Prolonging Network Lifetime of Clustered-Wireless Sensor Networks

by

Muattaz Elaneizi

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Mater of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2008

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Wireless Sensor Networking is envisioned as an economically viable paradigm and a promising technology because of its ability to provide a variety of services, such as intrusion detection, weather monitoring, security, tactical surveillance, and disaster management. The services provided by wireless senor networks (WSNs) are based on collaboration among small energy-constrained sensor nodes. The large deployment of WSNs and the need for energy efficient strategy necessitate efficient organization of the network topology for the purpose of balancing the load and prolonging the network lifetime. Clustering has been proven to provide the required scalability and prolong the network lifetime. Due to the bottle neck phenomena in WSNs, a sensor network loses its connectivity with the base station and the remaining energy resources of the functioning nodes are wasted.

This thesis highlights some of the research done to prolong the network lifetime of wireless sensor networks and proposes a solution to overcome the bottle neck phenomena in cluster-based sensor networks. Transmission tuning algorithm for a cluster-based WSNs is proposed based on our modeling of the extra burden of the sensor nodes that have direct communication with the base station. Under this solution, a wireless sensor network continues to operate with minimum live nodes, hence increase the longevity of the system.

An information theoretic metric is proposed as a cluster head selection criteria for breaking ties among competing clusters, hence as means to decrease node reaffiliation , and hence increasing the stability of the clusters, and prolonging the network lifetime. This proposed metric attempts to predict undesired mobility caused by erosion.

Acknowledgment

I would like to express my deep and sincere gratitude to my supervisor, Professor Otman Basir. His wide knowledge and his logical way of thinking have been of great value for me. His understanding, encouraging and personal guidance have provided a good basis for the present thesis.

Special thanks also to all Pami group members, especially; Akrem El-Ghazal and Nabil Drawil for sharing the literature and invaluable assistance. Thanks to Dr,. Ramadan El-Shatshat and Dr. Hu-Pin Han for their invaluable insight and input throughout this research work.

Dedication

This thesis is dedicated in memory of my mother. She was more than a mother; she was a very dear friend. I dedicate this work to her memory. Her strength, endurance, character, friendliness, and love meant a lot to me. She will always be missed. Thank you Mom for the sacrifice of love.

I also dedicate this thesis to my father, who taught me that even the largest task can be accomplished if it is done one step at a time.

To my family, who offered me unconditional love and support throughout the course of this thesis.

Table of Contents

List of Figures

List of Tables

Abbreviations and Symbols

List of Abbreviations

ACK	Acknowledgment
CDMA	Code division multiple access
CTS	Clear to send
DSDV	Destination sequenced distance vector
DSR	Dynamic source routing
LCA	Linked cluster algorithm
FDMA	Frequency division multiple access
LEACH	Low energy adaptive clustering hierarchy
ADV	Advertisement
WCA	Weighted clustering algorithm
EWCA	Entropy-based weighted clustering algorithm
HEED	Hybrid energy efficient distributed
TDC	Time delay-based clustering
VCA	Voting-based clustering algorithm
AODV	Ad hoc on demand distance vector
TORA	Temporally ordered routing algorithm
DD	Directed diffusion
MCFA	Minimum cost forwarding algorithm
TTDD	Tow-tier data dissemination
MAC	Medium access control

MACA	Multiple access collision avoidance
MACAW	Multiple access collision avoidance for wireless
CSMA	Carrier sense multiple access
CSMA/CA	Carrier sense multiple access with collision avoidance
DCF	Distributed coordination function
PCF	Point coordination function
PS	Power saving
PAMAS	Power aware multi-access with signaling
SYNC	Synchronization
CCA	Clear channel assessment
QoS	Quality of service
RTS	Request to send
TDMA	Time division multiple access
WSN	Wireless sensor network
CH	Cluster head
BS	Base station
TEEN	Threshold-sensitive energy efficient
APTEEN	Adaptive periodic threshold-sensitive energy efficient
RDND	Parallel and distributed network dynamics
S-MAC	Sensor medium access control

List of Symbols

E_{TX}	Transmitter’s energy consumption
E_{RX}	Receiver’s energy consumption
E_t	Electronics energy
ε_{fs}	Amplifier energy for short distance
ε_{mp}	Amplifier energy for long distance
$E_{aggregate}$	Energy consumption in the aggregation process
E_{node}	Energy consumption of a member node
E_{CH}	Energy consumption of a cluster head
E_{relay}	Energy consumed to relay a message

Chapter 1

Introduction

A bridge broke into sections and collapsed into the Mississippi River last summer, sending vehicles, concrete, and twisted metal crashing into the water. Police reported more than nine people dead and 60 were seriously wounded [1]. This disaster could have been avoided if the bridge had been equipped with a wireless sensor network (WSN). Such a network would have provided enough lead time to either shut down the bridge or trigger a precautionary maintenance to prevent serious failures.

Wireless sensor networking has been introduced to provide wide-scale connectivity to the physical world at a lower cost than that of the wired alternative. In addition to the cost advantage, wireless sensor networks provide self-configuring, robustness against node failure because of their distributed operation, ease of deployment, high spatial resolution because of their close proximity to the physical phenomena, uniform coverage, low obtrusiveness, and reliable service. They can also be deployed in difficult terrain where placing a wireline network is difficult. In what follows, I briefly define WSNs, showing its benefits, applications, and challenges.

1.1 Wireless Sensor Networks

A wireless sensor network typically consists of a large number of inexpensive, small, low-power communication devices called sensor nodes and one or more computing centres. Advances in energy-efficient design and wireless technologies have enabled the manufacture of the small devices to support several important wireless applications, including real-time multimedia communication [2], medical application, surveillance using WSNs [3,5,6,7], and home networking applications [4,8].In WSNs, the sensor nodes have the ability to sense, process data, and communicate with one another. Figure 1.1 shows the main hardware components of a sensor node: memory that stores programs and intermediate data, a controller that processes all the data and controls the other components, a limited power supply (e.g., battery), a transceiver that performs the functions of both a transmitter and receiver with a limited transmission range, and a sensor device that senses the ambient environment. Figure 1.2 shows examples of modern sensor nodes. Sensor nodes collaborate to detect events or phenomena depending on the application, to collect and process data, and to transmit the sensed information to the computing center (base station) by hopping the data from node to node. Although the sensor nodes individually have limited capabilities, their collaboration to perform a specific task produces an enhanced view of the physical world.

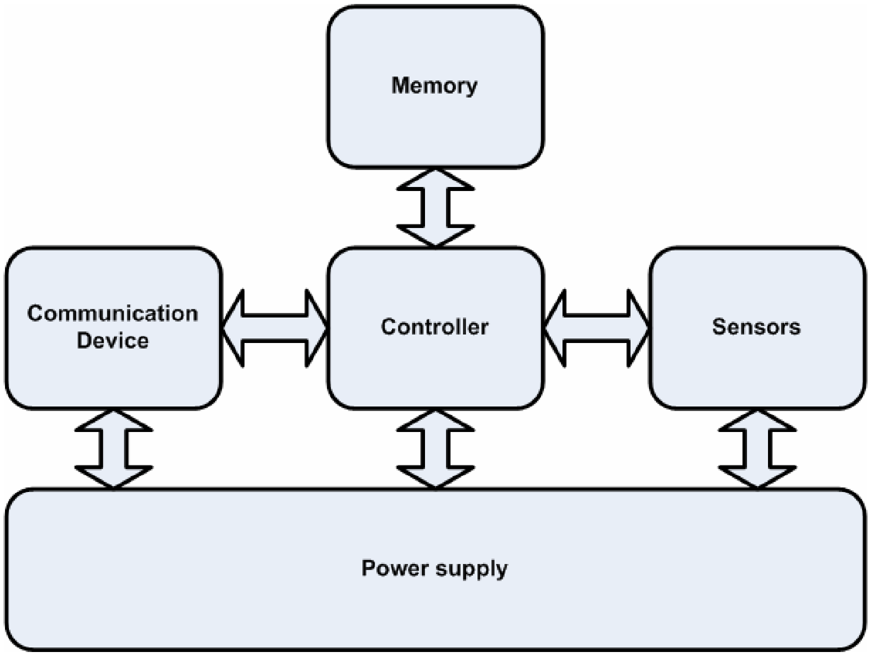


Figure 1.1: Basic sensor node hardware

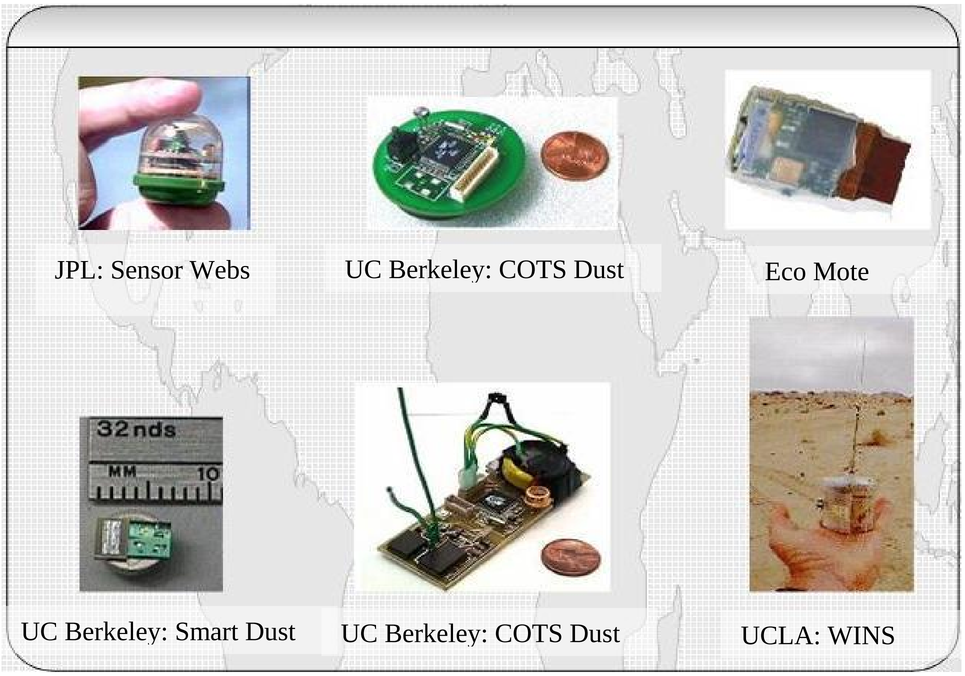

JPL: Sensor Webs
UC Berkeley: COTS Dust
Eco Mote
UC Berkeley: Smart Dust
UC Berkeley: COTS Dust
UCLA: WINS

Figure 1.2: Examples of Sensor nodes

1.1.1 Applications

WSNs can be used in a wide range of exciting applications, such as target field imaging, intrusion detection, weather monitoring, security and tactical surveillance; distributed computing; the detection of ambient conditions such as temperature, movement, sound, and light or the presence of specific objects, inventory control, and disaster management.

Most applications fall into one of four classes: environmental data collection, security monitoring, node tracking, and hybrid networks. In environmental data collection [9,10],a scientist wishes to collect several readings from sensor nodes over a long period of time (i.e. several months or years) in order to detect trends and interdependencies. This type of application is characterized by a large number of randomly deployed nodes continually sensing and transmitting data to a base station that stores the data to be analyzed later. The Great Duck Island project [11] is one example of this class of applications.

Security monitoring in which a moderate number of sensor nodes deployed in predetermined locations. The main difference between security monitoring and environmental monitoring is that the nodes transmit sensed data only when a security violation occurs or a predefined threshold is exceeded. These applications are thus event driven. Node-tracking applications, on the other hand, represent the third class. Unlike the first two classes, in this class of application, the topology continually changes as the targets that are being tracked move through the network. Node-tracking applications are characterized by having a moderate number of sensor nodes that are deployed at fixed locations and send data in an event-driven fashion to base station. The fourth class, hybrid networks contain aspects of the other three classes.

WSNs applications differ widely in their characteristics and QoS requirements. A protocol designed to support one application may not be appropriate for another. Therefore, the design of protocols for such networks should take their diverse characteristics into consideration.

1.1.2 Challenges and Mechanisms

Sensor nodes are resource constrained: They have limited processing speed; storage capacity; communication bandwidth; and most importantly, energy supply. These constraints lead to a number of challenges that must be addressed and problems that must be resolved before these applications can become reality. WSNs require special mechanisms to efficiently utilize the limited resources. In the following, I summarize some of these required mechanisms and challenges.

- Network lifetime is an important of WSN .It is defined as the time until the first node runs out of battery. Due to the limited energy supply (e.g., battery) and the difficulty, if not impossibility, of recharging the battery, WSNs have stringent requirements with respect to power consumption. Therefore, an energy-efficient mechanism is required to save energy and prolong the network lifetime. Other more related lifetime requirements have been used, including the requirement that as long as there is a sensor node alive, the network is considered to be alive.

- Most applications need a large number of nodes to provide good coverage of the sensed phenomena. Since sensor nodes have limited transmission ranges, multi-hop wireless communication is essential for the network nodes to connect with one another. Another requirement is scalability**:** all protocols at different layers must be able to scale to a large number of nodes.

- Central control is an impractical approach for WSNs because it is expensive in terms of energy consumption for each node to send its neighborhood's information to the base station (BS) so that the BS can determine the routing table, sleeping schedule, etc. Sensor networks favour **distributed approach** in which sensor nodes exchange their information locally (one-hop neighborhood). For WSNs to use distributed techniques, the network must be able to configure itself into a connected network.

- The close proximity of the sensor nodes results in redundant information coming from neighboring nodes, i.e., ones with correlated data. This redundancy can be exploited so that the network is able to **tolerate faults** that arise when a node dies or when the wireless link is interrupted. The **reliability** of the information can also be determined with the use of redundancy. In some applications, a single sensor is not able to decide whether an event has happened; instead several sensor nodes must collaborate to detect an event and only the joint data of many sensors provides enough information to be aggregated and produce a reliable measurement. Furthermore, the aggregation process can be done by nodes to produce reliable data and reduce the amount of data that is transmitted to the base station.

1.2 Motivations and Objective

The most essential requirements of WSNs are the ability to scale to hundreds or even thousands of sensor nodes and to operate for a long period of time. Clustering has been proven to be an effective technique that prolongs the network lifetime by reducing energy consumption and provides the required scalability [16, 17, 18, 19].. Essentially, a clustering algorithm determines a set of nodes that can provide a backbone to connect the network to the base station. This set of nodes is called cluster head (CH) set and the rest of nodes are called member nodes (or regular nodes). The clustering algorithm assigns each regular node to be a member of one of the cluster head nodes. Thus the network is partitioned into groups called clusters, and each cluster has one cluster head node that works as a coordinator of this cluster. The main objective of this thesis is to propose techniques that can employed in clustering algorithms to further prolong the network lifetime. This objective is broken down into two areas of concerns.

Firstly, a cluster head node has an extra burden as it must receive messages from its cluster members, aggregate them, transmit the aggregated massage to the next hop towards the BS, and relay the aggregated messages originated by other cluster head nodes. Re-clustering the network is often necessary in order to achieve load balancing;

however, re-clustering consumes energy as well, thus, to maximize the benefit of the clustering algorithm this energy consumption must be minimized.

Environmental monitoring applications have a "*convergecast*" traffic pattern [69] in which traffic is oriented towards the BS. As a result, the nodes that lie in the neighborhood of the BS become a bottleneck for the network [12, 13], and hence these nodes run out of energy faster than the other nodes, consequently isolating the network from the BS. The objective here is to model the extra burden of the nodes that have direct communication with the BS, so as to derive transmission range tuning mechanism that balances energy consumption among clusters. As a result, the network lifetime will be prolonged without compromising the performance of its nodes.

