**CLUSTER-BASED ROUTING PROTOCOL
FOR MOBILE SENSOR NETWORKS**

**A Thesis
Presented to
The Faculty of Graduate Studies
of
The University of Guelph**

**by
LILIANA MARIA ARBOLEDA COBO**

**In partial fulfillment of requirements
for the degree of
Master of Science
May, 2007**

© Liliana Arboleda, 2007

Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 978-0-494-33828-5

Our file Notre référence

ISBN: 978-0-494-33828-5

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

ABSTRACT

CLUSTER-BASED ROUTING PROTOCOL FOR MOBILE SENSOR NETWORKS

Liliana María Arboleda Cobo
University of Guelph, 2007

Advisor:
Professor Nidal Nasser

Hierarchical routing and clustering mechanisms in Wireless Sensor Networks (WSN) help to reduce the energy consumption and the overhead created when all the sensor nodes in the network are sending information to the central data collection point or base station. Most of the routing and clustering protocols currently used or proposed for WSN assume that the nodes are stationary. However, in applications like habitat monitoring or search and rescue, that assumption makes those clustering mechanisms invalid, since the static nature of sensors is not real.

In this thesis, we present a Cluster-based ROuting protocol for MObile Sensor networks (CROMOS). The protocol considers the following design aspects: mobility of sensors, zones and routes maintenance, information update and communication between sensor nodes. A simulation model has been designed and developed to evaluate the performance of the proposed protocol. Simulation results and comparisons with different scenarios show the effectiveness and strengths of the CROMOS protocol. CROMOS shows a low routing and mobility overhead, while achieving a good performance in WSN using small zone sizes and sensors with low speed.

Acknowledgements

I extend my thanks to my advisor, Professor Nidal Nasser, for his guidance and collaboration with this project. Prof. Nasser's permanent follow-up, support, suggestions and ideas have greatly enriched the content of my research.

While at University of Guelph, I had the privilege of interacting with wonderful and talented people. Thanks to all of them, who helped me either with their comments and suggestions, with their help related to non-academic matters or by giving some fresh air to my days during this research.

I would like to thank my family for their love, encouragement and support. Thanks to my parents, Segundo y Alicia, for teaching me the value of education, the joy of love and the importance of family; to Rubiela for her love and dedication; to my brother Alejandro and my syster-in-law Alejandra, for giving us a new joy: Santiago and to Ruth, my mother-in-law for her support during my new experiences in Canada.

Finally, I would like to thank my husband Jason. Jason's love and support have been decisive when giving me the encouragement I needed for completing this new step of my professional and academic career.

Liliana María Arboleda Cobo

Department of Computing and Information Science

University of Guelph

May, 2007

Table of Contents

CHAPTER 1 INTRODUCION	1
0.0. Wireless Sensor Networks Components	3
0.0.0. Sensor Nodes (SN)	3
0.0.0. Base Station (BS)	4
0.0. Clustering and Wireless Sensor Networks	4
0.0.0. Clusters	5
0.0.0. Cluster Heads (CH)	7
0.0. Types of Clusters in WSN	8
0.0. Communication Patterns in WSN	10
0.0. Static Clustering vs. Dynamic Clustering	12
0.0. Clustering Process	13
0.0. Importance of Routing	15
0.0. The Role of Mobility	17
0.0. Cluster-based Routing Protocol for Mobile Sensor Networks Objectives ...	17
0.0. Targeted Applications	18
0.0.0. Cattle Herding.....	19
0.0.0. Turtle Tracking	19
0.0.0. More Applications	20
 CHAPTER 2 RELATED WORK: CLUSTER-BASED PROTOCOLS	 22
1.0. Dynamic Clustering for Acoustic Target Tracking	22
1.0. LEACH-Based Protocols	24
1.0.0. LEACH: Low Energy Adaptive Clustering Hierarchy	25
1.0.0. LEACH-C: LEACH-Centralized	28
1.0.0. LEACH-F: LEACH with Fixed Cluster and Rotating Cluster-Head	29
1.0. Protocols for Proactive Networks	30
1.0.0. HEED: Hybrid Energy-Efficient Distributed Clustering	31
1.0.0. EECS: Energy Efficient Clustering Scheme	32
1.0.0. Sensor Aggregates	34
1.0.0. ACE: Algorithm for Cluster Establishment	37
1.0.0. EEDC: Energy Efficient Data Collection	42
1.0.0. TASC: Topology Adaptive Spatial Clustering	44
1.0. Protocols for Reactive Networks	47
1.0.0. TEEN: Threshold-sensitive Energy Efficient Sensor Network	48
1.0.0. APTEEN: Adaptive Periodic Threshold-sensitive Energy Efficient sensor Networks protocol	50
1.0.0. CAG: Clustered AGgregation technique	52
1.0.0. Upgraded CAG	55
1.0. Summary	57
 CHAPTER 3 CLUSTER-BASED ROUTING PROTOCOL FOR MOBILE SENSOR NETWORKS	 59

2.0. Cluster-Based Routing Protocol Overview	59
2.0.0. Protocol Assumptions	59
2.0.0. Cluster-Based Routing Protocol for Mobile Sensor Networks Stages ...	62
2.0.0.0. Zones Definition Stage	62
2.0.0.0. Cluster Head Election Stage	64
2.0.0.0. Zone-Based Routing Stage	65
2.0. System Parameters	66
2.0.0. Network Initialization	69
2.0. Zones Creation Process	72
2.0.0. Considerations for the Information Update	75
2.0. Cluster Head Election Process	77
2.0.0. Mobility Factor	78
2.0.0. Cluster Head Election Criteria	78
2.0.0. Zone Head Maintenance	81
2.0. Neighbor Discovery Process	81
2.0. Zone-Based Routing Process	82
2.0.0. Zone-Based Routing Operation	83
2.0.0.0. Overview of Route Discovery Procedure	83
2.0.0.0. Description of Zone-Based Routing Protocol	84
2.0.0.0. Route Maintenance	87
2.0. Summary	87
 CHAPTER 4 PERFORMANCE EVALUATION	 89
3.0. Simulation Model	90
3.0.0. Sensors Deployment	90
3.0.0. Energy Model	91
3.0.0. Mobility Model	94
3.0.0. Traffic Model	95
3.0. Simulation Results	95
3.0.0. WSN with Small Zones and Low Velocity	98
3.0.0. WSN with Big Zones and Low Velocity	104
3.0.0. WSN with High Velocity	106
3.0. Discussion of Performance of CBRP	109
3.0.0. The Effect of Clustering	109
3.0.0. The Effect of Mobility	110
3.0.0. The Effect of Nodal Density	111
3.0.0. The Effect of Zone Size	112
3.0.0. The Effect of Using Levels	112
3.0. Summary	113
 CHAPTER 5 CONCLUSIONS	 115
4.0. Future Work	116
 BIBLIOGRAPHY	 117

List of Tables

TABLE 2.1	Clustering Protocols for WSN	57
TABLE 3.1	Initialization Parameters	70
TABLE 4.1	Radio Model Values	94

List of Equations

EQUATION (2-1)	Threshold Value in LEACH	26
EQUATION (3-1)	Zone Size ₁	73
EQUATION (3-2)	Zone Size ₂	73
EQUATION (3-3)	Zone ID's i Coordinate Calculation	75
EQUATION (3-4)	Zone ID's j Coordinate Calculation	75
EQUATION (3-5)	Update Timer Value	76
EQUATION (3-6)	Update Timer Value for Zone Size ₁	76
EQUATION (3-7)	Update Timer Value for Zone Size ₂	77
EQUATION (3-8)	Mobility Factor	78
EQUATION (4-1)	Energy Expended to Transmit a k -bit Message with a Distance d	93
EQUATION (4-2)	Energy Expended to Receive a k -bit Message	93

