October 2002.

[22] R. Nagpal and D. Coore, “An algorithm for group formation in an amorphous computer”, in *Proceedings of the 10th International Conference on Parallel and Distributed Systems* (PDCS98), Las Vegas, NV, October 1998.

[23] M. Demirbas, A. Arora and V. Mittal, “FLOC: A fast local clustering service for wireless sensor networks”, in *Proc. of Workshop on Dependability Issues in Wireless AdHoc Networks and Sensor Networks (DIWANS04)*, Florence, Italy, June 2004.

[24] G. Li and T. Znati, "RECA: A ring-structured energy-efficient clustering architecture for robust communication in wireless sensor networks", *International Journal Sensor Networks*, 2(1/2), 2007.

[25] H. Zhang and A. Arora, "GS3: Scalable self-configuration and self-healing in wireless networks", in *Proceedings of the 21st ACM Symposium on Principles of Distributed Computing (PODC 2002)*, Monterey, CA, July 2002.

[26] L. Yu, N. Wang, W. Zhang, and C. Zheng, "GROUP: A grid-clustering routing protocol for wireless sensor networks", in *Proceedings of International Conference on Wireless Communications*, Networking and Mobile Computing, 2006.

[27] S. Banerjee and S. Khuller, A clustering scheme for hierarchical control in multi-hop wireless networks, in *Proceedings of 20th Joint Conference of the IEEE Computer and Communications Societies (INFOCOMŠ 01)*, Anchorage, AK, April 2001.

[28] A.K. Parekh, "Selecting routers in ad-hoc wireless networks", in *Proceedings of SBT/IEEE International Telecommunications Symposium*, ITS, 1994.

[29] A. Amis, R. Prakash, T. Vuong, and D. Huynh, "Max-Min D-Cluster formation in wireless Ad Hoc networks", *IEEE INFOCOM*, March 2000.

[30] C. Liu, K. Wu, and J. Pei, "A dynamic clustering and scheduling approach to energy saving in data collection from wireless sensor networks", in *2nd Annual IEEE Conf. on Sensor and Ad Hoc Communications and Networks (SECON'05)*, September 2005.

[31] C. Wen and W. Sethares, "Automatic decentralized clustering for WSNs", *EURASIP Journal on Wireless Communications and Networking*, 5(5), 686–697, 2005.

[32] K. Yanagihara, J. Taketsugu, K. Fukui, S. Fukunaga, S. Hara and K.I. Kitayama, "EACLE: Energy-aware clustering scheme with transmission power control for sensor networks", Springer, *Wireless Personal Communications*, 40(3), 401–415, 2007.

[33] H. Chan and A. Perrig, "ACE: an emergent algorithm for highly uniform cluster formation", in *Proceedings of the 1st European Workshop on Sensor Networks (EWSN)*, Berlin, Germany, January 2004.

[34] S. Lindsey and C.S. Raghavendra, "PEGASIS: Power-efficient gathering in sensor information networks", Computer Systems Research Department, the Aerospace Corporation, 3, pp. 3-1125–3-1130, Los Angeles, CA, 2002.

[35] H.O. Tan and I. Korpeoglou, "Power efficient data gathering and aggregation in wireless sensor networks", *Issue on Sensor Networks Technology, ACM SIGMOD Record*, 32(4), 66–71, 2003.

[36] S. Yi, J.Heo, Y. Cho, and J. Hong, "PEACH: Power-efficient and adaptive clustering hierarchy protocol for WSNs", Elsevier, *Computer Networks*, 30, 2842–2852, 2007.

[37] S. Basagni, "Distributed clustering for Ad Hoc networks", in *Proceedings of Intl. Symp. on Parallel Architectures, Algorithms & Networks (ISPAN)*, pp. 310–315, 1999.

[38] M. Chatterjee, S.K. Das, and D. Turgut, "WCA: A weighted clustering algorithm for mobile ad hoc networks", *Clustering Computing*, 5, 193–204, 2002.

[39] P. Ding, J. Holliday, and A.Celik, "Distributed energy efficient hierarchical clustering for wireless sensor networks", in *Proceedings of the IEEE International Conference on Distributed Computing in Sensor Systems(DCOSS05)*, Marina Del Rey, CA, June 2005.