Secondly, there are often situations where the nodes of the network are deployed to be stationary in hostile environments. However, such sensors may be subjected to external forces such as wind erosion, water erosion, etc. this results in undesirable movement of the nodes leading to degraded performance due to the wireless links connecting the cluster head nodes and their members being interrupted. The interruptions invoke the process of re-clustering, hence more energy is consumed. The objective is to propose a metric that can be used to predict the movement of the sensor nodes as such less re-clustering effort is needed. To this end, an information theoretic metric is introduced as a measure of relative mobility of nodes.

1.3 Thesis Organization

This thesis is organized into five chapters. Chapter 2 presents a literature review to provide the background necessary for a general understanding of challenges related to WSNs. In Chapter 3, a Novel algorithm for tuning sensor transmission in clustered WSN is proposed. Chapter 4 describes an information theoretic metric that predicts undesired mobility of sensor nodes. Conclusions and future work are presented in Chapter 5.

Chapter 2

Background and Literature Review

Designing Energy-efficient algorithms is an essential requirement for extending the network lifetime, or the longevity, of a network due to the limited energy supply. The random deployment of sensor nodes in a hash environment, such as being dropped by a helicopter, make the power supply difficult, if not impossible, to replace. Furthermore, most WSN applications require lager number of sensor nodes to cover a vast area and provide reliable information. Energy-aware algorithms able to scale to a large number of sensor nodes are therefore needed. Energy can be wasted due to idle listening, overhearing, retransmitting collided packets, and control packets. Many medium access control (MAC) algorithms have been devolved in order to address these issues and attempt to reduce energy consumption [47, 42, 50, 51]. The aim of many routing protocols for WSNs is to provide the route to the base station that requires low overhead and as few of the control packets as possible [53, 52, 41, 39]. Clustering is an important component that can provide a scalable architecture and prolong the network lifetime. In [54,55],several design issues and techniques for WSNs describing the physical constraints on sensor nodes, applications, architectural attributes, and the protocols proposed in all layers of the network stack

2.1 Clustering

The process of partitioning the network into groups is known as clustering, and it has been proven to prolong the longevity of the network and provide the required scalability [16,17]. The old clustering techniques, such as K-means and G-means [19], are not applicable to WSNs because they iteratively optimize a cost function in a centralized fashion. Rather, a distributed low-complexity clustering algorithm is desired because the goal is to reduce, not increase, energy consumption. Each cluster has a coordinator, often referred as a master node or cluster head. Based on specific criteria, the sensor node may elect a cluster head or the cluster head is chosen through pre-assignment by the network designer. In environment monitoring applications, the cluster head is elected from sensor nodes that have the same capabilities. In a cluster-based network, a routing protocol needs to consider only the set of cluster heads. Building a routing table for a subset of the sensor node that is much smaller than the whole set is cheaper and more energy efficient. Clustering also efficiently utilize the communication bandwidth since it localizes the interactions among the nodes, which improves the performance of the MAC protocols. In addition to facilitating the functionality of the routing and MAC protocols, clustering reduces the topology maintenance overhead because sensor nodes are concerned about connecting only with their cluster heads. The main objective of the clustering algorithms varies according to the application and the network model. This review surveys the class of distributed clustering algorithms because of their simplicity, feasibility, and effectiveness in providing energy efficiency, load balancing, and scalability. Clustering algorithms have been intensively investigated as either stand alone algorithms, e.g., [62],

[15], [63], [14], [64], [17], or in the context of routing, e.g., [60], [65], [66], [16]. Among the early distributed clustering algorithms for WSNs proposed in the literature is the linked cluster algorithm (LCA) [20]. This algorithm favours sensor nodes with higher identifiers (IDs), assuming that every node has a unique identifier and that these identifiers are uniformly assigned throughout the field. The LCA is a distributed algorithm, has a variable convergence time ($O(n)$), and focuses on maximizing network connectivity. Time-based medium access is assumed so that each node is assigned a unique time slot in the frame. This assumption is not feasible for distributed algorithms since a centralized control to provide the unique time slot assignment. At the set-up stage, each node broadcasts its ID in its designated time slot and listens to the transmissions of other nodes. Each node has a list of its neighbors and, the node with the highest ID is elected as a cluster head. The LCA induces a large number of cluster heads in order to insure a maximum inter-cluster connectivity and thus a large routing overlay. In [21], an extension is proposed to reduce the number of CHs. Neither the LCA nor the LCA2 may be suitable for WSNs because they favor some energy-constrained sensor nodes regardless of the residual energy.

In [14], a Max-Min clustering algorithm is proposed as an extension of the LCA that will produce fewer cluster heads and, hence, a smaller routing overlay. The Max-Min algorithm has two broadcast stages. First, each node broadcasts its CH, set initially as the node’s ID, to its entire one-hop neighbors, then chooses the highest ID to be the new CH, and continues for d rounds. This process is denoted as floodmax. Second, the same process is repeated for another d rounds, but this time the lowest ID is favoured. This distributed algorithm has a very high complexity compared with constant convergence time algorithms. The Max-Min algorithm assumes a routing infrastructure in place, and again, cluster head selection based on the node ID does not ensure that the nodes with high energy can become cluster heads. In other words, the nodes with the lowest IDs might be the ones with the highest residual energy, and the role of cluster head is thus restricted to the ones with high IDs.

Most clustering algorithms periodically trigger re-clustering in order to balance the load because of the over use of the cluster head nodes compared with the regular nodes. This factor is also the focus of a number of studies in [1][2][3][4]. Amis and Prakash proposed a load-balancing algorithm to be built on top of the LCA and LCA2 since these two algorithms do not consider load balancing [34].

Low energy adaptive clustering hierarchy (LEACH) is one of the most poplar clustering algorithms for WSNs [16]. LEACH predetermines the optimal number of cluster head k given the number of sensor nodes N and the field's dimensions $M \times M$ using the computation and communication energy models provided.

$$k_{opt} = \frac{M}{d_{CH}^{2}} \sqrt{\frac{N\varepsilon_{fs}}{2\pi\varepsilon_{mp}}}$$

LEACH is divided into rounds, each of which has a clustering phase and a steady-state phase. Initially in the clustering phase, each node elects itself to be a cluster head with a probability P_i, so that the expected number of cluster heads given P_i is the optimal number of cluster heads *k*. The role of being a cluster head is rotated among the sensor nodes, thus providing load balancing among the nodes. LEACH performs a randomized rotation by having each node choose a random number between 0 and 1. A node becomes a cluster head in the current round if the number is less than P_i, which is assigned a value of zero if the node has already been a cluster head. The nodes are assumed to have a long transmission range so that they are all within one another's range and have a direct link with the base station. This assumption is not feasible for large networks since the regular nodes are usually deployed in large numbers over a wide area. The probabilistic criteria for selecting cluster heads also not does guarantee that the nodes with low residual energy will not become cluster heads. An attempt is made to ensure that the node has a probability P_i that is a function of its energy. However, each node must then have an estimate of the total energy of all the nodes, resulting in a very high overhead. In general, LEACH motivated a number of new algorithms [17, 28], has a constant convergence time, and is distinguished by the improvement of the network lifetime.

The weighted clustering algorithm (WCA) is another important clustering algorithm that has motivated the development of many contemporary clustering algorithms [15]. This algorithm was originally proposed for mobile ad hoc networks, since mobility is considered as one of the criteria for selecting cluster heads. Most of the distributed algorithms favor one metric for cluster head selection criteria and use another metric as a fitness function for breaking ties. A regular node must choose which cluster head to subscribe to when it hears an advertisement message (ADV) from more than one cluster head at the same time. This process is known as breaking ties. The WCA favors a combined weight metric that takes into consideration four metrics, with each metric assigned a weight indicating the importance of that particular metric compared with the others so that each node *v* has a weight, as follows:

$$W_v = w_1\Delta_v + w_2 D_v + w_3 M_v + w_4 P_v$$

The first metric is the difference between a node degree's and a pre-defined threshold. This threshold is the ideal number of nodes in a cluster (cluster size). In general, a node that has the smallest sum of distances (D_v), the lowest mobility, and the lowest power consumption among it neighbor is elected as a cluster head. In the WCA, each node broadcasts its ID along with its weight and stores the neighbors' weights it receives. This process continues until all the nodes have information about the entire network. A global solution is claimed to be found with a very high overhead in spite of the fact that a local minima with low overhead would be sufficient. Another approach is considered in which all the information is processed at the base station. However, such a centralized approach is not feasible for ad hoc networks in general, including WSNs.

In [31], the centralized WCA is optimized by simulated annealing to provide the optimal set of CHs. Yu-Xuan Wang proposed an entropy-based weight clustering algorithm (EWCA) as a variation of WCA [30]. Another variation of the centralized approach of WCA is proposed in [29]. Using a method similar to that of WCA, a combined metric is used as the cluster head selection criterion. The information is processed and optimized using a genetic algorithm to provide the optimal set of CHs at the base station.

In [32], a different method of entropy-based clustering is proposed. The different node parameters are characterized in terms of entropy, in particular, mutual information. Every node collects a history of the broadcast message received from its neighbor during a specific period of time. These statistics allow an approximation of the distribution, and the mutual information between a nodes and its CH will determines the stability of the cluster. Cluster heads are selected based on the combined metric of the energy consumption uncertainty and the mutual information as follows:

$$H_{total} = w_1 H_{mobility} + w_2 H_{energy}$$

The exact algorithm operation is not addressed in the paper, indicating that the same WCA process is adopted, since they refer to WCA. The idea of using an information theoretic metric is very effective if the mobility distribution is provided in advance rather than being built as the data is being collected. Estimating the mobility distribution is a very expensive process because of the need to collect the statistical information, store it, and then exchange the collected information.

Hybrid energy-efficient distributed (HEED) clustering algorithm is considered to be a state-of-art distributed clustering algorithm for WSNs [17]. HEED favors nodes with high residual energy to become cluster heads and periodically executes re-clustering to achieve load balancing. The nodes that have been cluster heads will have a low probability of becoming cluster heads again, thus ensuring that all the nodes will carry the role of being a cluster head equally. HEED uses node degree as a fitness function if the requirement is to distribute the load among the cluster heads, the inverse of the node degree if the requirement is to create a dense cluster, or the mean of the minimum power levels required by all the nodes within the node's transmission range to reflect the communication cost within a cluster. In the clustering phase, each node sets its probability of becoming a cluster head as follows:

$$P_i = C \times \frac{E_{residual}}{E_{\max}}$$

$E_{\max}$ corresponds to a fully charged battery, $E_{residual}$ is the current residual energy, and C is to limit the initial announcement messages because it should be a small percentage. Each node iteratively doubles its probability until the probability reaches one. It announces itself as a cluster head, and the nodes that hear the ADV message will withdraw from the election process and join the advertised cluster head. If it hears from more that one cluster, a regular node breaks ties according to one of the above fitness function. The announcement messages are delayed based on the node's residual energy, meaning that the nodes with high residual energy will advertise themselves before the low-energy ones do. HEED assumes that a node has two levels of transmission range: low-level transmission range for intra-cluster communication and a high-level transmission range for inter-cluster communication that should be at least double the low-level one to ensure inter-cluster communication since HEED does not adapt the use of gateways to provide the desired connectivity. HEED is completely distributed, terminates within a fixed number of iterations, produces well-distributed clusters over the field in terms of cluster size, scales for very large networks, and significantly increases the network lifetime.

As a variation of HEED, time delay based clustering (TDC) is introduced in [28]. In TDC, all nodes compete to be CH until they hears ADV message, and then withdraw from the election process and subscribe to one of the elected cluster heads. As with HEED the announcement message is delayed, but with a slightly different delay mechanism. In HEED, the nodes are delayed by a simple iterations mechanism as discussed above. In TDC, three time delay schemes are studied: fully randomized, fixed slope, and steeping slope.

Qin and Zimmermann propose a voting-based clustering algorithm (VCA) in [33]. In the VCA, each node exchanges residual energy with its neighbors and casts a vote for itself and its neighbors as follows:

$$v(v_i, v_j) = \begin{cases} \dfrac{e_j}{\sum\limits_{d_{ik} \le R} e_k} & , d_{ij} \le R \\ 0 & , d_{ij} \ge R \end{cases}$$

e_i denotes the residual energy of sensor node v_i so that the node with the highest residual energy will become a cluster head. VCA resembles HEED in many ways: nodes with high residual energy receive the highest number of votes and become CHs in VCA in the same way that in HEED high-residual-energy nodes are the fastest to send the announcement messages and then become CHs.

Most of the above algorithms adopt a simple radio and energy consumption model that was originally developed in [16]. An analysis of the energy consumption and lifetimes of WSNs is discussed in [26]. An attempt to mathematically model the energy consumption of a d-hop cluster-based network given the number of hops (d) and the transmission range (r) is introduced in [27].

Abbasi and Younis survey different clustering algorithms specifically for WSNs, highlighting their objectives, features, etc [22]. In [23], developments and deployment challenges are discussed, and some of the open issues in this area are addressed. Wei and Chan survey clustering schemes for ad hoc networks in general and classify them into ad hoc sensor network clustering schemes and mobile ad hoc clustering schemes [24].In [25], clustering schemes for mobile ad hoc networks are surveyed providing description of the mechanisms, evaluation of their performance and cost, and discussion of the advantages and disadvantages of each clustering schemes.

2.2 Routing

Routing protocols that were originally developed for wireless networks such as DSDV [35], DSR [37], AODV [36], TORA [38], etc in both forms, proactively and reactive, are not applicable for WSNs. These IP-based protocols require a high overhead and a global addressing scheme. Providing unique ID for a large number of sensor nodes and the high maintenance required is not feasible for WSNs. Furthermore, for the base station the data is more important than identifying the source. IP-based protocols are also not suitable for WSNs due to resource limitation (e.g., energy, memory). Flooding-based routing protocol do not require global addressing scheme, thus avoiding the above difficulty. However, conventional flooding techniques consume a large amount of energy due to implosion and packet overlap. Although, gossiping routing-based protocols overcome the problem of implosion, they still have high energy consumption and large delay [56].

Directed diffusion (DD) is one of the most popular routing protocols for WSNs [39, 40]. Directed diffusion consists of several elements: interests, data messages, gradients, and reinforcement. The base station floods the network with a query about events they are interested in. The query is called the interest in the sensor networks. Each interest contains a description of the sensing task. The data messages are the events generated by a single node or a group of nodes in response to the query sent by the base station. In directed diffusion, a query is named using attribute value pairs. The interest queries are propagated throughout the network setting up the gradients within the network in order to draw events back to the BS. A gradient is direction state created in each node that receives an interest. The source node sends the events back to the base station along multiple gradient paths with a low data rate. The base station reinforces one particular path in order to draw down data with high transmission rate. Aggregation techniques are applied along the path to reduce the amount of data due to data redundancy. Hence, the communication cost is reduced, and the longevity of the network is increased. However, directed diffusion may not be applied to applications that require continuous data delivery to the base station.

Braginsky and Estrin proposed a variation of directed diffusion in [41], called rumour routing. Rumour routing is a probabilistic protocol for matching queries with data events. Flooding only the nodes that have observed interested events reduces the cost of the initial flooding process. Rumour routing utilizes long-lived packets, known as agents, that are stored in a local table. Every source node floods the entire network with agents advertising its sensed data so that the injected query will be directed to the source nodes. Rumour routing outperforms directed diffusion only when the number of events is small.

Minimum Cost Forwarding Algorithm (MCFA) proposed in [52] is intended for applications that have stationary sensor nodes and a fixed base station. Each nodes in the network maintains a cost field. The cost field specifies the minimum cost required to reach the base station. To forward a packet to the base station, the nodes check the cost field associated with a neighbour and then choose the rout with the minimum cost. The cost field can store any metric, such as hop-count, energy, latency or loss. The MCFA is a simple protocol with low overhead and has no need to maintain the path information.

However, the same link is used every time, causing these nodes to fail much faster than in the other networks.