List of Figures

FIGURE 1.1	Wireless Sensor Networks Components	3
FIGURE 1.2	Wireless Sensor Networks Components, with Clustering	6
FIGURE 1.3	Clustering Process Phases	15
FIGURE 2.1	LEACH-Based Protocols	24
FIGURE 2.2	Flow-Graph of the Distributed Cluster Formation Algorithm for LEACH	26
FIGURE 2.3	Clustering Protocols for Proactive Networks	30
FIGURE 2.4	ACE (Algorithm for Cluster Establishment)	38
FIGURE 2.5	Clusters in TASC	46
FIGURE 2.6	Clustering Protocols for Reactive Networks	48
FIGURE 2.7	Time Line for TEEN	49
FIGURE 2.8	Time Line for APTEEN	51
FIGURE 3.1	Different Sensor Fields' Shapes Circumscribed Into Squares ...	63
FIGURE 3.2	Zone IDs	64
FIGURE 3.3	Locate Zone Method	71
FIGURE 3.4	Definition of the Sensor's Next Target Location	72
FIGURE 3.5	Zone Size and Neighboring Zones in the Cluster Field	74
FIGURE 3.6	Update Time Scenario.....	76
FIGURE 3.7	Cluster Head Election Process	80
FIGURE 3.8	Neighbor Discovery Process	82
FIGURE 3.9	Previous CH Selection	86
FIGURE 4.1	Simulator's GUI	90
FIGURE 4.2	First Order Radio Model	93
FIGURE 4.3	Simulation Results for WSN with Small Zones and Low Velocity	100
FIGURE 4.4	Simulation Results for WSN with Small Zones and Low Velocity, Combining Mobility Factor and Remaining Energy	103
FIGURE 4.5	Simulation Results for WSN with Big Zones and Low Velocity ...	105
FIGURE 4.6	Simulation Results for WSN with High Velocity	108
FIGURE 4.7	Packet Loss Rates	109

List of Acronyms

ACE	Algorithm for Cluster Establishment
APTEEN	Adaptive Periodic Threshold-sensitive Energy Efficient sensor Networks protocol
BS	Base Station
CAG	Clustered AGgregation technique
CDMA	Code Division Multiple Access
CH	Cluster Heads
CM	Cluster Members
CROMOS	Cluster-based ROuting protocol for MObile Sensor networks
DAM	Distributed Aggregate Management
EBAM	Energy-Based Activity Monitoring
EECS	Energy Efficient Clustering Scheme
EEDC	Energy Efficient Data Collection
EMLAM	Expectation-Maximization Like Activity Monitoring
GPS	Global Positioning System
HEED	Hybrid Energy-Efficient Distributed Clustering
LEACH	Low Energy Adaptive Clustering Hierarchy
LEACH-C	LEACH Centralized
LEACH-F	LEACH with Fixed Cluster and Rotating Cluster-Head
MAC	Medium Access Control
MMSE	Minimum Mean Square Estimation
MMV	Maximum Moving Velocity
MTR	Maximum Transmission Range
RREP	Route Reply Message
RREQ	Route Request Message
SN	Sensor Nodes
TASC	Topology Adaptive Spatial Clustering
TDMA	Time Division Multiple Access
TEEN	Threshold-sensitive Energy Efficient Sensor Network
WSN	Wireless Sensor Network

CHAPTER 1

INTRODUCTION

A Wireless Sensor Network (WSN) consists of a large set of *wireless sensors* (also called *wireless nodes*) deployed over a geographic area known as the *sensors field*, to monitor specific characteristics of the environment or characteristics of specific subjects immerse in that environment. Wireless sensors also collect the data related to that phenomena and forward it to a central unit for processing and analysis. Examples of the data that can be collected include temperature, humidity, light conditions, seismic activities, etc. This data can be used to detect certain events and to trigger activities. For example, sensors distributed over a large woodland could automatically raise an alarm if a fire has broken out somewhere [Ben1]. In another example, sensors distributed over a large farmland could trigger irrigation if the ground of a field is not moist enough [Ben1].

Sensors are small devices with limited resources: limited battery power, low memory, little computing capability, very low data rates, low bandwidth processing, variable link quality, etc. However, despite their constraints, when the sensors are deployed in large numbers they can provide us with a very real picture of the field being sensed. WSN can provide an area coverage that was not possible with other wired and wireless networks.

When creating a WSN, in most cases the sensors are deployed randomly in the sensor field, without prior knowledge of the network topology. This reduces the complexity of

having to locate the nodes one by one on specific places and also allows their use in constrained environments like military fields or wild terrains. Once deployed, the sensors communicate wirelessly among each other, and send the collected data towards a central point called the *base station* or *sink*. This central data collection point allows the processing of the collected data and the connection of the WSN with other networks, either WSN or other types of networks like a wired corporate network or the Internet.

In order to control the transmission of the sensed data towards the base station, several routing protocols had been proposed, from basic flat routing protocols based on flooding to complex routing protocols that use graph networks to manage the distribution of the messages inside the WSN. However, those protocols generally assume the wireless sensors to be stationary and not change their position once they have been deployed in the sensor field. This thesis focuses on proposing a routing protocol that considers mobile wireless sensors, while maintaining the same performance level obtained in previously proposed protocols for stationary wireless sensors.

In the following section we explain in detail the components of a WSN. We then review different schemes from the literature which show how to organize those elements and obtain a better communication structure inside the WSN. We also provide definition for different terms that we use in this thesis. In addition, in this chapter we explain the importance of routing in WSN and the challenges introduced by the sensor's mobility. Finally, we explain the objectives of our proposed cluster-based routing protocol for

mobile sensor networks and give some examples of the type of applications that our protocol will be useful for.

1.1. Wireless Sensor Networks Components

Two main elements constitute a WSN (Figure 1.1): the wireless sensor nodes that sense the data within the sensor field and the base station that collects the data and generally processes the information required by the end-user.

Figure 1.1: Wireless Sensor Networks Components

1.1.1. Sensor Nodes (SN)

Sensor nodes are a set of devices present in the network, arranged to sense the environment and collect the data. These sensors are generally stationary, but in a

wireless mobile environment, they can be mobile. The nodes can be location-aware (using a Global Positioning System - GPS) or location-unaware, homogeneous (same construction characteristics) or heterogeneous. All types of sensors are energy-constrained, and can show different energy levels. The sensors in the WSN can be deployed in different environments and can be permanently attended or can be left unattended once they have been deployed in the field.

The main task of a SN in a sensor field is to detect events, perform quick local data processing, and then transmit the data. But the greatest constraint is the power consumption, which is usually caused when the sensor is observing its surroundings, and communicating (sending and receiving) data.

1.1.2. Base Station (BS)

The base station is the data processing point for the data received from the sensor nodes, and where the data is accessed by the end-user. It is generally considered fixed and far from the sensor nodes. The BS has constant power supply and no energy constraints. It provides a direct path for the down-link channel. The communication between the BS and the sensor nodes can be symmetric or asymmetric.

1.2. Clustering and Wireless Sensor Networks

The use of the WSN potential provides efficient and cost-effective solutions for several problems. However, it is necessary to implement mechanisms or procedures to deal with

the sensors' constraints. Due to the limitation of their physical size, the sensors tend to have storage space, energy supply and communication bandwidth so limited that every possible means of reducing the usage of these resources is aggressively sought. The use of clustering techniques has been proposed to help solve some of those constraints, by allowing the organization of the sensors in a hierarchical manner, grouping them into clusters and assigning specific tasks to the sensors in the clusters, before forwarding the information to higher levels.