[40] R. Virrankoski, A. Savvides, "TASC: Topology adaptive clustering for sensor networks", in *Proceedings of the Second IEEE International Conference on Mobile Ad-Hoc and Sensor Systems, (MASS 2005)*, Washington, DC, November 2005.

[41] J. Kamimura, N. Wakamiya, and M. Murata, "A distributed clustering method for energy-efficient data gathering in sensor networks", *Int. J. Wireless and Mobile Computing*, 1(2), 113–120, 2006.

[42] S. Selvakennedy, S. Sinnappan, and Y. Shang, "A biologically inspired clustering protocol for wireless sensor networks", *Computer Communications* 30, 2786–2801, 2007.

[43] N.M. Abdul Latiff, C.C. Tsimenidis, and B.S. Sharif, "Energy-aware clustering for wireless sensor networks using particle swarm optimization", in *IEEE Intl. Symposium PIMRC'07*, pp. 1–5, Athens, Greece, Sept. 2007.

[44] A. Manjeshwar and D.P. Agrawal, "TEEN: A routing protocol for enhanced efficiency in wireless sensor networks", in *Proceedings of the 15th International Parallel and Distributed Processing Symposium*, San Francisco, CA, 2001.

[45] A. Manjeshwar and D.P. Agrawal, "APTEEN: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks", in *Proceedings of the International Parallel and Distributed Processing Symposium*, 2002.

[46] S. Yoon and C. Shahabi, "Exploiting spatial correlation towards an energy efficient Clustered AGgregation technique (CAG)", in *IEEE International Conference on Communications*, pp. 82–98, 2005.

[47] S. Yoon and C. Shahabi, “The Clustered AGgregation (CAG) technique leveraging spatial and temporal correlations in wireless sensor networks”, *ACM Transactions on Sensor Networks*, 3(1), March 2007.

[48] Y. Xu and H. Qi, “Decentralized reactive clustering for collaborative processing in sensor networks”, in *Proceedings of the 10th International Conference on Parallel and Distributed Systems (ICPADS’2004)*, pp. 54–61, July 2004.

[49] B. Guo and Z. Li, “A dynamic-clustering reactive routing algorithm for wireless sensor networks”, *Wireless Networks Journal*, Springer, ISSN 1572-8196 (Online Publication), September 2007.

[50] C.M. Liu and C.H. Lee, “Power efficient communication protocols for data gathering on mobile sensor networks”, in *IEEE 60th VTC,* Los Angeles, CA, pp. 4635–4639, 2004.

[51] X. Zhang, H. Wang, and A. Khokhar, “An energy-efficient data collection protocol for mobile sensor networks”, in *IEEE 64th VTC*,Montreal, Quebec, Canada, pp.1–5, 2006.

[52] G. Smaragdakis, I. Matta & A. Bestavros, “SEP: A stable election protocol for clustered heterogeneous wireless sensor networks”, *in Proceedings of Second International Workshop on Sensor and Actor Network Protocols and Applications (SANPA 2004)*, 2004.

[53] R. C. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory”, *in Proceedings of the Sixth International Symposium on Micromachine and Human Science*, Nagoya, Japan, pp. 39-43, 1995.

[54] J. Kennedy and R. C. Eberhart, “Particle swarm optimization”, *in Proceedings of IEEE International Conference on Neural Networks, Piscataway,* NJ, pp. 1942-1948, 1995.

[55] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer", *in Proceedings of IEEE International Conference on Evolutionary Computation*, pp. 69–73, 1998.

[56] B. Al-kazemi and C. K. Mohan, “Multi-phase discrete particle swarm optimiziation”, *in Proceeding of Fourth International Workshop on Frontiers in Evolutionary Algorithms (FEA 2002),* 2002.

[57] T. Hendtlass, and M. Randall, “A survey of ant colony and particle swarm meta-heuristics and their application to discrete optimisation problems”, *in Proceedings of The Inaugural Workshop on Artificial Life (AL 2001)*, pp. 15-25, 2001.

[58] J. Kennedy, and R. C. Eberhart, “A discrete binary version of the particle swarm algorithm. Proceedings of the World Multiconference on Systemics,Cybernetics and Informatics 1997, Piscataway”, NJ, pp. 4104-4109, 1997.