The authors in [53] proposed a two-tier data dissemination (TTDD) protocol intended for applications that have multiple mobile base stations and location-aware stationary nodes. A data source node builds a grid structure to be used in disseminating data to the mobile BSs. Each data source node chooses itself as the starting crossing point of the grid, and sends an announcement message to each of its four adjacent crossing points using simple greedy geographical forwarding to build the grid structure. One node that is close to the crossing point is chosen as a dissemination point, and the process continues until the massage stops at the border of the network. When the grid structure is in place, a BS can flood to the dissemination points. TTDD has a high overhead associated with maintaining and recalculating the grid.

Combined clustering and route setup has also been considered for maximizing networks lifetime [57, 60, 61]. Two cluster-based routing protocols called threshold-sensitive energy efficient protocol (TEEN), and adaptive periodic threshold-sensitive energy efficient protocol (APTEEN) are proposed in [58] and [59], respectively. Every member node receives a hard threshold, which is the threshold value of the sensed attribute and a soft threshold, which is a small change of the attribute value form its cluster head. These thresholds enable the network administrator to reduce the amount of the transmitted data as a trade-off between energy efficiency and data accuracy. APTEEN includes count time, which is the maximum period between two successive reports sent by a node. The count time is included to overcome the drawback of TEEN, which is, if the thresholds are not received, the node will never communicate.

2.3 Medium Access Control

Mac protocols are used to create predefined ways for multiple users to share a channel. WSNs require designing an energy-efficient MAC protocol that reduces wasteful energy consumption, such as idle listening, overhearing, and retransmitting collided packets. However, the evolution of MAC protocols and the fact that every new protocol is an extension of an exiting protocol requires an understanding us of the historical development of MAC protocols. There are two fundamentally different ways to share the channel bandwidth among different nodes: fixed-assignment channel-access methods, such as time-division multiple access (TDMA), frequency-division multiple access (FDMA), and space-division multiple access (SDMA), and random access methods, such as IEEE 802.11, carrier sense multiple access (CSMA), multiple access collision avoidance (MACA), and MACA for wireless (MACAW).

Fixed-assignment MAC protocols allocate each user a given amount of bandwidth, slicing the spectrum according to time (TDMA), frequency (FDMA), code (CDMA), or space (SDMA). Since each node is allocated a unique part of the spectrum, there are no collisions among the data. However, fixed assignment schemes are inefficient when not all nodes have data to send, since scarce resources are allocated to nodes that are not using them. Random-access methods, on the hand, do not assign users fixed resources.

These are contention-based schemes, in which nodes that have information to transmit try to obtain bandwidth while minimizing collisions with other nodes' transmissions. These protocols are more efficient than fixed-assignment MAC protocols when node have bursty data. However, they suffer from possible collisions of the data, as all nodes are contending for the resources. Often protocols use a hybrid approach, e.g., combining TDMA and FDMA by allocating a certain time and frequency slot for each node. MAC protocols can be evaluated in terms of energy dissipation, fairness, and throughput: the protocol is typically optimized to minimize energy dissipation, give each node its fair share of the bandwidth, and achieve high throughput.

CSMA is an important contention protocol [47]. Its central idea is listening before transmitting. The purpose of listening is to detect whether the medium is busy, also known as a carrier sense. There are several variant of CSMA, including non-persistent CSMA, 1-persistent CSMA, and p-persistent CSMA. In non-persistent CSMA, if a node detects an idle medium, it transmits immediately. If the medium is busy, it waits a random amount of time and starts the carrier sense again. In 1-persistent CSMA, a node transmits if the medium is idle. Otherwise, it continues to listen until the medium becomes idle, and then transmits immediately. In p-persistent CSMA, a node transmits with a probability p if the medium is idle and with a probability (p-1) to back off and restart the carrier sense. Woo and Culler examined the performance of CSMA with various configurations when it is used in wireless sensor networks [44]. They proposed a MAC protocol for sensor networks, which combined CSMA with an adaptive rate control mechanism. This protocol is based on a specific network setup in which there is a base station that tries to collect data equally from all sensors in the field.

In a multi-hop wireless network, however, the CSMA alone is not sufficient due to the hidden terminal problem. The CSMA with collision avoidance (CSMA/CA) was developed to address the hidden terminal problem and has been adopted by the wireless LAN standard IEEE 802.11 [45]. The basic mechanism in a CSMA/CA is to establish a brief handshake between a sender and a receiver before the sender transmits data. The handshake starts from the sender by sending a short request-to-send (RTS) packet to the intended receiver. The receiver then replies with a clear-to-send (CTS) packet. The sender starts sending data after it receives the CTS packet. The purpose of RTS-CTS handshake is to make an announcement to the neighbors of both the sender and the receiver.

Based on CSMA/CA, Karn [48] proposed MACA, which added a duration field in both RTS and CTS packets to indicate the amount of data to be transmitted so that other nodes know how long they should back off. Bharghavan further improved the performance of MACA in their protocol MACA for wireless network (MACAW) [49]. MACAW proposed several additions to MACA, including use of an acknowledgement (ACK) packet after each data packet, thus allowing rapid link-layer recovery from transmission errors. The transmission between a sender and a receiver follows the sequence of RTS-CTS-DATA-ACK.

IEEE 802.11 adopted all the features of CSMA/CA, MACA, and MACAW in its distributed coordination function (DCF) and made various enhancements, such as virtual carrier sense, binary exponential back off, and fragmentation support. DCF is designed for ad hoc networks, while the point coordination function (PCF), also known as infrastructure mode, adds support when designated access points (or base stations) mange wireless communication.

Both Piconet and the 802.11 PS modes try to save energy by reducing the idle-listening time. They do not address the overhearing problem. PAMAS, proposed by Singh and Raghavendra [47], avoids the overhearing problem by putting nodes into a sleep state when their neighbors are transmitting. PAMAS uses two channels, one for data and one for control. All control packets are transmitted in the control channel. After a node wakes up from sleep, it probes the control channel to find any possible ongoing transmissions and their duration. Probing the control channel avoids interfering with a neighbour’s transmission in the data channel, and the neighbour is able to answer the probe in the control channel without interrupting its data transmission. However, the requirement of separate control channels and data channels makes PAMAS more difficult to deploy, since multiple channels require multiple radios or additional complex channel allocation. PAMAS also does not reduce idle listening.

S-MAC is a low power RTS-CTS protocol for wireless sensor networks inspired by PAMAS and 802.11[42]. S-MAC periodically sleeps, wakes up, listens to the channel, and then returns to sleep. Each active period is of a fixed size, 115 ms, with a variable sleep period. The length of the sleep period dictates the duty cycle of S-MAC. At the beginning of each active period, nodes exchange synchronization information (SYNC). Following the SYNC period, data may be transferred for the reminder of the active period using RTS-CTS. In a follow-up paper [43], the authors add adaptive listening: when a node overhears a neighbor's RTS-CTS packets, it wakes up for a short period at the end of the neighbour's transmission to immediately transmit its own data. By changing the duty cycle, S-MAC trades off energy for latency. S-MAC includes a fragmentation mechanism that uses RTS-CTS to reserve the channel and then transmits packets in a burst. Although S-MAC achieves low-power operation, it does not meet the goals of simple implementation, scalability, and tolerance to changing networks conditions. As the size of the network increases, S-MAC must maintain an increasing number of neighbors' schedules or incur additional overhead through repeated rounds of resynchronization.

T-MAC improves on S-MAC's energy usage by using a very short listening window at the beginning of each active period [50]. After the SYNC section of the active period, there is a short window for sending or receiving RTS and CTS packets. If no activity occurs in that period, the nodes return to sleep. By changing the protocol to have an adaptive duty cycle, T-MAC saves power at the cost of reduced throughput and additional latency. In variable workloads, T-MAC, uses one-fifth the power of S-MAC. In homogeneous workloads, T-MAC and S-MAC perform equally well. T-MAC suffers from the same complexity and scaling problems encountered in S-MAC. Shortening the active window in T-MAC reduces the ability to snoop on surrounding traffic and to adapt to changing network conditions.

One example of a good MAC protocol for wireless sensor networks is B-MAC [51]. B-MAC is highly configurable and can be implemented with a small code and memory size. It has an interface that allows it to choose only the functionality needed for a particular application. B-MAC consists of four main parts: clear channel assessment (CCA), packet backoff, link layer acks, and low-power listening. For CCA, B-MAC uses a weighted moving average of samples when the channel is idle in order to assess the background noise and to better detect valid packets and collisions. The packet backoff time is configurable and is chosen from a linear range as opposed to the exponential backoff scheme typically used in other distributed systems. This feature reduces delay and works well because of the typical communication patterns found in a wireless sensor network. B-MAC also supports a packet-by-packet link layer acknowledgement: only important packets are allocated resources. A low-power listening scheme is employed when a node cycles between awake and sleep cycles. While awake, it listens to the preamble long enough to assess whether it needs to stay awake or can return to sleep mode. This scheme saves significant energy. Many MAC protocols use a request to send (RTS) and clear to send (CTS) style of interaction. This works well for ad hoc mesh networks in which packet sizes are large (thousands of bytes). However, the overhead of RTS-CTS packets to set up a packet transmission is not acceptable in wireless sensor networks in which packet sizes are in the order of 50 bytes. B-MAC, therefore, does not use an RTS-CTS scheme.

2.4 Summary

This chapter presented and discussed clustering algorithms developed for WSNs. It also briefly discussed routing protocols and MAC protocols. A review of Research work that aimed at reducing energy consumption so as to increase network lifetime was provided. Clustering was shown to prolong the network lifetime, providing the required scalability, and functionality of both routing and MAC protocols.

It has been shown that despite the diversity and richness of research in this area, existing work does not fully address the unique characteristic of WSNs, namely, overuse of the nodes that are within the neighborhood of the BS and undesired mobility due to external factors.

Chapter 3

Transmission Range Tuning of Clustered Wireless Sensor Networks with Topological Consideration

It is well acknowledged that the hierarchy provided by a clustering algorithm is an efficient way to save energy and prolong the network lifetime of a wireless sensor network (WSN). However, none of the existing clustering approaches considers the location of the sensor nodes relative to the base station (BS). It is obvious that the nodes that are close to the base station become a bottleneck in the network, as shown in Figure 3.1.

Most WSN applications have a convergecast communication pattern [69], in which sensor nodes detect or sense events and send messages to an information center (BS) by hopping the message until it reaches the destination. Consequently, the nodes within the transmission range of the base station relay messages originated by the nodes that have no direct communication with the information centre. Since these nodes are burdened with extra overhead, they deplete their energy and fail or die much faster than other nodes. Furthermore, the network loses its connectivity and the base station becomes isolated from the network.

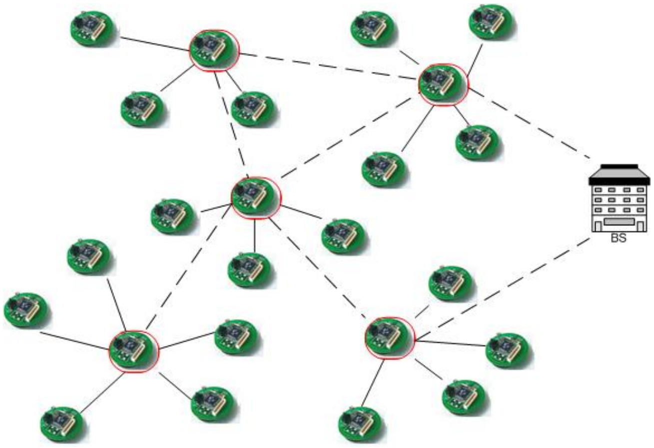


Figure 3.1: Cluster-based network

In general, one of the important requirements of WSNs is to have distributed algorithms and protocols and to be self-configuring. If the network is self-configuring, it continues to function, and the nodes continue transmitting sensed messages to neighbouring nodes within the range without knowing that the network has lost its connectivity with the base station. Therefore, the energy remaining in the alive or functioning nodes is wasted.

In a cluster-based network, a source communicates with a far-off destination (i.e., a BS or a common sink) by sending the message to its cluster head (intra-cluster communication), and the cluster head can then communicate with the base station through an overlay network of cluster heads that form a virtual backbone (inter-cluster communication). As with the individual nodes, cluster heads that have direct communication with the base station run out of energy faster than the other cluster heads, and the network becomes isolated from the information centre.

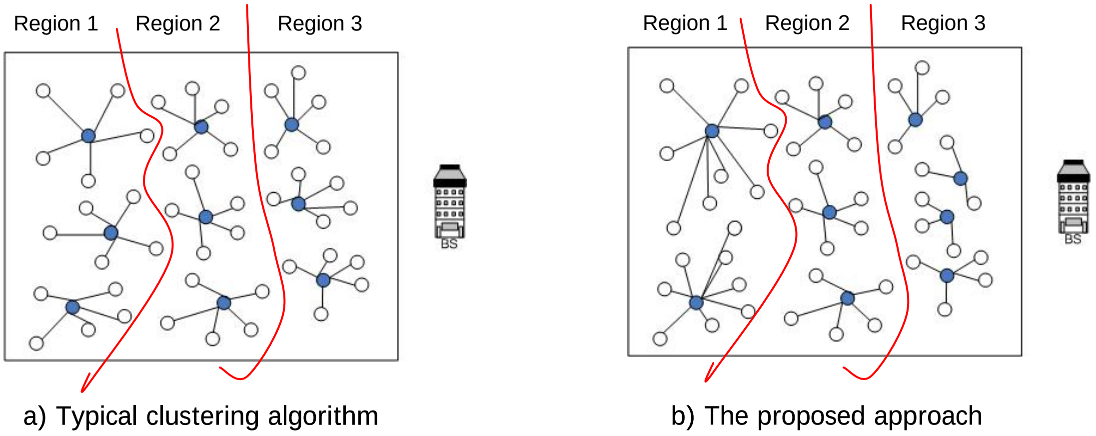
Compared to the other techniques; clustering has been proven to prolong network lifetime [17, 28, 15, 16]. The goal of most clustering algorithms is to cluster the network in a uniform distributed fashion, meaning that the number of nodes within each cluster is the same in order to achieve load balancing, and then rotate to the role of cluster head among the nodes by repeating the clustering process. However, load balancing can not be achieved without a consideration of the extra burden of some of the nodes. Therefore, the clusters should be uniform in terms of energy consumption rather than the number of nodes. Only if all nodes have the same level of energy consumption, will being uniform in terms of the number of nodes provide load balancing.

In the next section, the philosophy of load balancing is introduced. In section 3.3, the extra burden that the nodes that lie within the neighbourhood of the base station is formulated as extra virtual member nodes. In section 3.4, a pre-clustering algorithm is given based on the formulation. Experimental work is shown in section 3.5, and a summary and conclusions are presented in section 3.6.

3.1 Topological Considerations

Load balancing is an important factor in achieving better performance and improving the longevity of the network (network lifetime). A clustering algorithm has two levels of load balancing. First, the energy consumption among the nodes and their cluster head must be equalized. Thus, all the nodes must carry the load of the cluster head equally. Second, the energy dissipation among cluster heads must be equalized as well. Therefore, the cluster heads that are burdened with extra duties must be assigned fewer member nodes, which means that these cluster heads must have a smaller transmission range.

A method of classifying the nodes is needed so that they can have different transmission ranges. Thus, the network should be divided into *n* regions. The number of regions (*n*) is the ratio between the transmission ranges of the sensor nodes and the length of the field. Accordingly, the sensor nodes are categorized into *n* classes. For instance, the nodes that fall into region 1 are called region 1 nodes. Figure 3.2 a. illustrates a cluster-based network that is divided into three regions (*n*=3). Region 1 is denoted as the rear region and region *n* as the front region. If the extra burden can be estimated as extra virtual member nodes, then the transmission range can be tuned to reduce the number of real member nodes. This indicates that the transmission range of the front region nodes must be smaller than the transmission range of the rear region nodes in order to achieve load balancing. As a result, the number of clusters in the front region will be larger than in the rear region as shown in Figure 3.2 b.



a) Typical clustering algorithm

b) The proposed approach

Figure 3.2: Cluster-based networks based on topological consideration

Furthermore, the optimal number of clusters is inversely proportional to the distance to the base station, as equation 1 implies []. The nodes that communicate directly with the base station prefer to send their own messages directly to the BS rather than sending them to a cluster head and then having the cluster head sends them to the base station. In other words, the cost to communicate directly with the base station is almost the same as the cost to communicate with the cluster head, so why should the cluster head be burdened when it is more efficient to send the data directly to the base station? Again this arrangement explicitly states that the number of clusters in region i should be larger than the number of cluster in region $i-1$.

$$k_{opt} = \frac{M}{d_{CH}^{2}} \sqrt{\frac{N\varepsilon_{fs}}{2\pi\varepsilon_{mp}}} \tag{1}$$

$$k_{opt} \propto \frac{1}{d_{BS}} \tag{2}$$

The network lifetime has been defined in various ways. The most common are the time until the first node dies and the time until the last node dies. However, the latter definition is useless when the network loses its connectivity with the base station. Therefore, it should be redefined as the time until the failure of the last node that has a direct link with the information centre.