In general terms, clustering is the classification of similar objects into different groups or subsets. The formed subsets in some sense belong together, because they share one or more similar characteristics or behaviors. Examples of such common characteristics could be: proximity according to some defined distance measure, similar behaviors, common data patterns, etc. In the most general problem the number of clusters or groups is unknown as are the properties that make them similar. The concept of clustering is very useful in different contexts of WSN. Clustering is a fundamental mechanism to design scalable sensor network protocols. The use of clustering introduces two new components to the WSN: clusters and cluster heads.

1.2.1. Clusters

A cluster in WSN means to interconnect the sensor nodes hierarchically in order to improve their performance. Clusters provide reliability by having sensors taking over the workload of failing sensors. Clusters also improve data aggregation mechanisms, reduce the workload of each sensor to save energy and thus increase the overall lifetime of the

system. The hierarchical organization of sensors helps in making a better management of the available bandwidth. Figure 1.2 shows a WSN using clusters.

Figure 1.2: Wireless Sensor Network Components, with Clustering

The creation of clusters in a WSN field is generally done taking into account the proximity between the sensors, measured through the radio frequency signal they emit. Clustering helps solve some of the sensors' constraints by: reducing the cost of transmitting data to base stations, reducing the power consumption in the devices, facilitating the gathering of sensed data, maximizing the routing process execution and allowing scalability.

1.2.2. Cluster Heads (CH)

The cluster head sensor is one of the nodes in the cluster and it acts as a gateway between the sensor nodes and the BS. The role of the cluster head is to perform common functions for all the nodes in the cluster, like aggregating the data before sending it to the BS. Some of the purposes of having only one cluster getting connected to the BS are to save energy in the other nodes, reduce the number of nodes contending for the channel access, and minimize the control overhead. If the purpose is to save energy, the role of CH must rotate between the network's sensor nodes all the time to avoid having only one node spending all its energy, because this would cause its death and a cascading effect that would cause the failure of the whole system.

In some way, the cluster head is the sink for the cluster nodes, and the base station is the sink for the cluster heads. This structure formed between the sensor nodes, the cluster heads and the base station can be replicated as many times as it is needed, creating the different layers of the hierarchical wireless sensor network.

The sensors that are not acting as CH in the WSN are cluster members (CM). By assuming roles within a cluster hierarchy, the nodes in a WSN can control the activities they perform and therefore, reduce their energy consumption. However, the election of when to act as a simple data provider (saving energy) and when to act as a gateway (cluster head) between the nodes and the base station is a difficult problem. To make this decision it is necessary to take into account several aspects including power level signal,

transmission schedules, nodes localization and networking function (proactive or reactive).

Clustering techniques have been proposed in wireless networks in order to achieve high energy efficiency and assure long network lifetime, for bandwidth reuse, for data gathering [Yem1] and target tracking [Che1], one-to-many, many-to-one, one-to-any, or one-to-all communications, routing [Alk1][Hei2][Liu1][Man1], etc. Clustering is particularly useful for applications that require scalability to hundreds or thousands of nodes. Scalability in this context implies the need for load balancing, efficient resource utilization, and data aggregation [Aur1]. Also, many routing protocols can use clustering to create a hierarchical structure and minimize the path cost when communicating with the base station.

In many sensors network applications where data collection and processing can be done “in-situ”, this hierarchical approach is a promising method for efficiently organizing the network. Also, many signal processing algorithms used for extraction of final information from the data gathered by the sensors are well suited for local processing of data within the clusters.

1.3. Types of Clusters in WSN

Clusters in wireless sensor networks can be broadly classified as heterogeneous and homogeneous with respect to the type, characteristics and functionality of the nodes in the network.

In heterogeneous sensor networks, there are generally two types of sensors:

- a) Sensors with high processing capabilities and complex hardware, used generally to create some sort of backbone inside the WSN. They are designated as the cluster head nodes, and therefore have to serve as data collectors and processing centers for data gathered by other sensor nodes.
- b) Member sensors, with lower capabilities than the previous ones, used to actually sense the desired attributes in the field.

An example that shows the difference between cluster head nodes and participating sensors is when the former have GPS (Global Position System) location capabilities, while the latter only communicate directly to that CH, without the need of knowing their position in the field.

Because the sensors in the backbone constitute the communication path to the BS, the multi-path route to send the gathered data is easy to determine. On the other hand, the network lifetime is determined by the cluster heads' functioning time, which is directly related to cluster head activity and energy consumption. This can lead to the creation of "hot spots" [Rag1] or "black holes" in the network, generated by the energy depletion on one of the cluster heads (the CH uses its energy at a much higher rate and die much faster than the other cluster heads), which causes the isolation of the participating sensors attached to that CH. Managing the load becomes necessary in order to prevent the problem of premature battery drainage for particular cluster head nodes.

In homogeneous networks, all nodes have the same characteristics in terms of hardware and processing capabilities. An example of homogeneous network is when the sensors are deployed in battle fields. In this case, every sensor can become a CH. The cluster head role is usually periodically rotated among the nodes to balance the load, ensure that sensors consume energy more uniformly and try to avoid the hot spot (black hole) problem.

1.4. Communication Patterns in WSN

Two models are used for transmitting the data from the sensors to the base station: single-hop communication model and multi-hop communication model.

In single-hop communication, every sensor node can directly reach the destination. This limits the expansion of the network and forces the nodes to use a huge amount of their energy while transmitting their information to the base station. This energy consumption is more critical in the nodes that are farther from the BS.

In multi-hop communication, the nodes have limited transmission range and therefore are forced to route their data over several hops until the data reach the final destination. This increases the time needed to transmit the data from the sensor to the BS if there is a big number of hops between them. Also, this will increase the problem of the black hole,

because the nodes closer to the base station would have to spend more energy as they serve as intermediaries for all other sensors in the network.

The cluster-based communication scheme helps solve the previous problems. Once the WSN has been divided into clusters, the communication between nodes can be intra-cluster or inter-cluster. Intra-cluster communication comprises the message exchanges between the participating nodes and the CH. Inter-cluster communications includes the transmission of messages between the CHs or between the CH and the BS.

The fact that only the CH is transmitting information out of the cluster helps avoiding collisions between the sensors inside the cluster, because they don't have to share the communication channel with the nodes in other clusters. This also helps saving energy and avoiding the black hole problem. Latency is also reduced. Although the data has to hop from one cluster head to another, it covers larger distances than when the sensor are using a multi-hop communication model as the one used in Direct Diffusion [Ibr1]. The cluster-based communication model also facilitates the use of data aggregation models. In this case, only the CH performs data aggregation operations, helping the participating nodes inside the cluster to save energy.

Although cluster-based communication helps solving the problem of unbalanced energy dissipation among different nodes, it also has some drawbacks. First, the cost of creating the clusters can be energy-consuming. Second, as the distance between cluster levels increases, the energy spent is proportional to the square of the distance, which also

increase the energy expenditure [Ibr1]. However, the advantages of clustering overcome its disadvantages.

1.5. Static Clustering vs. Dynamic Clustering

The cluster architecture in wireless sensor networks can be created either in a static manner or in a dynamic manner.

Static clusters are usually created when the network is formed of heterogeneous nodes and the network designers want to create the clusters around the more powerful nodes. In this case, the clusters are formed at the time of network deployment. The attributes of each cluster, such as the size of a cluster, the CH, the number of participating sensors and the area it covers, are static.

Static clusters are easy to deploy, but their use is only appropriate for limited scenarios where the sensor field is predetermined, the targets to monitor are not in motion and it is easy to perform maintenance tasks (i.e. sensors replacements) in the network. If a CH consumes all its energy power, all the sensors in the cluster become useless. If the sensors inside the cluster (other than the CH) die, the CH also becomes useless because it doesn't have participating nodes to support.

Dynamic cluster architectures, on the other hand, make a better use of the sensors. Sensors do not statically belong to a cluster and may support different clusters at different

times. This communication scheme is generally used in WSN with homogeneous sensors, but can also be used in heterogeneous WSN.

The formation of a cluster can be triggered by using a special message sent to the cluster every certain period of time, or can be triggered by the occurrence of certain events (i.e. the detection of a big change in the monitored attributes). No explicit leader (CH) election is required at deployment time, which decreases the number of initial messages needed to create the WSN. However, a CH election method, a cluster formation method and cluster maintenance methods must exist in the network. Dynamic clustering in WSN is also more feasible when monitoring moving targets, due to the possibility of clusters reconfiguration.

1.6. Clustering Process

The use of clusters for transmitting data to the BS leverages the advantages of small transmit distances for most nodes, and having a smaller number of nodes (the CH) transmitting data to the base station. During the establishment of the cluster, it is necessary to take into account aspects like: cluster size and form, how to select the cluster head, how to control inter-cluster and intra-cluster collisions, and energy saving issues.

The design of the clustering process is one of the more important issues for the correct functioning of the WSN, due to the proved efficiency of using a hierarchical scheme for communications between the network elements. Many clustering algorithms and

protocols have been proposed for WSN. A clustering algorithm splits the network into disjoint sets of nodes each centering on a chosen cluster-head. A good clustering imposes a regular, high-level structure on the network [Cha1]. In general, a clustering algorithm should partition the network so that the nodes inside each cluster have high correlation. This correlation can be spatial, temporal or based on the data collected by the sensor.

A clustering algorithm also must ensure the proper localization of the sensor inside the cluster. This localization does not mean that the algorithm will change the physical location of the node, but that the nodes clustered together are close and in some cases, that the sensors are evenly spaced in order to maximize resource utilization and reduce errors. A clustering algorithm should have mechanisms to ensure that, after the initial formation of the clusters, it is possible to perform maintenance and reconfiguration functions to ensure the proper function of the WSN.

In all the cluster-based protocols we can identify three main phases during the clustering establishment process (Figure 1.3):

- Cluster head election
- Cluster formation (set-up phase)
- Data transmission phase (steady-state phase)

Figure 1.3: Clustering Process Phases

Different approaches exist to implement each one of these stages. For example, it is possible to use a fixed distribution of the SN and the CH, or to use a dynamic algorithm for the location of the sensors and the CH election. Depending on the scheme used to create the cluster and the special tasks assigned to the CH, the system can have different scalability, lifetime and energy efficiency levels.

As discussed in Chapter 2, it has been studied in many research projects and papers, that the hierarchical routing and specifically the clustering mechanisms make a huge improvement in WSN and help to reduce the energy consumption and the overhead created when all the sensor nodes in the network are sending information to the central data collection point or BS.

1.7. Importance of Routing

Once the sensors are organized into clusters and are assigned their specific role as CM or CH inside the cluster, routing mechanisms must be implemented to regulate the

transmission of messages towards the BS. As explained in Section 1.4, there are messages that involve intra-cluster communication between the CM and their correspondent CH, and inter-cluster communications among the CHs in the sensor field. The inter-cluster communications require the existence of formally defined routing protocols for WSN.

The responsibilities of a routing protocol in WSN, as in any other type of network, include functions like: sharing information about the paths in the network, search for a trusted path to the destination, using criteria like path length, number of hops in the route, minimum required transmission energy, maximum available power, etc., gathering information about the routes, minimize the bandwidth and energy consumption rates.

When working in wireless mobile environments, there are also other factors to consider, like (a) mobility of the nodes, which can cause broken routes, temporary cycles formation and collisions, (b) bandwidth restrictions, due to the number of nodes in the same areas or traffic conditions, (c) limited resources, because the routing operations can increase the rates of energy depletion in the nodes. Taken this into account, when designing a routing protocol for wireless environments, it is necessary to satisfy requirements like:

- a) distributed routing,
- b) minimum delay in the route discovery process,
- c) loop-free routes,
- d) minimum overhead,
- e) routes reconfiguration and
- f) scalability.

1.8. The Role of Mobility

Mobility in wireless sensor networks has attracted a lot of attention in the recent years and has introduced unique challenges in aspects like resource management, coverage, routing protocols, security, etc. Thus, the next evolutionary step for sensor networks is to handle the mobility effect in all its forms.

Most of the routing and clustering protocols currently used or proposed for WSN assume that the nodes are stationary [Arb1]. However, in applications like habitat monitoring (animal tracking) or search and rescue, that assumption makes those clustering mechanisms invalid, since the static nature of sensors is not real. One of the reasons for the sensors to be taken as stationary is because that assumption facilitates the simplification of the clustering protocols, making them have a very low overhead. It also avoids having to manage the mobility patterns of the sensors and allows saving more energy, since the localization information that the network has to manage is very low.

1.9. Cluster-based Routing Protocol for Mobile Sensor Networks

Objectives

In this thesis we propose a novel routing protocol which we call Cluster-based ROuting protocol for MObile Sensor networks (CROMOS). CROMOS considers the mobility of the nodes and its effect on the routing and communication processes among the sensors and between the sensors and the base station. Our main goal is to design a cluster-based

routing protocol for low mobility wireless sensor networks, where the clusters are formed based on the mobility patterns of sensors, but the information overhead caused by that mobility is low. The proposed protocol will help us to study how the mobility of the wireless sensors in the sensor field affects the transmission of information to the base station, and how the propagation schemes can be designed to obtain a longer life of the sensors resources, while accurately monitoring and reporting the positions of the sensors in the field.

To be able to analyze the repercussions of the sensor's movement, we consider in this thesis only low mobility sensors (with speeds between 1 and 10 m/s), because this will show more clearly the interaction of the nodes in the sensor field. The slow motion of the sensor will depend on the targets being observed in the fields, which would carry the sensors with them. The clustering and routing protocol also includes mechanisms to perform the re-routing processes and to avoid having to manage aspects like the broken routes caused by the movement of the sensors in the field.

1.10. Targeted Applications

Taking into account the previous consideration about the sensor's motion, we can suggest some example applications that could be suitable for using our protocol. These applications had been proposed either for field experiments, commercial products or research projects. We consider applications that will gain benefits from using our

protocol, due to the design facts achieved with a hierarchical organization of the sensors in the field.

1.10.1. Cattle Herding

Using wireless sensors to implement virtual fences, with an acoustic stimulus being given to animals that cross a virtual fence line is proposed in [But1]. The objective of this system is to reduce the overheads of installing and moving physical fences and to improve the usage of feedlots. In this case, according to the data collected by the researches the speeds of the cows are in a range of up to 1.25 m/s, depending on the activities being performed by the cows (grazing or walking). In the original proposal, the sensors located in the cows' collars form a multi-hop ad hoc network, forwarding movement data directly to a base station. We believe that with our proposal, it would be easy to control a group of cows that are close to the virtual fence, only by receiving the data from the sensor acting as a cluster head closer to those cows in the sensor field.