3.2 Quantitative Analysis with Topological Considerations

This section introduces formulation of the extra burden as virtual nodes**.** The radio communication and energy consumption described in [16] is adopted: for short distance transmission, such as intra-cluster communication, the energy consumed by a transmiting amplifier is proportional to r^2 and for long distance transmission, such as inter-cluster communication, the energy consumption is proportional to r^4. Using the given radio and energy consumption models, the energy consumed in transmitting one message among cluster heads for a distance *d* is given by

$$E_{TX} = lE_t + l\varepsilon_{fs} d_{node}^4 \qquad (3)$$

Similarly, the energy consumed when the senor node works as a regular (member) node, that is, the energy consumed in transmitting a massage within a cluster for a short distance d, is given by

$$E_{node} = lE_t + l\varepsilon_{fs} d_{node}^2 \qquad (4)$$

$$E_{node} = lE_t + l\varepsilon_{fs} d_{node}^2 \qquad (5)$$

The energy consumed in receiving a message is given by

$$E_{RX} = lE_t \qquad (6)$$

The duties of a cluster head are to receive messages from *n* members; aggregate the messages, including its own sensed message; transmit the aggregated message to the next hop (neighboring cluster head); and relay messages coming from neighboring cluster heads. Thus, the energy consumed by a cluster head is given by

$$E_{CH} = nlE_r + lE_{aggregate}(n+1) + lE_t + l\varepsilon_{mp} d_{CH}^4 + E_{relay} \tag{7}$$

The energy consumed in relaying a message, when a cluster head receives the message originating from another cluster head and transmits it to the next cluster head closer to the base station, is given by

$$E_{realy} = lE_r + lE_t + l\varepsilon_{mp} d_{CH}^4 \tag{8}$$

However, the energy consumption varies from one cluster head to another depending on which reign it falls into. Let E_{relay_i} denote the energy dissipated in relaying packets by the nodes in region i. The energy dissipated in relaying messages by the nodes of region i is equivalent to the energy consumption of relaying packets by the nodes of regions $i-1, i-2, ..., 1$. This energy must be considered when the energy consumption of cluster head nodes is defined. Since the nodes of region 1 do not realy any messages, E_{relay_1} is equal to zero. Similarly, let E_{CH_i} denote the energy consumption of a region i cluster head:

$$E_{CH_i} = E_{CH} + E_{relay_i} \tag{9}$$

and

$$E_{relay_i} = \sum_{j}^{i-1} n_j E_{relay} \tag{10}$$

where n_j is the number of clusters in region j. Assuming a uniform distribution of relayed packets in region i among the cluster head of this region; Equation 9 becomes

$$E_{CH_I} = E_{CH} + \frac{\sum_{j}^{i-1} n_j}{n_i} E_{relay} \tag{11}$$

In order to achieve connectivity of in cluster-based network, the transmission range of the cluster heads (d_{CH}) must be at least double the transmission range of the member nodes (d_{node}) [heed,it ref]. Therefore, the ratio between E_{CH} and E_{node} can be approximated as follows:

$$\frac{E_{CH}}{E_{node}} = 2m \tag{12}$$

where m is the number of member nodes assigned to a cluster head. Similarly, the ratio between E_{relay} and E_{node} can be approximated to

$$\frac{E_{relay}}{E_{node}} = 2.1 \tag{13}$$

Substituting Equations 12 and 13 in Equation11, results in

$$E_{CH_I} = E_{CH} + 2.1\frac{\sum_{j}^{i-1} n_j}{n_i} E_{node} \tag{14}$$

Equation 14 shows that the clusters in region i have a burden of an extra $2.1\frac{\sum_{j}^{i-1} n_j}{n_i}$ nodes, as if these clusters have extra member nodes. It should be noted that the energy consumption rate of a cluster head depends on the number of its member nodes.

3.3 Transmission Range Tuning of Cluster-Based Networks

The transmission range controls the number of clusters: large transmission ranges produce a small number of clusters and vice versa. If a clustering algorithm achieves the maximum lifetime of a network with a specific number of clusters (*k*) using *r* as a transmission range, the lifetime can be further improved by using a smaller transmission range in the front region and a lager transmission range in the rear region to produce the same number of clusters ($r_i \prec r_{i-1} \prec r_{i-2} \prec ...... \prec r_1$). These clusters are uniform in terms of energy consumption but not in terms of the size ($k_1 \prec k_2 \prec ...... \prec k_{i-1} \prec k_i$). Figure 3.2 b depicts a cluster-based network divided into three regions and shows that the nodes of each region have different transmission radiuses. This network has the same number of clusters as the network shown in Figure 3.2 a.

The number of member nodes depends on the number of neighboring nodes. For instance, if the number of neighboring nodes is decreased by *x*, then most likely the number of member nodes is decreased by *x* as well. The number of neighboring nodes can be estimated by the following equation:

$$\#neighbors = desity \times r^2 \tag{15}$$

Since the nodes are assumed to be uniformly distributed, the density of the nodes is given as follows:

$$density = \frac{N}{A} \tag{16}$$

where N is the number of nodes and A is the area of the network field. The transmission range of region i can be obtained from the following:.

$$r_i^2 = \frac{density}{\#nieghbors - 2.1\frac{\sum_{j=1}^{i-1} n_j}{n}} \tag{17}$$

Since it is not assumed that the nodes have location awareness, the network is divided into two regions (front and rear). The front region nodes are the ones that can communicate directly with the base station. The rest of the nodes are classified as the rear region nodes. Initially, each node has a transmission radius identifier (r-idf) assigned to be zero. The base station broadcasts a message at the initial setup; the nodes that receive the broadcast message set their transmission range identifier to one.

If the number of clusters is provided by the network designer, i.e., centralized approach [15,16], then the new radius can be calculated using equation 17. Otherwise, initially clusters are formed with the original transmission range and each cluster head sends a small packet. The base station counts the number of these packets, and the number of clusters in the rear region can then be estimated. In either scenario, the new transmission radiuses can be obtained using.

Since different transmission ranges are allowed for, two nodes *A* and *B* are considered neighbors when *A* falls within the range of *B* and *B* falls within the range of *A*. Figure 3.3 shows two nodes A and B in a situation in which they are not neighbors.

.

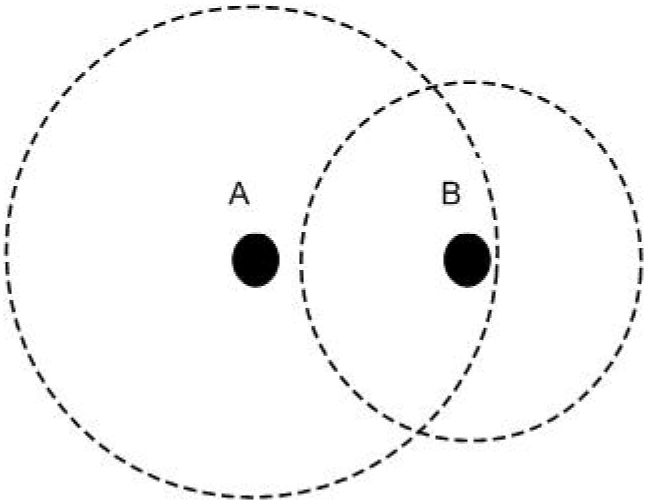


Figure 3.3: Nodes with different transmission ranges

Therefore, regular nodes are assigned to be a member of a cluster head only when they are neighbors based on the above condition. For instance, if a node hears an advertisement message from a cluster head, it withdraws from the cluster head election and considers itself a member of a cluster head that is out of its range. Figure 3.4 shows the pseudo code of the new algorithm as a pre-clustering algorithm.

I. Pre-clustering

```
BS broadcast a message
if a node hears the massage
    r-idf ← one
    $r_i^2 = \frac{density}{\# nieghbors - 2.1\frac{\sum_{j=1}^{i-1} n_j}{n}}$
else
    r-idf ← zero
end
Snbr ← {v:v lies within my transmission range and vice versa}
```

Figure 3.4: Pre-Clustering Algorithm

3.4 Experimental Results

The HEED algorithm is considered a state-of-art distributed algorithms [17]. Therefore, its performance is evaluated in the next section, and the impact of the proposed algorithm on HEED is experimentally investigated.

3.4.1 Network Setup

For theses experiments, a network of N sensor nodes in a $100 \times 100 \quad m^2$ area is considered. The N nodes are assumed to be uniformly distributed over the area. Each node collects the data periodically and sends them to its cluster head until it runs out of battery. The cluster head conveys the aggregated message to the base station through a multi-hop of cluster heads. The base station is assumed to be located outside the network, and a node that depletes its energy resource is considered failed or dead. The shortest path algorithm is used as a routing algorithm for the inter-cluster network. All parameters are given in Table 3.1. A simple radio model that also can be found in [16] has been adopted.

Parameter	**Value**
Network size	100×100
Number of nodes	200
BS	At (50, -20)
E_{ele}	50nJ/bit
ε_{fs}	10 pJ/bit/m2
ε_{amp}	0.0013 pJ/bit/m4
Data packet size	500
Broadcast packet size	50
Packet header size	25
Initial energy	1J

Table 3.1: Simulation parameters

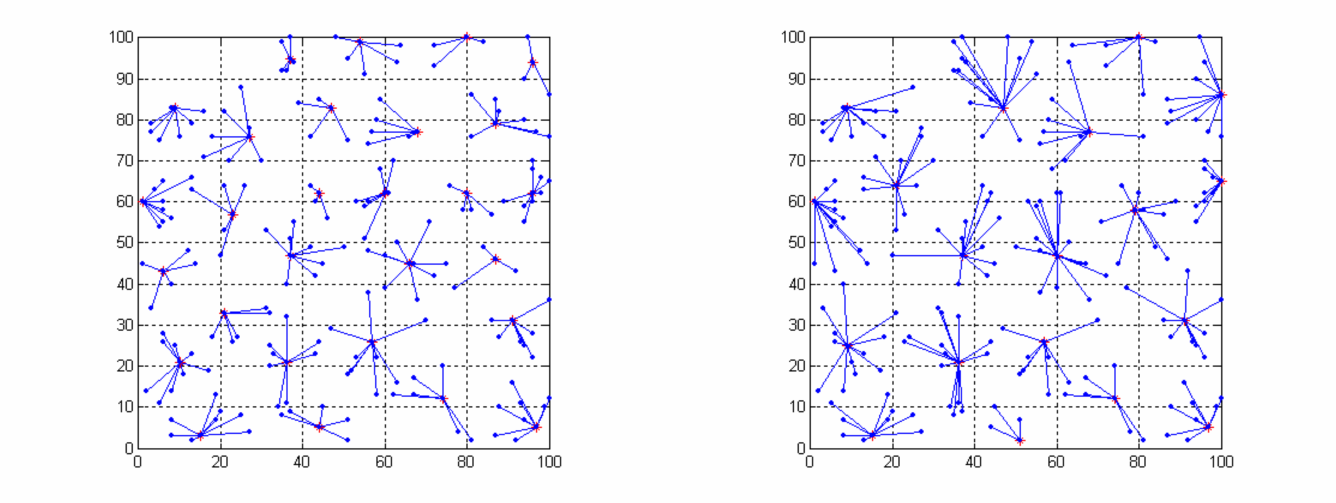
3.4.2 Performance Analysis of HEED

Based on the above setup the HEED algorithm was implemented and tested using Matlab. Figure 3.5 shows the clustering results of the HEED algorithm using different transmission ranges. It is obvious that the number of clusters induced depends on the transmission range. After the clusters are formed, the transmission radius of each regular node is tuned to reach its cluster head (most likely a smaller radius than the original), and the new cluster head's transmission range is doubled. Figure 3.6 shows inter-cluster networks at various transmission ranges. As can be seen from Figure 3.6 a, the virtual backbone of cluster heads has a transmission radius of 30 meters (15 meter doubled).

Figure 3.7 depicts the residual energy of individual nodes sorted according to the proximity to the base station in an ascending order when the network loses its connectivity. The nodes of the front region are dead, and the nodes of the rear region still have some energy that will be wasted since they will continue to operate until they die with no messages received by the BS. This is common for distributed algorithms. Table 3.2 shows the average remaining energy of the alive node at different transmission radiuses.

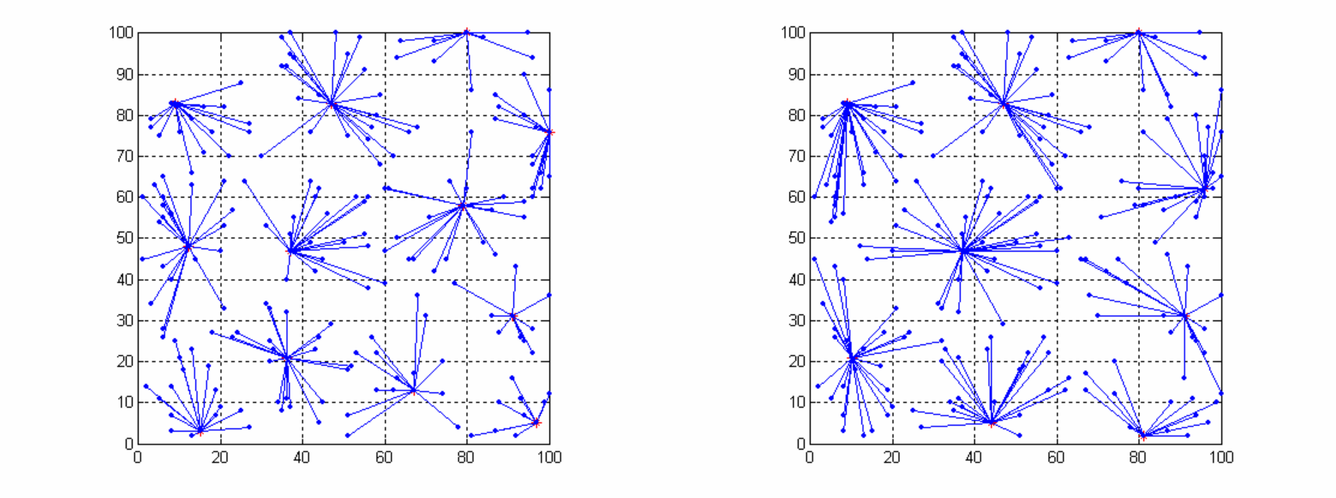
Transmission Range	**The Average Remaining Energy**
15	94%
20	80%
25	75%
30	85%
35	98%

Table 3.2: The average remaining energy



a) Transmission radius=15 m

b) Transmission radius=20 m



c) Transmission radius=25

d) Transmission radius=30

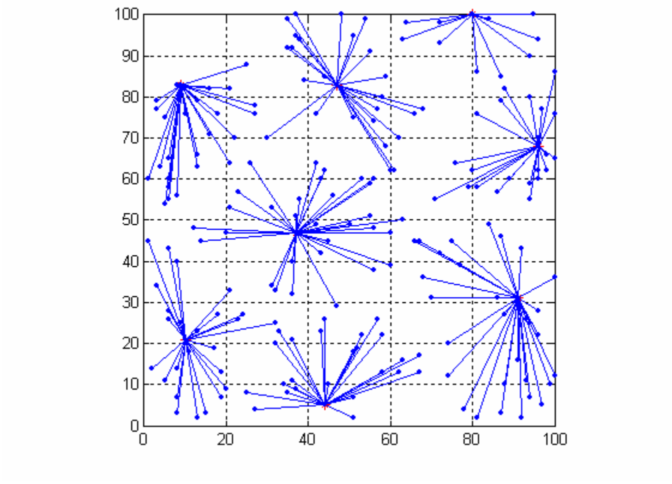
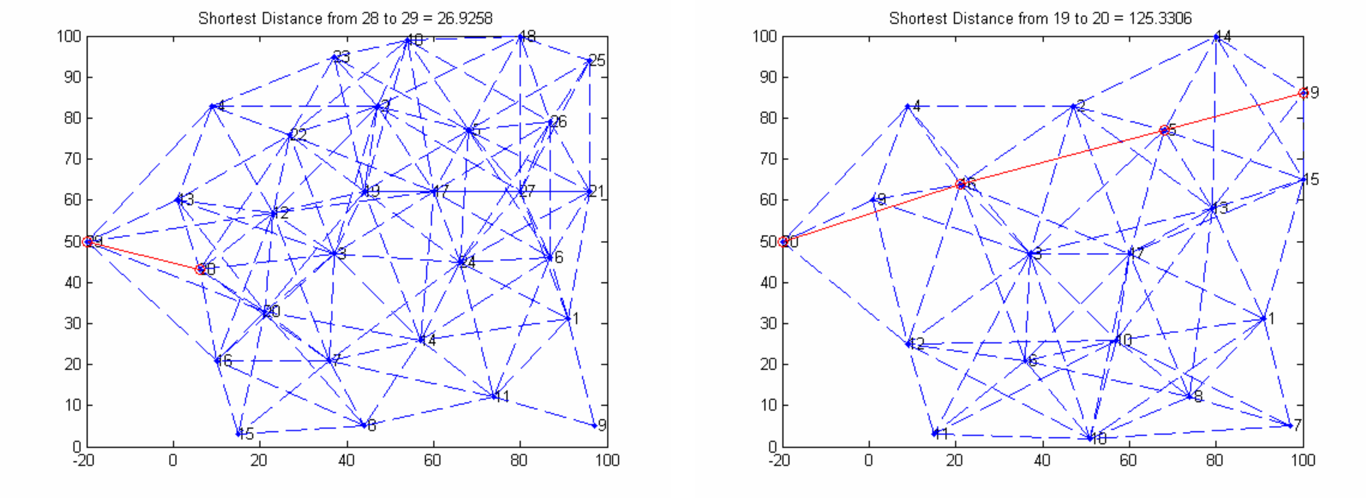


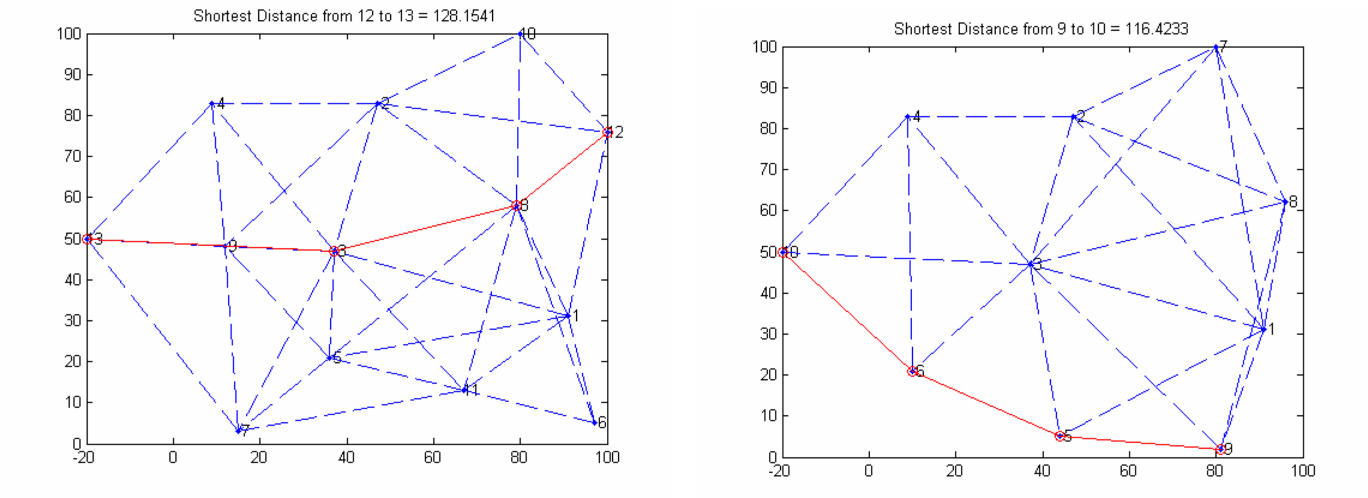
Figure 3.5: The Clustering results of the HEED algorithm at different transmission radiuses


Shortest Distance from 28 to 29 = 26.9258

Shortest Distance from 19 to 20 = 125.3306

a) Transmission radius=15 m

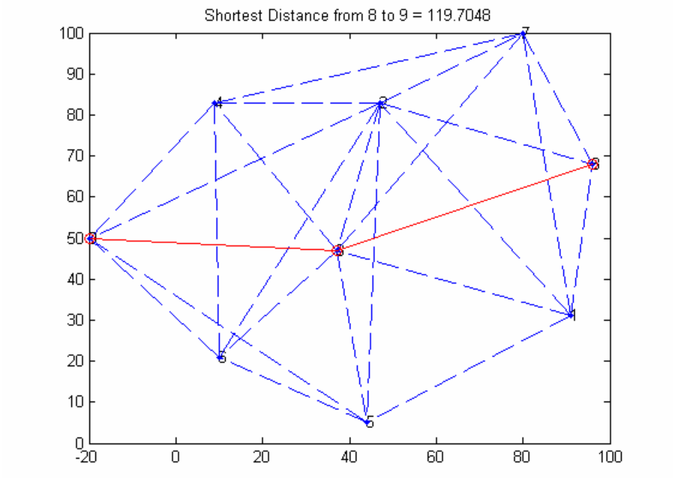
b) Transmission radius=20 m


Shortest Distance from 12 to 13 = 128.1541

Shortest Distance from 9 to 10 = 116.4233

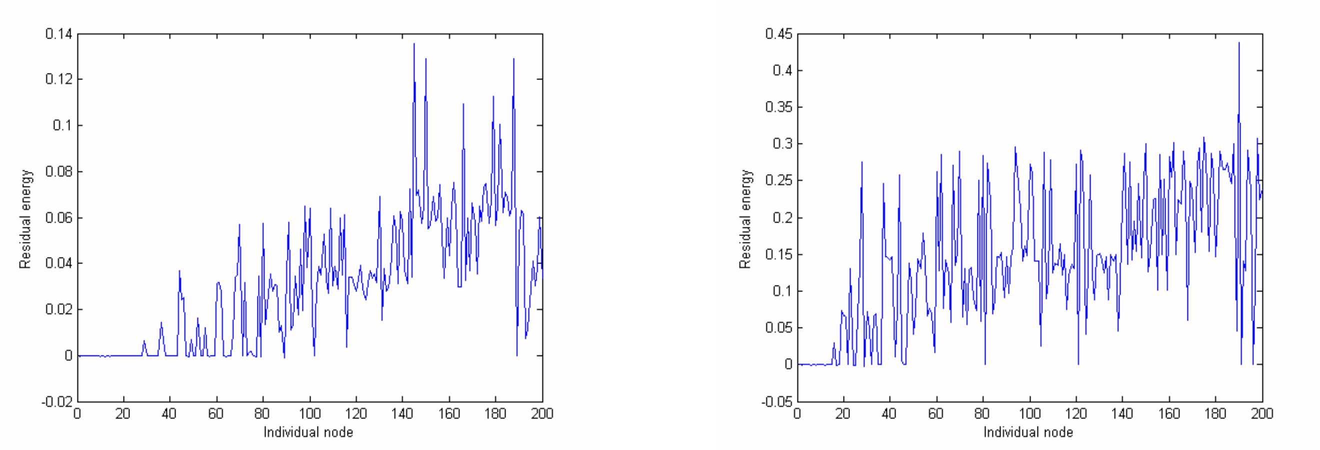
c) Transmission radius=25 m

d) Transmission radius=30 m


Shortest Distance from 8 to 9 = 119.7048

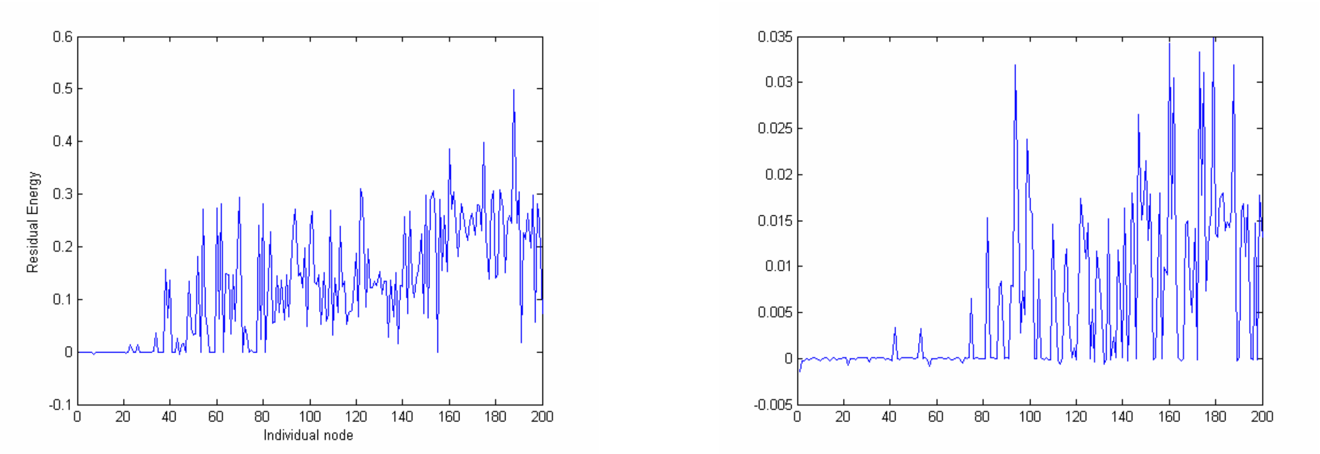
e) Transmission radius=35 m

Figure 3.6: The inter-cluster network at different transmission ranges



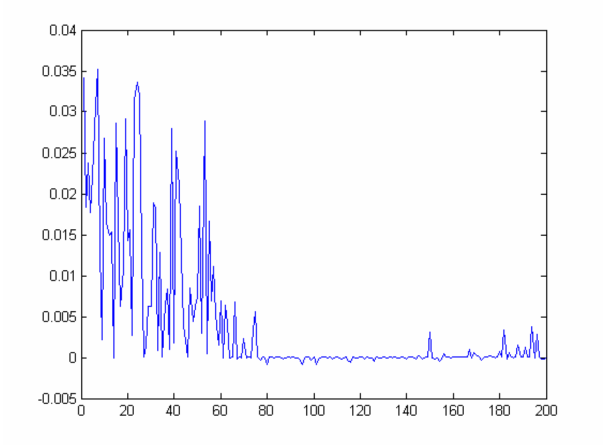
a) Transmission radius=15

b) Transmission radius=20



d) Transmission radius=30

c) Transmission radius=25



e) Transmission radius=35

Figure 3.7: Residual energy of individual nodes at different transmission radiuses

Figure 3.8 shows the throughput of the system achieved by HEED at various transmission radii until the last sensor that has a direct communication with the base station depletes its energy. The best throughput of HEED is achieved at a 25 m transmission radius.

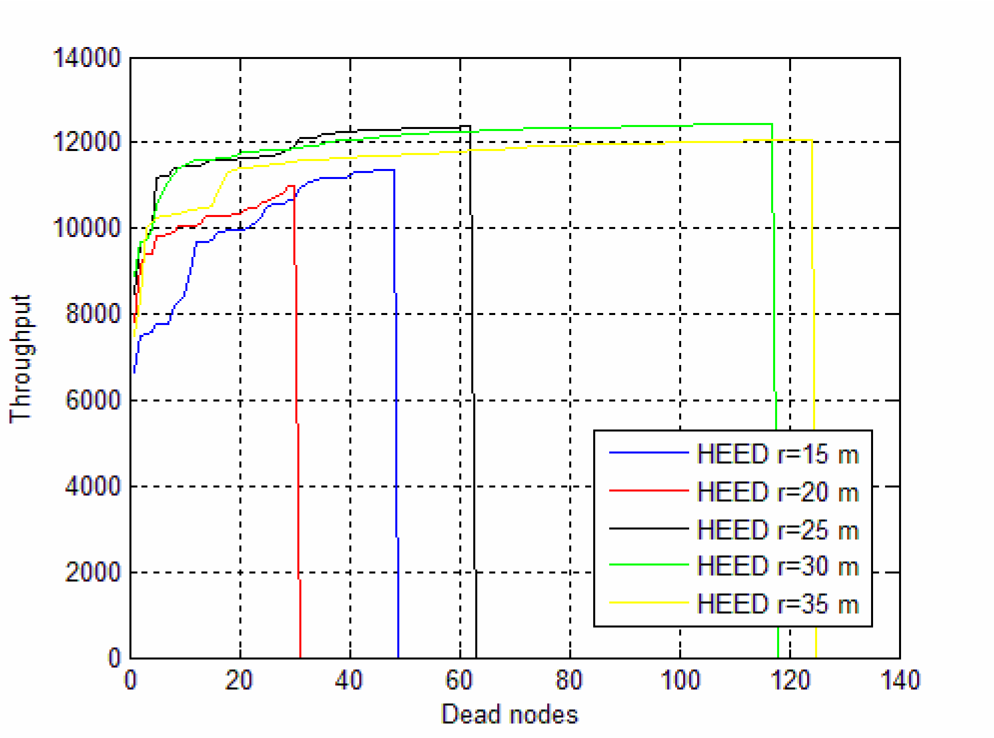


Figure 3.8: The throughput of the system from each node

It is interesting to notice that the network of 28 clusters, shown in Figure 3.5 a., outperforms the network of 19 clusters, shown in Figure 3.5 b., in terms of the throughput. A possible reason is that the BSs in the two networks have 6-k connectivity and 4-k connectivity, respectively. Thus, the node of the front region lasts longer in the first network than in the second network resulting in a longer lifetime. Figures 3.5. d and 3.5. e show two networks: 9-cluster network and 8-cluster network, respectively. Although, there is only a one-cluster difference between the two networks, and both of them have the same number of clusters in the front region, the first one outperforms the second because the second network has an extra cluster in the rear region, which results in a longer inter-cluster transmission range and the nodes in the rear region run out of energy faster than in the first network. Thus, it can be concluded that it is favourable to have fewer clusters in the rear region and more clusters in the front region, which verify the above discussion.

3.4.3 Performance Evaluation of the Proposed Algorithm

This section examines the impact of the proposed algorithm on HEED. The best performance of HEED is achieved at the 25m transmission range; so the impact of pre-processing HEED with the proposed algorithm is investigated at this particular radius. The network is divided into two regions, front and rear, since we assume that the nodes are unaware of their location. The transmission range obtained by equation 17 for this scenario of the front region nodes is 10 meters. Various transmission ranges were tested to verify that the best performance is achieved with 10-meter radius in the front region. Figure 3.9 shows that preprocessed HEED outperforms HEED by 35% in terms of the throughput of the system until the network loses its connectivity with the BS. Furthermore, the network sustains connectivity with the base station until the very last node dies.