1.10.2. Turtle Tracking

Some protocols like "TurtleNet" [Tur1] aim to monitor the behavior of threatened turtle species, including terrestrial turtles and sea turtles. In the specific case of TurtleNet, "Wood Turtles" (*Clemmys Insculpta*) are being tracked in an effort of conservation biologists to protect this specie, which is rapidly declining due to habitat destruction and highway mortality. This kind of turtle is found throughout the Northeast and Great Lakes regions and into Canada. They live primarily in and along streams; however, they are also terrestrial for about 4 months of the year. An adult wood turtle could be able to cover a

distance of 140 m in 25 minutes (0.093 m/s). Taking into account this maximum speed, our protocol would be suitable to track turtle communities to get knowledge of their behavior during their terrestrial periods.

1.10.3. More Applications

Similar to the previous two applications, other research projects had been previously proposed for tracking animals, like Zebranet [Jua1] or Great Duck Island [Mai1]. However, in the first project, the routing protocol used to track the zebras is based on flooding information towards the base station, which considerably reduces the life of those sensors being used only as re-transmitters of information. In the second project, the sensors are not mobile, but located in specific places of the island to monitor the birds' movements when they get closer to them. Using our routing protocol, the systems proposed in applications like these, would be able to monitor animals using moving sensors, but at the same time, improving the energy consumption schemes used by the sensors, in order to make their batteries longer lasting.

The remainder of this thesis is organized as follows. In Chapter 2, we introduce related work of clustering and routing protocols for wireless sensor networks. We will review and describe in more detail different algorithms and protocols previously proposed for WSN. We classify them according to their fundamental operations or assumptions. We will briefly explain the principal characteristics the protocols use while executing their clustering and routing processes. In Chapter 3, our proposed "Cluster-Based Routing Protocol for Mobile Sensor Networks" is described. We will explain the characteristics

included in the protocol in order to manage the sensor's mobility. Chapter 4 shows the results obtained through simulation experiments. We evaluate the behavior of our cluster-based routing protocol for mobile sensor networks based on its design characteristics and the collected results. Finally, Chapter 5 presents conclusions and future work.

CHAPTER 2

RELATED WORK: CLUSTER-BASED PROTOCOLS

In this chapter, we present some well known clustering protocols and highlight their characteristics, related to the clustering phases identified in the previous chapter.

For each protocol we will present its main assumptions and characteristics, the steps involved in its clustering process and some additional aspects of its design.

2.1. Dynamic Clustering for Acoustic Target Tracking

The initial clustering schemas proposed for WSN were based in some sort of manual formation of the clusters, a consequence of the type of networks where the sensor were being used. In cases like the Dynamic Clustering for Acoustic Target Tracking [Che1], due to the manual location of the sensors, it is possible to create a heterogeneous setting of the network. The more capable nodes can be located in strategic places to allow them to act as cluster heads, and at the same time, lower capacity sensor can be placed around that CH to sense the data and send it to the BS using the backbone created. Mobility in these protocols is only considered regarding the approach of the targets to the stationary sensor nodes.

The Dynamic Clustering for Acoustic Target Tracking system [Che1] was proposed for acoustic tracking systems. In this specific application, the sensors are stationary and can be classified accordingly with their functions. They require two types of computation: (1) sensing the energy level of signals, which is not computational intensive and can be handled by sensors with minimal computation power; and (2) sound analysis, classification, and data fusion, which requires much higher computation power.

This is an example of a pseudo-dynamic clustering schema, where the high-capability nodes (in terms of available resources) are used as a static backbone, and the remaining sensors, with fewer resources, are clustered around them. The backbone sensors are placed sparsely in the sensor field and assume the role of a CH upon triggered by certain events of interest; and the task of the moderately to densely populated low-end sensors is to provide sensor information to the CH upon request.

This type of clustering protocol is used for reactive networks, since the CH becomes active in an on-demand fashion, when the acoustic signal strength detected by the CH exceeds a pre-determined threshold.

After becoming active, the CH broadcasts an information solicitation packet, asking sensors in its vicinity to join the cluster and provide their sensing information. After receiving a sufficient number of replies from sensors, the CH applies a localization method to estimate the location of the target and send a report to the subscribers

CHs and sensors construct several tables to facilitate determination of the back-off timer values (to be used when a CH intends to volunteer itself as a leader and when a sensor intends to respond to a CH).

2.2. LEACH-Based Protocols

LEACH (Low Energy Adaptive Clustering Hierarchy) [Hei2], was probably the first dynamic cluster head protocol proposed specifically for WSN using homogeneous stationary sensor nodes randomly deployed. In this section we review the main characteristics of LEACH and two additional protocols that either extend or modify the original operations proposed in LEACH. Figure 2.1 shows a summary of the LEACH-based protocols.

LEACH	<ul style="list-style-type: none"> • Low Energy Adaptive Clustering Hierarchy.
LEACH-C	<ul style="list-style-type: none"> • LEACH-Centralized • The base station has the responsibility of the cluster creation. • Nodes need to inform the base station about their location and signal strength.
LEACH-F	<ul style="list-style-type: none"> • LEACH-FIXED • Uses LEACH-C global clustering scheme. • Fixes the clusters once they are formed. • Reduces the overhead of cluster formation.

Figure 2.1: LEACH-Based Protocols

2.2.1. LEACH: Low Energy Adaptive Clustering Hierarchy

Probably, the first and the better known cluster-based routing protocol proposed was the Low Energy Adaptive Clustering Hierarchy (LEACH) [Hei2]. LEACH uses a distributed cluster formation algorithm, explained very clearly in [Hei1].

By using clustering techniques, the authors of leach demonstrated that they could achieve an eight (8) times better performance than the one obtained when using direct diffusion protocols. This protocol assumes that all the sensors: are stationary, have computational power to support different MAC protocols, and can transmit with enough power to reach the BS. It also assumes that nodes always have data to send, and this is the reason to use the protocol in proactive networks, instead of reactive networks.

In LEACH all nodes have a chance to become CH, to balance the energy spent per round by each sensor node. The CH for the cluster are selected randomly and in a rotary scheme based on their energy load. In every round 5% of the nodes in the WSN are CH, and all of them transmit data to the BS.

For the CH selection, each sensor node chooses a random number, r , between 0 and 1. The node calculates a threshold value, $T(n)$, that is going to be used to decide is it is a CH or not. Eq.(2-1). $T(n)$ incorporates the node's desired percentage to become a cluster head, the current round, and G that is the set of nodes that haven't been selected as CH in the last $(1/p)$ rounds.

$$T(n) = \begin{cases} \frac{p}{\left[1 - p \left(r \bmod \left(\frac{1}{p} \right) \right) \right]}, & \text{if } n \in G; \\ 0, & \text{otherwise} \end{cases}$$

Eq.(2-1) Threshold Value in LEACH

Figure 2.2: Flow-Graph of the Distributed Cluster Formation Algorithm for LEACH

After the CH has been selected, each CH broadcasts an advertisement message to the other nodes. See Figure 2.2. The nodes that receive the CH advertisement, decide which cluster they want to belong to, based on the advertisement's signal strength. The nodes inform to the selected CH that they are going to belong to its cluster. The number of

rounds N used for the CH selection is equal to the number of nodes in the cluster. The clusters are formed dynamically in each round and the times to perform the rounds are selected randomly.

The data collection in the cluster is centralized and is performed periodically. For this, once the clusters are formed, the CH creates a TDMA schedule giving the order in which the cluster members can transmit their data and broadcast it to the nodes. The sensor nodes send data to the CH according to the schedule. After completing the schedule, the CH fuses all the data and transmits it to the BS.

LEACH uses two times: T_F y T_R . T_F (Time Frame) is the total time required to complete the TDMS schedule. T_R (Report Time) is the period between successive reports sent by a node. Thus, T_F must be equal to T_R for each node. After a certain time, the network goes back into setup phase again and initiates a new cluster round.

LEACH uses TDMA/CDMA MAC as a complement to reduce inter-cluster and intra-cluster creation. Despite of the good performance of LEACH and its improvements over direct diffusion techniques in WSN, it has some drawbacks, i.e., it is possible that the elected CH will be concentrated in one part of the network and some nodes will not have any CH in their vicinity; LEACH clustering terminates in a constant number of iterations, but it does not guarantee good cluster head distribution and assumes uniform energy consumption for CH; and, the cost of the overhead to form the clusters is expensive

2.2.2. LEACH-C: LEACH-Centralized

LEACH-C (LEACH-Centralized) is a variation of LEACH, which uses a central-control algorithm to form the clusters. This may produce better clusters by dispersing the CH nodes throughout the network. The assumptions and characteristics for this protocol, Remain the same as in original LEACH.

In LEACH-C, the nodes send energy and location information to the BS at the end of the cluster change period. Based on this information the BS has a global network view and can optimally form clusters, appoint a fixed number of nodes as cluster heads, and evenly distribute the number of nodes in each cluster.

The BS computes the average node energy, and whichever nodes have energy below this average can not be CH for the current round. With the remaining sensor nodes, the BS runs a simulated annealing algorithm to select the CH; this is also used to select the associated sensor nodes that are going to belong to the cluster. Once the optimal CH and associated nodes are found, the BS broadcast a message that contains the CH-ID for each node. If a node's CH-ID matches its own ID, it becomes CH. Otherwise, the node determines its turn in the TDMA schedule and goes to the sleep state until that turn comes.

The cluster formation phase and the data transmission phase for LEACH-C remain the same as in original LEACH. LEACH-C requires each node to transmit information about

its location to the BS at the beginning of each round. Thus, the sensor nodes have to be location-aware.

2.2.3. LEACH-F: LEACH with Fixed Cluster and Rotating Cluster-Head

LEACH-F (Fixed Cluster, Rotating Cluster-Head) is also a variation of LEACH, which uses fixed clusters. The assumptions and characteristics for this protocol remain the same as in original LEACH, with one variation: LEACH-F does not handle node mobility.

The CH are selected using the same scheme as in LEACH-C, but after the first round is complete and the CH and associated sensor nodes are known, the clusters are fixed. The CH in the successive rounds is going to be selected taking into account the TDMA schedule, which is going to determine the order to rotate the CH position.

The cluster formation phase and the data transmission phase for LEACH-F remain the same as in original LEACH.

Due to the fixed clusters, LEACH-F will not be practical in any sort of dynamic system. This scheme does not allow new nodes to be added to the system and does not adjust its behavior based on nodes dying.

2.3. Protocols for Proactive Networks

One of the assumptions on LEACH-based protocols is that the sensors always have data to send. That is the reason for considering them all during the cluster formation. The same assumption is made for other protocols that combine the sensor nodes' "willingness" to send information, with the revision of specific attributes of special interest in the sensor field. Examples of this kind of protocols are: HEED[You1], EECS[Yem1], Sensor Aggregates[Fan1], ACE[Cha1], EEDC[Liu1] and TASC[Vir1].

Figure 2.3 shows a summary of the protocols for proactive networks.

HEED & EECS	<ul style="list-style-type: none"> • Hybrid Energy-Efficient Distributed Clustering & Energy-Efficient Clustering Schema. • Use the nodes' residual energy and power level measure to select the cluster heads and produces a uniform distribution of the cluster heads in the WSN.
Sensor Aggregates	<ul style="list-style-type: none"> • Three algorithms to create and manage the clusters: DAM (Distributed Aggregate Management), EBAM (Energy-Based Activity Monitoring) and EMLAM (Expectation-Maximization Like Activity Monitoring) • Nodes are grouped according to the collaborative processing task they perform.
ACE	<ul style="list-style-type: none"> • Algorithm for Cluster Establishment • The node that covers more nodes declares itself a CH and forms a new cluster.
EEDC	<ul style="list-style-type: none"> • Energy Efficient Data Collection • Clustering based on the overlaps of sensor's sensing ranges and on the analysis of surveillance data reported by the sensors.
TASC	<ul style="list-style-type: none"> • Topology Adaptive Spatial Clustering. • Locally isotropic, non-overlapping clusters.

Figure 2.3: Clustering Protocols for Proactive Networks

2.3.1. HEED: Hybrid Energy-Efficient Distributed Clustering

HEED (Hybrid Energy-Efficient Distributed Clustering) is a protocol that periodically selects cluster heads basing its decision on the node residual energy and a secondary parameter such as node proximity to its neighbors or node degree.

HEED preaches not to make any assumption about the nodes, but in the simulations presented in [You1], the sensor nodes are quasi-stationary, are location unaware, have similar capabilities, and equal significance. Each node has a fixed number of transmission power levels and the links between the sensor nodes and the BS are symmetric

The network serves multiple mobile/stationary observers, which implies that energy consumption is not uniform for all nodes.

Before a node starts executing HEED, it sets its probability of becoming a cluster head, CH_{prob} as: $CH_{prob} = C_{prob} * [E_{residual}/E_{max}]$, where $E_{residual}$ is the current residual energy in the node and E_{max} is a reference maximum energy which is typically identical for all the nodes. The cluster head selection is based on the residual energy of each node. This parameter is used to select the initial set of cluster heads. A node can elect to become a CH at consecutive cluster intervals if it has high residual energy and low cost. The intra-cluster communication cost (the second parameter) is used to separate the CH, depending of their range. The cluster range or radius is determined by the transmission power level used for intra-cluster announcements during clustering. This is the *cluster power level*.

The number of selected cluster heads varies according to the specified cluster radius. The smaller the radius, the larger the required number of CH to fully cover the entire network. The intra-cluster communication cost is a function of 1) cluster properties, such as cluster size, and 2) whether or not variable power levels are permissible for intra-cluster communication. A node joins the cluster head with minimum degree to distribute cluster head load, or joins the one with maximum degree to create dense clusters. The Nodes also automatically update their neighbor sets in multihop networks by periodically sending and receiving heartbeat messages.

HEED distribution of energy consumption extends the lifetime of all the nodes in the neighbor set, which adds to the stability of the neighbor set. Additionally, HEED guarantees a good cluster head distribution, due to the clustering termination in a constant number of iterations.

2.3.2. EECS: Energy Efficient Clustering Scheme

EECS (Energy Efficient Clustering Scheme) [Yem1] is a proposed energy efficient clustering scheme for periodical data gathering in WSN, introducing a method to balance the load among the cluster heads.

EECS assumes having N sensors uniformly dispersed within a square field. All sensors are stationary, location unaware, of equal significance and each one of them must be able

to compute the approximate distance based on the received signal strength if the transmission power is given.

The communication between the sensor nodes and the BS is symmetric, and based on the single hop scheme. The main idea is to pool the node's resources together to optimally direct those resources to task and cooperatively gather and process information while satisfying resource constraints.

In EECS, a constant number of candidate nodes are elected and compete for cluster heads according to the node residual energy. The BS sends HELLO messages to all nodes and the nodes use the power level of this signal's receiving strength to compute the distance to the BS. The SN broadcast its power level (residual energy) to the other nodes, using COMPETE_HEAD_MSG messages. Each sensor node compares its own power level with the other values received and if there is one greater than its own, it gives up trying to be the CH. If the power level is the greater, the SN becomes CH.

To form the cluster, each CH elected in the previous phase broadcasts HEAD_AD_MSG messages across the network. The sensor nodes receive the message and decide which cluster to join, using the distance metric and the received signal strength. EECS produces a uniform distribution of cluster heads across the network through localized communication with little overhead

The BS periodically broadcasts synchronization signals to all nodes. The energy expended during data transmission for far away cluster heads is significant, especially in large scale networks.