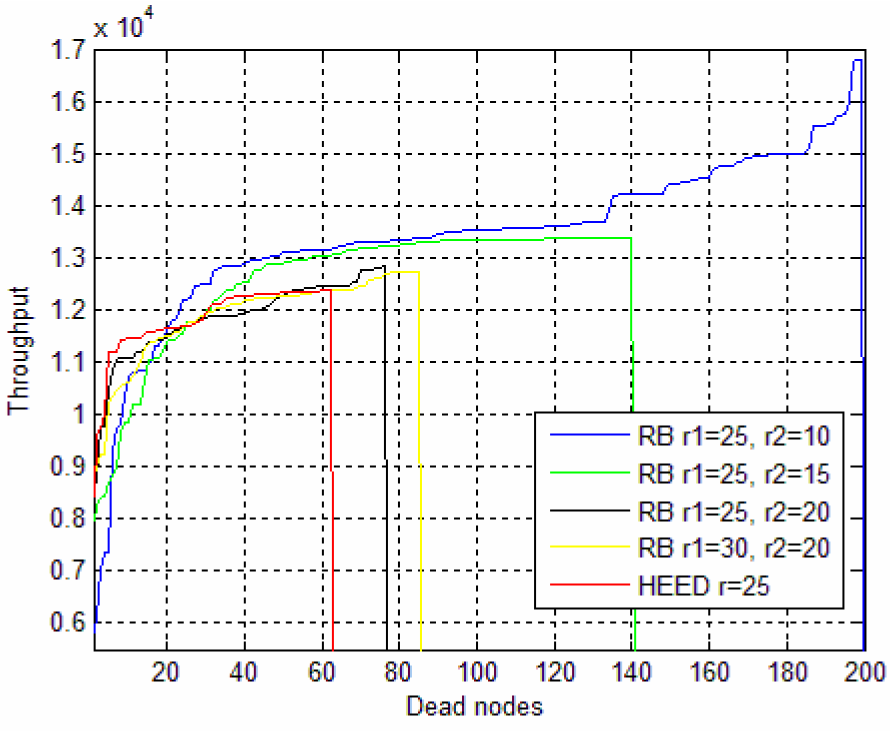


Figure 3.9: Throughput of the system from each node

Table 3.3 shows the improvement of preprocessing HEED with the proposed algorithm. A MAC protocol designed to operate over hundred of nodes would not be affected by a slight increase in the number of clusters.

Transmission Range	**# of clusters**	**Percentage improvement**
r1=25,r2=10m	25	35.9%
r1=25,r2=15m	18	7.95%
25	15	3.46%
30	13	2.72%
35	13	___

Table 3.3: Performance evaluation of the proposed algorithm

Figure 3.10 a shows that preprocessed HEED at various transmission ranges outperforms HEED in terms of the throughput of the last front-region node. The performance of HEED at the bottom and at the top preprocessed HEED at a 10 meter radius in the front region. Figure 3.10 b illustrates the average throughput per node when the network loses its connectivity with the base station arranged in a similar manner to figure 3.10 a. Many clustering algorithms have been evaluated in terms of the lifetime of the network until the last or the first node dies. However, the lifetime of each node on average is more important, and preprocessed HEED outperforms HEED in terms of the average throughput per node.

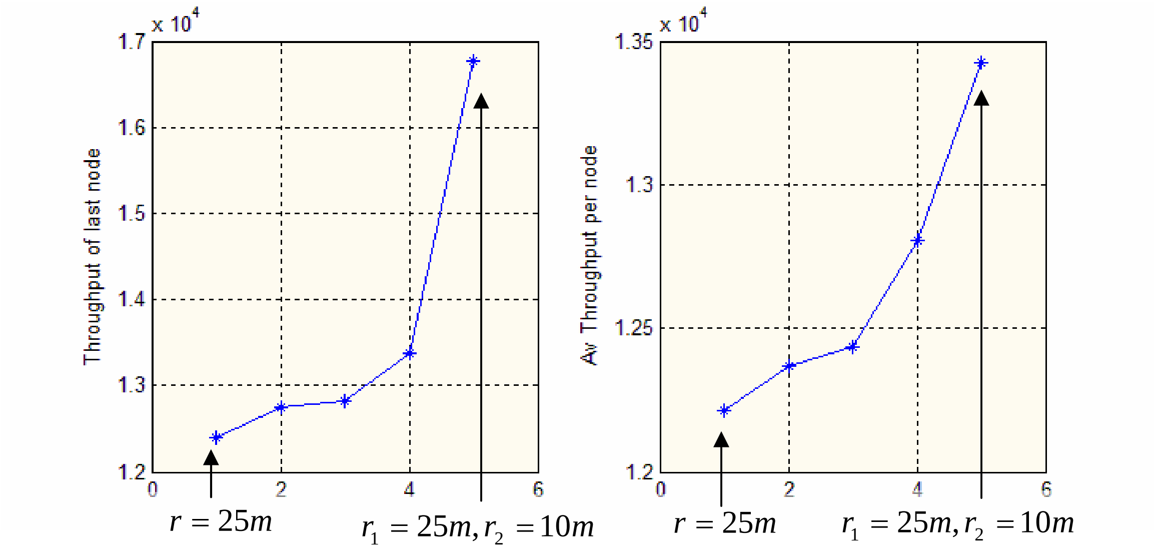


Figure 3.10: Performance comparisons between preprocessed HEED and HEED

3.5 Summary

This chapter, we formulated the extra burden of the front region nodes and proposed a transmission-tuning algorithm that can be built on top of any clustering algorithm for maximizing the lifetime of wireless sensor networks. The algorithm tries to balance the energy consumption of all clusters in order to sustain the connectivity with the sink till last node dies by allowing for different transmission radius. We showed that pre-processed HEED outperforms the actual HEED. As a future work, we need to study the effect on the functionality of the MAC protocols since we slightly increase the number of clusters in the front region.

Chapter 4

Improving the Performance of Clustered Wireless Sensor Networks by Considering Undesired Mobility

In nature erosion has an enormous effect and can be caused by many factors, such as wind, water, and ice. Given enough time, water can wear away just about any substance, as shown by the carving of the Grand Canyon by the Colorado River [5]. Wind, however, has a much greater frictional component in situations such deserts. A large mass of ice, moved by gravity, carries pieces of rocks, soil, and vegetation with it. Figure 4.1 illustrates the effect of water erosion. Small light-weight sensor nodes, deployed in such environments, are affected by erosion as well. The unpredicted movements of these sensor nodes disrupt the stability of the system and induce frequent topology changes.

In wireless sensor networks, mobility is usually not considered because of the common assumption that sensor nodes are stationary. In fact, most, if not all, applications require sensor nodes to be immobile. In such cases, clustering is the most effective technique for saving energy and prolonging the network lifetime. However, undesired mobility arises as a result of erosion or drifting, which drive nodes out of their clusters. They then detach themselves from their current cluster heads and try to join another cluster. This process is known as reaffiliation. If a node fails to reaffiliate, it triggers reclustering for the whole network. Obviously, a higher frequency of reaffiliation increases the communication overhead. Hence, energy consumption increases and the network lifetime degrades.

.

Undesired mobility caused by erosion has inspired the modeling of the impact of such mobility and the proposal of a metric that can be used as a cluster head selection criterion in order to provide a more stable network. Section 4.1 describes the types of mobility in WSNs. In section 4.1.1, cluster head selection considering undesired mobility is discussed. In section 4.2, a simple mobility model is introduced. Section 4.3 describes the use of divergence or relative entropy a metric to capture the effect of a node's mobility .A information theoretic metric that is a function of the mean and the variance derived from the divergence is also introduced.

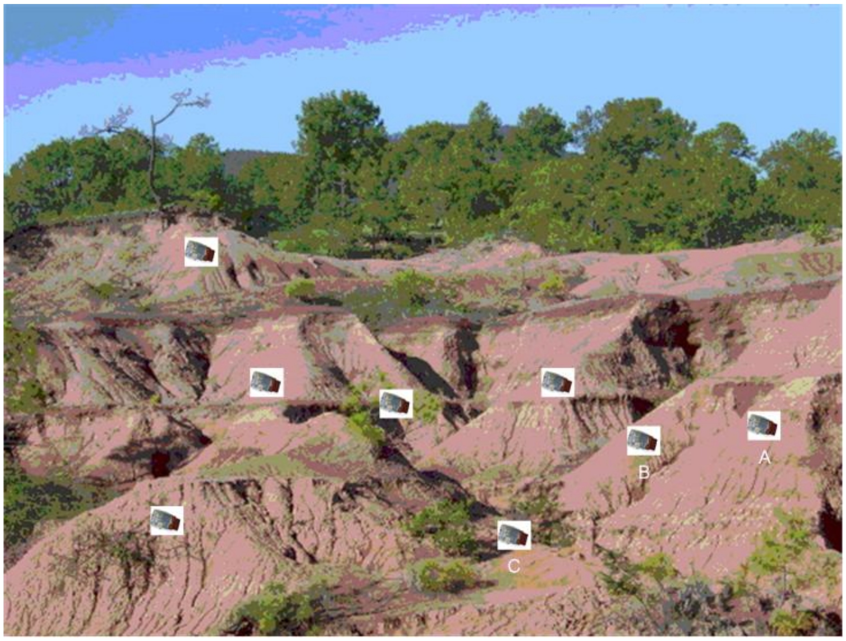


Figure 4.1: An example of real world environment

4.1 Desired Mobility vs. Undesired Mobility

Mobility in WSNs is often caused by external factors such as erosion, ocean current, etc. This type of mobility is known as uncontrolled mobility. Controlled mobility is desired to provide a level of control on network topology and increase the network capacity [73].Since controlled mobility is considered costly in terms of hardware resources, navigational need, and energy consumption; it is not feasible for most of WSNs applications or scenarios. In [70], two alternatives to provide controlled mobility are introduced: infrastructural support such as cableway or track, or limiting range of motion.

Uncontrolled mobility has been considered as an extra overhead to which the network must adapt, possibly at a loss of performance, which is called undesired mobility in this case. However, such mobility is desired form network coverage point of view in the sense that uncovered areas become covered as sensors move through them and covered areas become uncovered as sensors move away. The research in [71] studies the effect of uncontrolled mobility on the network coverage. Ma and Zhang formulated the law of motion using steepest decent method in optimization and proposed parallel and distributed network dynamics (PDND) that attempts to guide the node movements[72]. Their aim is improving WSNs sensing coverage

4.1.1 Cluster Head Selection Considering Undesired Mobility

Uncontrolled mobility is undesired from the clustering and energy consumption prospective. The clustering process should depend not only on the initial configurations but also on the mobility behavior of the sensor nodes. Although the distributed algorithms can adapt to a changing environment and re-compute a new set of cluster heads, more energy is consumed. The objective is to reduce the reaffiliation frequency in order to decrease energy consumption.

The randomness of node's mobility makes it difficult to predict the future location of the node. This randomness has a significant impact on the performance of a clustering algorithm. Predicting whether a cluster head and its members will remain neighbors, leads to fewer topology changes and more stable network.

Consider a set of sensory nodes indexed by set $\{i=1,2,\dots\}$. The current position of a sensor node *i* is denoted at time *t* as $P_i(t)$ and *t+1* as $P_i(t+1)$. Three types of neighbors can be defined:

Definition 1. Current neighbors: nodes that are within range of each other at $P_i(t)$ (the current position).

Definition 2. Future neighbors nodes that will be within range each other at $P_i(t+1)$ (the next position).

Definition 3. Relative neighbors: nodes that are current and future neighbors.

In the clustering procedure, a regular node should be assigned to a cluster head when they are relative neighbors. The election of cluster heads in most distributed clustering approaches is based on local properties, such as node ID, or residual energy. Because of the lack of information from neighbors about their undesired mobility, cluster formations generated by existing distributed clustering algorithms are often unsatisfactory in terms of stability. For example, in Figure 4.2, sensors A, B, and C all have the same communication range. Nodes A and C are elected as cluster heads and B is a regular node. Although node B is closer to A at time t, it should be assigned as a member of node C because nodes B and C are relative neighbors while nodes B and A are just current neighbors.

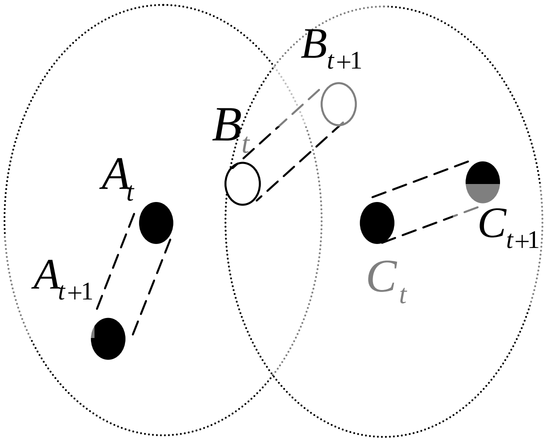


Figure 4.2: Sensor node movements

4.2 Mobility Model

Sensor nodes in relative motion can be assumed to be stationary. However, oscillation around the initial location or the movement trajectory of a sensor node is introduced because of external forces, e.g., obstacles along the drifting path. We consider a network consisting of a large number of sensors placed in a vast two-dimensional geographical region. For the initial configuration, we assume that, at time $t = 0$, the locations of these sensors are uniformly and independently distributed in the region. Such a random initial deployment is desirable in scenarios where prior knowledge of the region of interest is not available.

The random movement of a sensor is characterized by its initial location and range of motion. This movement is assumed to follow a Normal random distribution. This assumption is justifiable as the oscillation around a point over a sufficient period of time can be approximated as a Normal distribution, according to the central limit theorem. The standard deviation of the sensor's mobility is randomly chosen from a finite range $R \in [0, R_{\max}]$, according to a uniform distribution. The mobility of each sensor, due to disturbance, is assumed to follow a Normal distribution. To simplify the analysis it is assumed that the 2-dimensional motion disturbances acting on the sensor to be uncorrelated. Thus, P_x and P_y, respectively, are governed by the two probability distribution functions *p(x), p(y).* Figure 4.3 shows a typical Normal distribution capturing tendency in the 2 dimensions X and Y. The disturbed location of the sensor therefore can be described as:

$$\overset{\Delta}{P}_x(t) = P_x(t) + \eta_x \tag{18}$$

$$\overset{\Delta}{P}_y(t) = P_y(t) + \eta_y \tag{19}$$

Where the motion disturbances η_x and η_y follow Normal distributions $N(0, \sigma^2{}_x)$, and $N(0, \sigma^2{}_y)$, respectively.

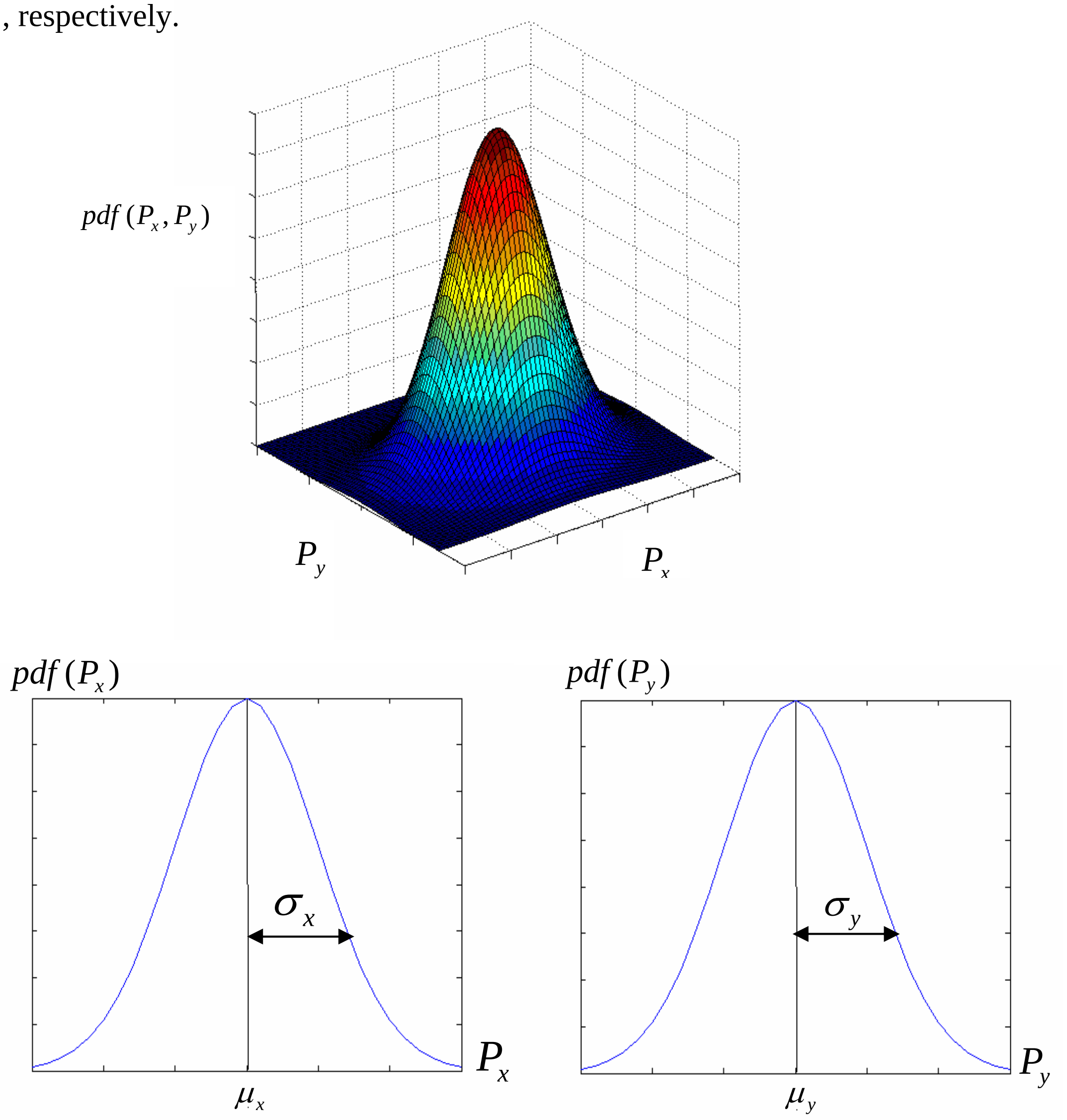


Figure 4.3: 3-D Normal distribution and its cross sections

4.3 Information Theoretic Metric to Capture the Effect of Undesired Mobility

Relative entropy, or Kullback–Leibler divergence, is a measure of the difference between two probability distributions. In addition, the divergence is used as a distance measure that captures the proximity at the current time and in the future as well. Considering one motion dimension *x,* If $p(x)$ and $q(x)$ are the mobility distributions of a node and its neighbor, respectively, then the relative motion entropy is described as Kullback–Leibler divergence $D(p//q)$. The divergence is given by

$$D(p//q) = \int p(x)\ln(p(x)/q(x)dx \qquad (20)$$

The mobility distribution of a node can be denoted as $p(x) = N(m_{x_1}, \sigma^2{}_{x_1})$ and a neighboring node mobility distribution as $q(x) = N(m_{x_2}, \sigma^2{}_{x_2})$. The distributions of a node and its neighbor are as follows:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma_{x_1}}}\exp(-\frac{(x-m_{x_1})^2}{2\sigma_{x_1}{}^2}) \qquad (21)$$

$$q(x) = \frac{1}{\sqrt{2\pi\sigma_{x_2}}}\exp(-\frac{(x-m_{x_2})^2}{2\sigma_{x_2}^2}) \qquad (22)$$

Substituting Equations 21 and 22 in Equation18 produces the following expression:

$$D_x(p//q) = \int p(x)\ln(\frac{\sigma_{x_2}}{\sigma_{x_1}} \times \exp(-\frac{(x-m_{x_1})^2}{2\sigma_{x_1}^2} + \frac{(x-m_{x_2})^2}{2\sigma_{x_2}^2}))dx \qquad (23)$$

Distributing the ln operation decomposes the expression into three terms, as follows:

$$\begin{aligned} D_x(p//q) &= \int p(x)\ln(\frac{\sigma_{x_2}}{\sigma_{x_1}})dx - \int p(x)\frac{(x^2 - 2m_{x_1}x + m_{x_2}^2)}{2\sigma_{x_1}^2}dx \\ &+ \int p(x)\frac{(x^2 - 2m_{x_2}x + m_{x_1}^2)}{2\sigma_{x_2}^2}dx \end{aligned} \qquad (24)$$

The expected value of a random variable and the expected value of a squared random variable are shown in the following Equations, respectively:

$$E(X_1) = m_{x_1} = \int x \cdot p(x)dx \qquad (25)$$

$$E(X_1^2) = \int x^2 \cdot p(x)dx \qquad (26)$$

Hence, the expression becomes

$$D_x(p//q) = \ln(\frac{\sigma_{x_2}}{\sigma_{x_1}}) - \frac{E(X_1^2) - m_{x_1}^2}{2\sigma_{x_1}^2} + \frac{E(X_1^2) - 2m_{x_1}m_{x_2} + m_{x_2}^2}{2\sigma_{x_2}^2} \qquad (27)$$

Since the expected value of a squared random variable is equal to the variance and the mean squared, shown as the following:

$$E(X_1^2) = \sigma_{x_1}^2 + m_{x_1}^2 \qquad (28)$$

The expression becomes as follows:

$$D_x(p//q) = \ln(\frac{\sigma_{x_2}}{\sigma_{x_1}}) - \frac{1}{2} + \frac{\sigma_{x_1}^2 + (m_{x_1} - m_{x_2})^2}{2\sigma_{x_2}^2} \qquad (29)$$

Similarly the divergence in the y dimension can be modeled as follows:

$$D_y(p//q) = \ln(\frac{\sigma_{y_2}}{\sigma_{y_2}}) - \frac{1}{2} + \frac{\sigma_{y_2}^2 + (m_{y_1} - m_{y_2})^2}{2\sigma_{y_2}^2} \qquad (30)$$

The final expression is a function of the mean and the variance of both distributions. Hence, the value of the divergence can be easily obtained. The exact value of the mobility would normally be determined by modeling it as a multivariate random variable. However, because considering the two dimensions separately provides sufficient information to describe the mobility, an exact value is not needed. A simpler approach is to calculate the divergence in each dimension individually and then add them to capture the proximity and relevance of the two distributions. Equation 30 shows the divergence of a node in two dimensions:

$$D(p//q) = D(p//q)_x + D(p//q)_y \qquad (31)$$

Since the divergence is not symmetric, meaning that $D\,(p//q)$ does not equal $D(q//p)$, the total the divergence between two nodes is defined as:

$$D_T = D(p//q) + D(q//p) \qquad (32)$$

This formula can be used as a metric to be considered when cluster heads are elected. For example, in [15] two separate metrics, proximity and mobility, can be replaced by this metric. Furthermore, the exact mobility is not defined in WCA while it is modeled in this work. As for HEED, this metric can be used as a combined primary metric with the residual energy or as a secondary metric to break ties among cluster head.

4.4 Divergence Based Member Election Algorithm

In a clustering algorithm, every sensor node is a candidate to be a cluster head until it hears an ADV message form a cluster head; it withdraws from the election process and joins the cluster head. A regular node needs to choose one cluster head to join when it hears an ADV from more than one cluster head. The cluster that has the smallest cost function is chosen. Figure 4.4 shows the pseudo code for breaking ties algorithm. The proposed mobility divergence metric is used as the cost function as apposed to Euclidian distance.

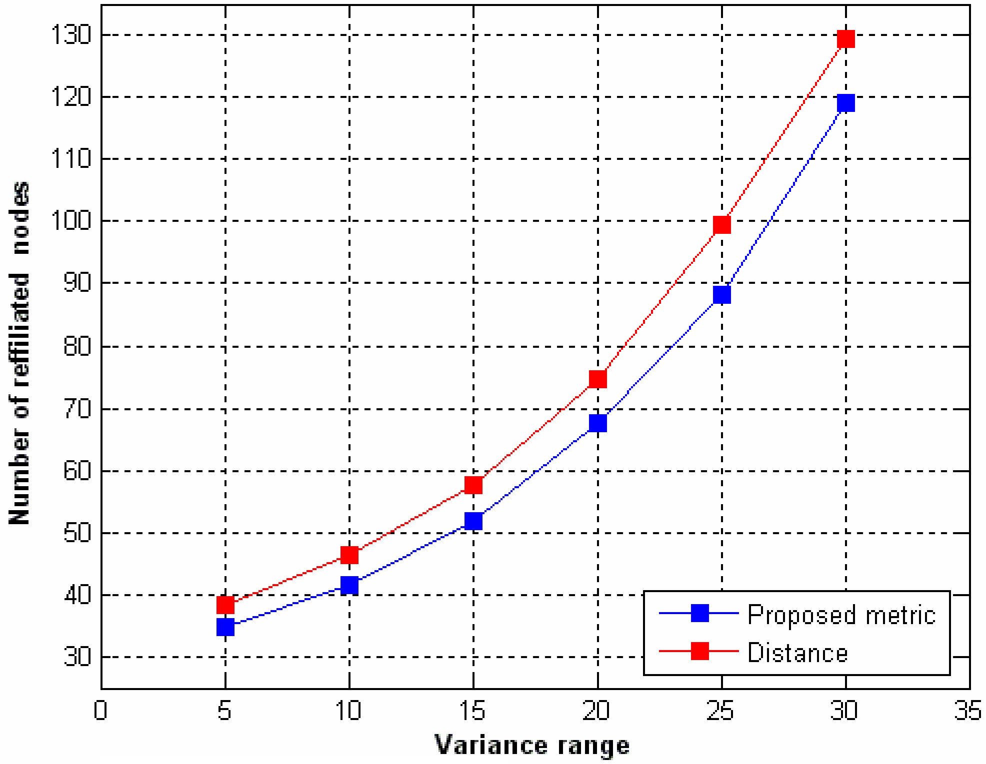
```
1. if listen (node, $S_{CH}$) is true then
2. if length($S_{CH}$)>1
3. for i=1 to length($S_{CH}$)
4. if cost function of $S_{CH}$ (i) is smallest then
5. my_cluster_head $\longleftarrow$ $S_{CH}$ (i)
6. else go to 3
7. else my_cluster_head $\longleftarrow$ $S_{CH}$ (1)
```

Figure 4.4: Algorithm pseudo code

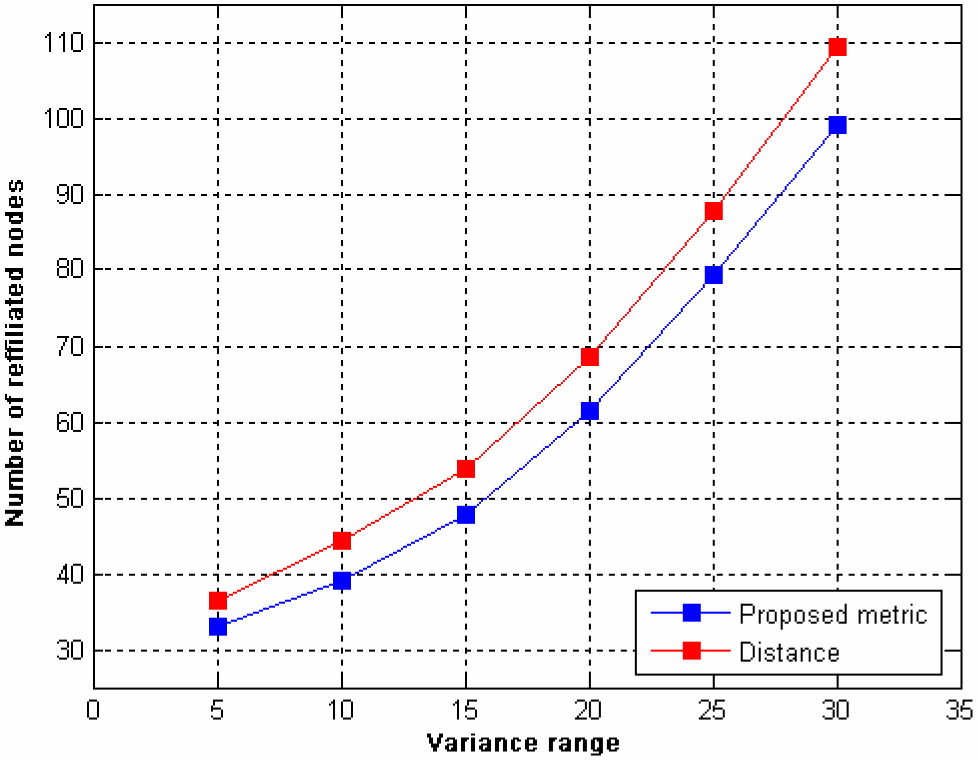
4.5 Experimental Work

The impact of using the proposed metric to break ties among cluster heads instead of the distance on HEED was examined. In our experiment, we consider a network of N sensor nodes in a 100×100 m^2 area. The nodes in the network are uniformly distributed over the area. One hundred random topologies are generated, and every node moves 100 times according to a normal distribution with the initial location of the node as the mean. The movement range of a node is controlled by the variance, which is generated according to a uniform distribution so that each node has different random variance value.

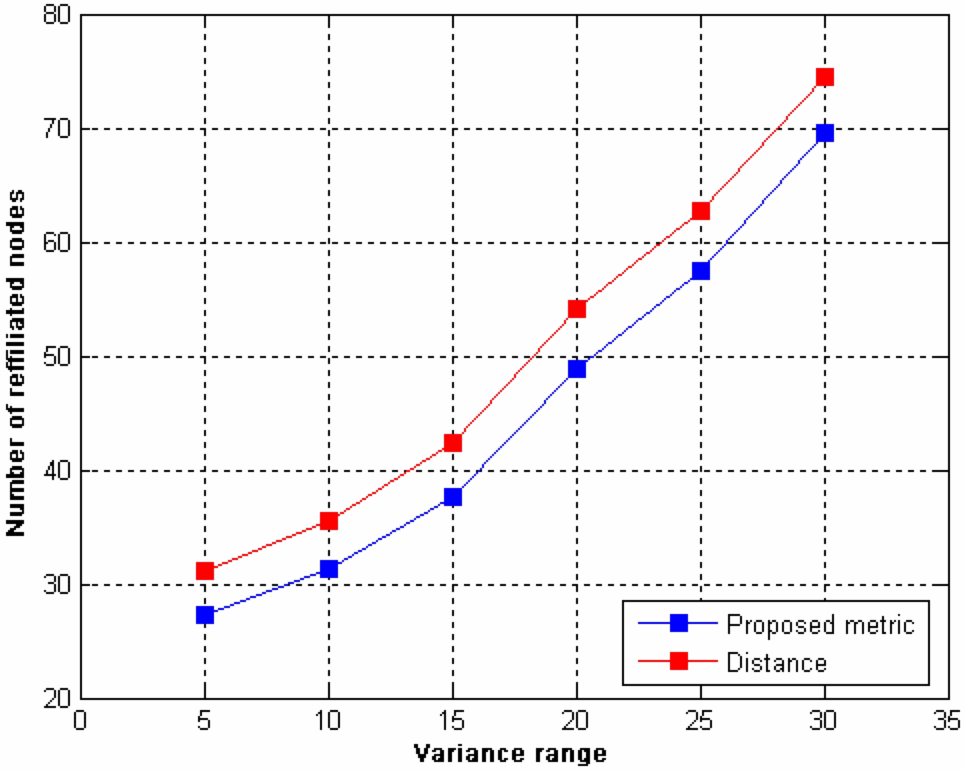
Figure 4.4 shows that using the proposed metric to break ties, reduces the number of reaffiliated nodes at different variance values. A variety of transmission ranges were tested in order to verify the effectiveness of our metric.



a) Transmission range=20m



b) Transmission range=25m



c) Transmission range=35m

Figure 4.5: Number of reaffiliated nodes at various transmission ranges

4.5 Summary

In this chapter, undesired mobility caused by erosion is introduced as a disturbance factor. To increase the stability of a cluster-based network subject to such a disturbance factor, only a CH’s relative neighbor should subscribe to it. Relative neighbor are defined as the neighboring nodes that remain neighbors at their next position. An information theoretic metric is introduced and applied to the HEED algorithm in order to reduce the ripple effect and prolong the network lifetime. The proposed metric is shown to decrease the number of reaffiliated nodes, which reduces the energy consumption and increases the longevity of the network.