2.3.3. Sensor Aggregates

In [Fan1] the authors propose three different algorithms to create and manage the clusters: DAM (Distributed Aggregate Management), EBAM (Energy-Based Activity Monitoring) and EMLAM (Expectation-Maximization Like Activity Monitoring). They work using the concept of *sensor aggregates*. Their final goal is to abstract the collaboration patterns of the sensors into a set of generic schemas to support a wide class of applications for sensor networks. A sensor aggregate comprises those nodes in a network that satisfy a grouping predicate for a collaborative processing task. The parameters of the predicate depend on the task and its resource requirements.

This protocol assumes that all the sensors are stationary, have similar computational power, can be placed in arbitrary and non-uniform patterns, can communicate with other sensors within a fixed radius larger than the mean inter-node distance and monitor specific targets in a sensor field. The targets are point sources of signals, and can be stationary or can be moving at any time. It also assumes that nodes always have data to send, and this is the reason to use the protocol in proactive networks.

The DAM algorithm is used to elect the local cluster leaders. The cluster formation in general, includes the execution of the three algorithms. In this approach the cluster

formation depend on the peaks formed by their sensed signal strength. One peak may represent one target, multiple targets, or no target in case the peak is generated by noise sources.

For the CH selection, each sensor exchanges information with all its one-hop neighbors. Each sensor broadcasts its sensed signal strength at the start of each protocol period. This value is used as a “qualification parameter” to create the sensors clusters, assuring that there is only one peak per cluster. If after this exchange of information the sensor finds that its signal strength is higher than its neighbors’ signal strength, it elects itself as a CH or cluster leader.

The DAM algorithm uses a *thresholdElection* parameter that is used to determine if a node is eligible to become a CH. The minimum signal amplitude a sensor has to receive from targets for it to participate in the leader election process has to be greater than the *thresholdElection* parameter. This limits the packet exchange to the geographical area around targets.

Each sensor joins the cluster defined by the highest peak that can reach that sensor through a strictly descending path in the landscape. A tree structure is formed with each sensor pointing to one of its one-hop neighbors as its parent. After forming the clusters and electing the leader, this tracking algorithm assumes that the unique leader on each cluster knows the geographical region of the collaboration. The EBAM algorithm allows the estimation of the number of targets within each cluster. The EBAM algorithm

estimate the energy level at each node by computing the signal impact area, combining a weighted form of the detected target energy at each “impacted” sensor assuming that each target sensor has equal or constant energy level.

Although it is relatively straightforward to determine the peaks in the sensor field, it is possible that those peaks are related with the signal of more than one sensor at the same time, leading to the election of one of them as a CH, but leaving the signal from the other sensor unattended. To solve this problem, a “ceiling” attribute is included and used to define the amount of the power sensor contribution that a sensor can add to the total signal power value to slight changes of target locations.

EBAM uses the area of Voronoi cells associated with each sensor as weight in computing the total power for each cluster. As targets move closer to one another, by exchanging information among the sensor leaders, the number of targets can be tracked. But, a problem arises when targets move away from each other. It is difficult to determine how many targets go into one part of a split cluster and how many go into another. In this case, EMLAM comes into action.

EMLAM uses the Minimum Mean Square Estimation (MMSE) method to detect: when two targets move close enough to each other such that they share one cluster leader, by passing information from the two previous leaders, the new leader is able to know that two clusters are merging into one and is responsible for the joint estimation of the two

targets. EMLAM is for intra-cluster counting using expectation-maximization (EM) like technique.

During the inter-sensor communication, the packets with very low strength are dropped to minimize the amount of inter-sensor data exchange.

The data is sent every “protocol period” or time duration of a leader election process.

According to the simulations shown in the paper, the system works well in tracking multiple targets when the targets are not interfering, and it can recover from inter-target interference once targets move apart.

2.3.4. ACE: Algorithm for Cluster Establishment

In [Cha1], the authors present an algorithm called ACE (Algorithm for Cluster Establishment). ACE is an *emergent* algorithm. Emergent algorithms are a class of localized algorithms. Localized algorithms are characterized by reducing the amount of central coordination necessary and only require each node to interact with its local neighbors. The emerging localized algorithms have the additional characteristic that the individual agents (i.e. sensor nodes) only encode simple local behaviors and do not explicitly coordinate on the global scale.

A localized protocol for a sensor network is a protocol in which each sensor node only communicates with a small set of other sensor nodes within close proximity in order to

achieve a desired global objective. Thanks to locality, the chances for protocol failure due to transmission errors and node failure are reduced.

The goal of ACE is to select the smallest set of cluster heads such that all nodes in the network belong to a cluster. The problem is similar to the *minimum dominating set* problem in graph theory. Figure 2.4.

Figure 2.4: ACE (Algorithm for Cluster Establishment)

ACE results in a highly uniform cluster formation that can achieve a packing efficiency close to *hexagonal close-packing*, which minimizes the overlap between uniform circular clusters while ensuring full coverage. ACE is scale independent (it completes in constant time regardless of the size of the network) and operates without needing geographic

knowledge of node positions or any kind of distance or direction estimation between nodes.

In ACE, the sensors are considered stationary and with uniformly random coordinates in the sensor field. The proposed algorithm completes in constant time regardless of the size of the network, and consists of two logical parts: the first controls how clusters can spawn (by having a node elects itself to be leader) and the second controls how clusters “migrate” dynamically to reduce overlap. The sensors are all the time computing the clusters, without the necessity of a specific event. This makes the protocol suitable for proactive networks.

The nodes initiate actions at random intervals to avoid collisions. Each time that an action can be initiated for a node is called a node’s *iteration*. The communication between nodes is local and do not require time synchronization.

ACE distinguish between three possible node states: unclustered (still during the cluster formation process), follower (a node that belongs to a cluster with a designated CH) or cluster head.

The *iteration interval*, or time interval between iterations among nodes, is uniformly random distributed. All nodes in the sensor field are initially in the *unclustered* state. When a node is unclustered at the beginning of its iteration, it assesses its surroundings and count the number of *loyal followers* it would receive if it declared itself a CH of a

new cluster. A loyal follower is a node that can belong only to the cluster that would be formed by the current node sensing its signal.

If the number of loyal followers for the node is greater than or equal to its *spawning threshold* function, the node will span a new cluster. The *spawning threshold* function of a node is depending on the time that had passed since the protocol was initiated for that node. This *spawning threshold* ($f_{\min}$) is given by the formula: $f_{\min} = (e^{-k_1(t / cl)} - k_2) d$, which is an exponentially decreasing function and causes that at the beginning of the algorithm the nodes have less probability of becoming a cluster head, since the number of loyal followers is usually going to be less than $f_{\min}$, but the probability grows as the cluster formation process progresses. Based on the $f_{\min}$ function, the node will decide if it is becoming a cluster head or not.

If at the beginning of the iteration interval a CH already exists, this will send a POLL message to all its followers to determine which of the is the best candidate to become the new leader of the cluster. The best candidate to become CH is the node which has the greatest number of possible followers, while minimizing the amount of overlap with existing clusters. Once the best candidate is determined by the current cluster head, it will PROMOTE the best candidate as the new cluster head and ABDICATE its position as the old cluster head. In some cases, the best candidate can be the same node that was already acting as CH.

When a node becomes a CH, it conserves the previous ID if it was promoted by a previous CH or generates a random cluster ID if it declared itself as CH. After generating the cluster ID, the CH broadcasts a RECRUIT message to its neighbors, who will become *followers* of the new cluster. A node can be a follower of more than one cluster while the protocol is running (it picks a single cluster for membership only at the end of the protocol).