Chapter 5

Conclusions and Future Work

The lower cost and easier installation of the WSNs than the wired counterpart pushes industry and academia to pay more attention to this promising technology. Large scale networks of small energy-constrained sensor nodes require techniques and protocols which are scalable, robust, and energy-efficient. Hierarchy provided by clustering techniques is an efficient approach to meet such requirements. Furthermore, data aggregation techniques are only performed by CHs.

In this thesis, the extra burden of the nodes that are within the neighborhood of the base station in a cluster-based network has been modeled as extra virtual member nodes. We virtually divide the network into regions according to the proximity to the BS denoting the closest region as the front region and the farthest region as the rear region. The nodes have been classified according to the region that they fall in. Based on our model, transmission tuning algorithm for cluster-based WSNs has been proposed to balance the load among cluster heads that fall in different regions. This algorithm is applied prior to a cluster algorithm to improve the performance of the clustering algorithm without affection the performance of individual sensor nodes. As a result, the network lifetime has been prolonged.

Undesired mobility caused by erosion is addressed in this work and modeled as a normal distribution. The nodes that are current neighbors and remain neighbors for a specific period of time are defined as relative neighbors. An information theoretic metric is proposed in an attempt to predict the random mobility of sensor nodes and is used as a metric to define the relative neighbors. The proposed metric has been used as a fitness function to break ties among cluster heads.

In the proposed implementation, a typical radio and energy consumption models are used for wireless sensor networks. The shortest path algorithm (Dijkstra's algorithm) is used to provide routs to the BS. Sensor nodes periodically transmit data to the information centre. A node is considered dead when it residual energy falls below a threshold assigned a value of E_{node}. HEED algorithm has been implemented and evaluated at various transmission radiuses. Our experiments shows that preprocessing HEED with the proposed algorithm increase the network lifetime by 35%.

The effect of the undesired mobility is studied in another set of experiments. In theses experiments each node moves according to a normal distribution model. The nodes that move out of their clusters are countered. Using our proposed metric as a fitness function has been shown to decrease the number of the reaffiliated nodes.

The proposed metric was used as a fitness function to break ties among CHs. However, it also can be combined with any other cluster head selection criteria, more research is needed to investigate the impact of such metric.

Bibliography

[1] http://www.dot.state.mn.us/i35wbridge/

[2] L. Robinson, “Japan’s new mobile broadcast company: Multimedia for cars, trains, and handhelds”. In Advances Imaging, pp 18-22, July 1998

[3] J. Agre and L. Clare, “An Integrated Architecture for Cooperative Sensing Networks. Computer”,vol 33, No. 5, pp106-108, May 2000

[4] Bluetooth Project. http://www.bluetooth.com, 1999

[5] Bult, K., and et al,”Low Power Systems for Wireless Microsensors”, In 1996 International Symposium on Low Power Electronics and Design, pp 17-21, August 1996.

[6] A. Chandrakasan, R. Amirt harajah, S.H. Cho, J. Goodman, G. Konduri, J. Kulik, W. Rabiner, and A. Wang,” Design Considerations for Distributed Microsensor Systems”, In IEEE 1999 Custom Integrated Circuits Conference (CICC), pp 279-286, May 1999.

[7] M. Dong, K, Yung, and W. Kaiser,” Low Power Signal Processing Architectures for Network Microsensors”, In Proceedings 1997 International Symposium on Low Power Electronics and Design, pp 173-177, August 1997.

[8] HomeRF Project. http://www.homerf.org, 1999.

[9] Cerpa, A.,and et al.,” Habitat monitoring: Application driver for wireless communications technology”, ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean, 2001.

[10] Mainwaring, A., et al., Wireless Sensor Networks for Habitat Monitoring, in ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'02).2002.

[11] The Great Duck Island Project. http://www.coa.edu/html/greatduckisland.htm

[12] Xu, Y., J. Heidemann, and D. Estrin," Geography-informed energy conservation for Ad Hoc routing ",ACM Press: SIGMOBILE : ACM Special Interest Group on Mobility of Systems, Users, Data and Computing,pp 70-84,2001.

[13] M.D.Yarvis," Real-world experiences with an interactive ad hoc sensor networks", International Conference on Parallel Processing Workshops, 2002.

[14] A. D. Amis, R. Prakash, T. H. P. Vuong, and D. T. Huynh, "Max-Min D-Cluster Formation in wireless Ad Hoc Networks," in Proc. IEEE INFOCOM, March 2000.

[15] M. Chatterjee, S. Das and D. Turgut, "WCA: A Weighted Clustering Algorithm for Mobile Ad hoc Networks", Journal of Cluster Computing, Special issue on Mobile Ad hoc Networking, No. 5, pp. 193-204, 2002.

[16] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocols for Wireless Microsensor Networks." In Proc. Hawaiian Int'l Conf. on Systems Science, January 2000.

[17] O. Younis and S. Fahmy, "Distributed Clustering in Ad-hoc Sensor Networks: A Hybrid, Energy-Efficient Approach", in Proc. IEEE INFOCOM 2004, Hong Kong, China, March, 2004.

[18] F. Ye, G. Zhong, S. Lu, and L. Zhang, “PEAS: A Robust Energy Conserving Protocol for Long-lived Sensor Networks,” in International Conferenece on Distributed Computing Systems *(ICDCS)*, 2003. *(Ref of HEED)*

[19] J. Han and K. Micheline, *Data Mining Concepts and Techniques*. (k-means) Morgan Kauffman, 2001.

[20] D. J. Baker and A. Ephremided,” The Architectural Organization of a Mobile Radio Network via a Distributed Algorithm”, IEEE Transactions on Communications, COM-29(11): 1964-1701, November 1981.

[21] A. Ephremided, J. E. Wieselthier, and D. J. Baker,” A Design Concept for Reliable Mobile Radio Networks with Frequency Hopping Signaling”, Proceeding of IEEE, 75(1):56-73, 1987.

[22] A. A. Abbasi, and M. Younis,” A Survey on Clustering Algorithms for Wireless Sensor networks”, Computer Communications 30: 2826-2841,2007.

[23] O. Younis, M. Krunz, and S. Ramasubramanian,” Node Clustering in Wireless Sensor networks: Recent Developments and Deployments Challenges”, IEEE Network, May/June 2006.

[24] D. Wei, and H. A. Chan,” Clustering Ad Hoc Networks: Schemes and classifications”, IEEE 2006.

[25] J.Y. Yu, and P. H. J. Chang,” A Survey of Clustering Schemes for Mobile Ad Hoc Networks”, IEEE Communications Surveys, Vol. 7, No.1, 2005.

[26] E. J. Duarte-Melo, and M. Liu,” Analysis of Energy Consumption and Lifetime of Heterogenence Wireless Sensor Networks”, IEEE, 2000.

[27] S. Bandyopadhyay and E. J. Coyle,” An energy efficient hierarchical clustering algorithm for wireless sensor networks”, IEEE INFOCOM, 2003.

[28] T. Zhong, S. Wang, S. Xu, H. Yu, and D. Xu,” Time delay based clustering in wireless sensor networks.”IEEE, 2007.

[29] S. Hussain, A. W. Matin, and O. Isalm,” Genetic algorithm for energy efficient Clusters: Wireless Sensor Networks”,

[30] Y. Wang,” An Entropy-based Weighted Clustering Algorithm and Its Optimization for Ad Hoc Netwoks”,

[31] D. Turgut, B. Turgut, R. Elmasri, and T. V. Le,“ Optimizing clustering algorithm in mobile ad hoc networks using Simulated Annealing”, IEEE Wireless Communications and Networking (WCNC), pages 1492-1497 Vol 3, 2003.

[32] K. Robinson, D. Tugut, and M. Chatterjee,” An Entropy-based Clutering in Mobile Ad Hoc Networks”,

[33] M. Qin, and R. Zimmermann,” An Energy-Efficient Voting-based Clustering Algorithm for Wireless sensor networks”, IEEE Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, pp 444-451, May 2005.

[34] A. D. Amis, and R. Prakash,” Load-Balancing Clusters in Wireless Ad Hoc Networks”, IEEE 2000.

[35] C.E. Perkins, T.J. Watson, Highly dynamic destination sequenced distance vector routing (DSDV) for mobile computers, in: ACM SIGCOMM_94 Conference on Communications Architectures, London, UK, 1994.

[36] S. Das, C. Perkins, E. Royer, Ad hoc on demand distance vector (AODV) routing, Internet Draft, draft-ietf-manetaodv- 11.txt, work in progress, 2002.

[37] D. Johnson, D. Maltz, J. Jetcheva, The dynamic source routing protocol for mobile ad hoc networks, Internet Draft, draft-ietf-manet-dsr-07.txt, work in progress, 2002.

[38] V.D. Park, M.S. Corson, A highly adaptive distributed routing algorithm for mobile wireless networks, in: Proceedings of INFOCOM, April 1997.

[39] C. Intanagonwiwat, R. Govindan and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks", in the Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'00), Boston, MA, August 2000.

[40] D. Estrin, et al., “Next century challenges: Scalable Coordination in Sensor Networks,” in the Proceedings of the 5th annual ACM/IEEE international conference on Mobile Computing and Networking (MobiCom’99), Seattle, WA, August 1999.

[41] D. Braginsky and D. Estrin, "Rumor Routing Algorithm for Sensor Networks," in the *Proceedings of the First Workshop on Sensor Networks and Applications (WSNA),* Atlanta, GA, October 2002.

[42] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient mac protocol for wireless sensor networks. In Proceedings of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002), June 2002.

[43] W. Ye, J. Heidemann, and D. Estrin. Medium access control with coordinated, adaptive sleeping for wireless sensor networks. In IEEE Transactions on Networking, Apr. 2004.

[44] A. Woo and D. E. Culler,” A transmission control scheme for media access in sensor networks”, In Proceedings of the seventh annual international conference on mobile computing and networking, July 2001.

[45] IEEE 802.11 Standard Group Web Site. Available from: <http://www.ieee802.org/11/>.

[46] S. Singh and C.S. Raghavendra, “PAMAS: Power aware multi-access protocol with signaling for ad hoc networks,” *ACM Computer Communication Review*, vol. 28, no. 3, pp. 5–26, July 1998.

[47] L. Kleinrock and F. Tobagi, “Packet switching in radio channels: Part I - carrier sense multiple access modes and their throughput delay characteristics,” IEEE Transactions on Communications, vol. 23, no. 12, pp. 1400–1416, Dec. 1975.

[48] Phil Karn, “MACA: A new channel access method for packet radio,” in *Proceedings of the 9th ARRL Computer Networking Conference*, London, Ontario, Canada, Sept. 1990, pp. 134–140.

[49] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, “MACAW: A media access protocol for wireless lans,” in *Proceedings of the ACM SIGCOMM Conference*, London, UK, Sept. 1994, pp. 212–225.

[50] T. van Dam and K. Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In Proceedings of the First ACM Conference on Embedded Networked Sensor Systems, Nov.2003.

[51] J. Polastre, R. Szewczyk, D. Culler,” Versatile low power media access for wireless sensor networks”, In Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys), November 3-5, 2004.

[52] F. Ye *et al.*, “A Scalable Solution to Minimum Cost Forwarding in Large Sensor Networks,” Proc. 10th Int’l. Conf. Comp. Commun. and Networks, pp 304-309,2001.

[53] F. Ye *et al.*, “A Two-Tier Data Dissemination Model for Large-Scale Wireless Sensor Networks,” *Proc. ACM/IEEE MOBICOM*, 2002.

[54] I. Akyildiz *et al.*, “A Survey on Sensor Networks,” *IEEE Commun. Mag.*, vol. 40, no. 8, Aug. 2002, pp. 102–14.

[55] S. Tilak, N. Abu-Ghazaleh, W. Heinzelman, “A Taxonomy of Wireless Micro-sensor Network Models,” *ACM SIGMOBILE Mobile Comp. Commun. Rev.*, vol. 6, no. 2, pp. 28–36, Apr. 2002.

[56] S. Hedetniemi and A. Liestman, “A Survey of Gossiping and brocadcasting in Communication Networks,” IEEE Network, vol. 18, no. 4, pp. 319–49, 1988.

[57] K. Dasgupta, M. Kukreja, and K. Kalpakis, “Topology-aware placement and role assignment for energy-efficient information gathering in sensor networks”, in Proceedings of 8th IEEE Symposium on Computers and Communication (ISCC’03), Kemer-Antalya, Turkey,July 2003.

[58] A. Manjeshwar and D. P. Agarwal, “TEEN: a Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks,” 1st Int’l. Wksp. on Parallel and Distrib. Comp Issues in Wireless Networks and Mobile Comp., April 2001

[59] A. Manjeshwar and D. P. Agarwal, “APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks,” *Proc. Int’l. Parallel and Distrib. Proc. Symp.*, pp. 195–202.

[60] C. R. Lin and M. Gerla, “Adaptive Clustering for Mobile Wireless Networks,” in IEEE J. Select. Areas Commun., September 1997.

[61] S. Banerjee and S. Khuller, “A Clustering Scheme for Hierarchical Control in Multi- hop Wireless Networks,” in *Proceedings of IEEE INFOCOM*, April 2001.

[62] S. Basagni, “Distributed Clustering Algorithm for Ad-hoc Networks,”in *International Symposium on Parallel Architectures, Algorithms, andNetworks (I-SPAN)*, 1999.

[63] T. J. Kwon and M. Gerla, “Clustering with Power Control,” in Proceeding of MilCOM’99, 1999.

[64] S. Bandyopadhyay and E. Coyle, “An Energy-Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks,” in *Proceedings of IEEE INFOCOM*, April 2003.

[65] V. Kawadia and P. R. Kumar, “Power Control and Clustering in Ad Hoc Networks,” in *Proceedings of IEEE INFOCOM*, April 2003.

[66] M. Gerla, T. J. Kwon, and G. Pei, “On Demand Routing in Large Ad Hoc Wireless Networks with Passive Clustering,” in *Proceeding of WCNC*, 2000.

[67] J.N. Al-Karaki, and A.E. Kamal, “Routing techniques in wireless sensor networks: A survey”, IEEE wireless communication, vol 11, No. 6, pp 6-28, Dec 2004.

[68]] I. Demirkol, C. Ersoy, and F.Alagoz,” Mac protocols for wirelss sensor networks: A survey”, IEEE communication magazine, vol 44, No. 4, pp 115-121,Apr 2006.

[69] S. S. Kulkarni, “TDMA Services for Sensor Networks,” *Proc. 24th Int’l. Conf. Distrib. Comp. Sys. Wksps*, pp. 604–09., Mar. 2004.

[70] A. Kansal, M. Rahimi, D.Estrin, W. Gregory, J. Pottie, and M.B Srivastava,” Controlled mobility for sustainable wireless sensor networks”,IEEE 2004.

[71] B. Liu, P. Brass, and O. Dousse,” Mobility Improves Coverage of Sensor Networks”, *MobiHoc’05,* May 25–27, 2005.

[72] K. Ma, and Y. Zhang,” Managing the Mobility of a Mobile Sensor Network Using Network Dynamics,” IEEE Transaction on parallel and distributed systems, Vol. 19, No. 1, January 2008.

[73] M Grossglauser and DNC Tse. Mobility increases the capacity of ad hoc wireless networks. *IEEE/ACM Transactions on Networking (TON)*, 10(4):477–486, 2002.