When the cluster head promotes a new node to CH the position of the cluster will appear to *migrate* in the direction of the new CH as some of the former followers of the old CH are no longer part of the cluster, while some new nodes near the new CH become new followers of the cluster.

Nodes that are not within one hop radius of any CH can pick a clustered neighbor to act as their bridge to the CH, becoming *two-hop followers*

The number of iterations that the algorithm should execute is a non fixed parameter that should be determined by the system using the algorithm, in order to determine a good trade-off between communication overhead and cluster size. This algorithm only covers the aspects related to the clusters formation, and does not include aspect related to data transmission after that.

According to the simulations presented in the paper, the authors determined that 3 iterations were good for its execution, but they do not warranty that number is going to be the correct one for all executions of the protocol.

2.3.5. EEDC: Energy Efficient Data Collection

EEDC (Energy Efficient Data Collection) [Yem1] proposes a *dynamic clustering and scheduling approach*, based on the overlaps of sensing ranges of sensors and on the analysis of the surveillance data reported by the sensors. Similar to CAG [Yoo1], EEDC takes advantage of the spatial correlation between the nodes, because usually the spatially correlated sensors may generate a lot of redundant data.

EEDC assumes a single-hop network architecture, where all the sensor nodes are within a single-hop radio transmission to the sink node, or to a local center. This clustering mechanism is useful for WSN where the nodes are stationary and use a proactive data transmit schema, because the clusters are formed based on the correlation between the data sensed by the nodes in the sensor field.

In each cluster, only one sensor node needs to work at any time instant. Therefore, that node is the CH in that instant. To decide which nodes are active in each cluster, the sink node (BS) executes a centralized clustering module to create the clusters and defines a schedule for the nodes to transmit.

To dynamically group sensor nodes into a set of disjoint clusters such that the sensor nodes within a single cluster have current strong spatial correlation, hence great similarity in observations, EEDC propose a greedy algorithm to heuristically find cliques that cover more vertices that have not been clustered. Heuristically, the vertices with larger degrees may have a better chance of appearing in larger cliques. Thus, the search starts from the vertex with the largest degree, until all vertices are covered. The output of this algorithm is a set of cliques that covers all vertices. The previous algorithm, dynamically partition the sensors into groups so that the sensors in the same group have similar surveillance time series. Within each cluster, the pair wise dissimilarity measure of the sensor nodes is below a given intra-cluster dissimilarity threshold, ensuring that all sensor nodes in the same cluster are correlated.

To minimize the number of clusters and therefore maximize the energy saving, EEDC models the cluster creation process as a clique-covering problem. A graph G is constructed, such that each sensor node is a vertex in the graph. An edge (u, v) is drawn if the dissimilarity measure between vertex u and vertex v is less than or equal to the given intra-cluster dissimilarity measure threshold. A cluster is a clique in the graph. Then, the clustering problem is to use the minimum number of cliques to cover all vertices in the graph.

The sink dynamically adjusts the clusters based on spatial correlation and the received data from the sensors. The number of clusters will keep increasing, since there is only splitting operation in the protocol, but the authors claim that: “(a) spatial correlation is

quite stable for some applications even if the monitored phenomenon changes dramatically and (b) the sink node can re-cluster the whole network when the current number of clusters becomes significantly larger than the number of clusters at the previous network-wide clustering” [Yem1].

The paper presented in [Int1], does not explain how the sink creates the schedule for the node’s transmissions. It only states that the node’s transmissions are done in a round robin fashion into each cluster.

2.3.6. TASC: Topology Adaptive Spatial Clustering

The Topology Adaptive Spatial Clustering (TASC) algorithm presented in [Vir1] is a distributed algorithm that partitions the network into a set of locally isotropic, non-overlapping clusters without prior knowledge of the number of clusters, cluster size and node coordinates. Targeting to the clusters that are formed *with respect to existing network topology* is the issue that makes TASC different than other sensor network clustering algorithms.

The spatial grouping of nodes with respect to regions of close proximity and similar deployment density: (a) promotes efficient data aggregation, (b) allows to manage different data compression rates in each cluster, (3) improves the overall compression rate in the whole network and (d) as a consequence, it also can help reducing the propagation of redundant data inside the network. Spatial clustering would also assist

transmission power control, since intra-cluster communication requires less transmission power in dense clusters.

TASC is useful for proactive networks, where the clusters provide the information needed for the cluster formation process. The nodes in the sensor field are assumed to be stationary. Each node can measure distances to its one hop neighbors and has knowledge of its 2-hop neighborhood.

TASC operates on combination of node weights and a dynamic density reachability criterion. The main objective of TASC is to cluster non-uniform sensor networks. In this type of network, the node density variations are *globally big* but there exist subgroups of nodes such that density variations are *locally small*. The idea of TASC is partition those networks in a way such that relative node density variation in individual clusters is smaller than relative node density variation in the whole network.

For the cluster formation, two different parameters must be previously specified: required minimum cluster size and density reachability parameter. The latter parameter allows each node to further limit the number of nodes that it can potentially nominate as CH by considering only density reachable nodes as nomination candidates. This effect pulls cluster leaders towards most dense groups in the cluster, but nomination among density reachable candidates is still based on weights. Figure 2.5.

Figure 2.5: Clusters in TASC

Initially, each node in the WSN computes its own weight based on shortest Euclidean paths in its 2-hop environment; in a non-uniform deployment WSN, the node that tends to be the midmost related to all shortest communication paths (in terms of hops) gets the biggest weight. Instead of incrementing the weight by one each time a node is used in a path, we increment the weight as a function of the distance a node contributes to the path.

The weight value calculated by the node is broadcasted to its 2-hop neighborhood. Since this is done by all nodes, each node also receives the weights of its 2-hop neighbors. By comparing the weights, each node nominates the node having biggest weight in the density-reachable subset of its 2-hop neighbors and broadcasts its nominee to its 2-hop neighborhood. After receiving all nominees in its 2-hop neighborhood, each node elects the closest nominee to its leader. Each node that ends up in a cluster where the total

number of nodes is smaller than pre-specified minimum cluster size joins to closest cluster, where the number of nodes exceeds the required minimum cluster size.

After creating the clusters, TASC uses an “*all-pairs-shortest path routing*”. Instead of trying to forward traffic to the neighboring node that is closest to the destination, TASC routing is based on distance measurements to extract information about the network topology. More specifically, node weight is a measurement of two key quantities 1) the frequency a node is found on the shortest path between pairs of nodes and 2) the distance contribution of the edges of that node with respect to the total length of the path.

For some systems where incremental deployment makes sense the leader election mechanisms need to be adapted to support the addition and subtraction of nodes from the network.

2.4. Protocols for Reactive Networks

Contrary to the protocols presented in the previous section, this new group of protocols usually take advantage of the queries performed by the user about the sensed data or of a specific triggering event occurred in the WSN. In this case the main idea is to save energy by aggregating information at the same time that a data transmission path is created to supply the request for information. Protocols that follow this pattern include: TEEN[Man1], APTEEN[Man2], CAG[Yoo1] and Upgraded CAG [Yoo2]. Figure 2.6 shows a summary of the protocols for reactive networks.

TEEN	<ul style="list-style-type: none"> • Threshold sensitive Energy Efficient sensor Network protocol. • All nodes take turns to become the cluster head using a cluster period parameter. • Cluster formation very similar to LEACH.
APTEEN	<ul style="list-style-type: none"> • Adaptive Periodic TEEN • Uses a combination of proactive and reactive network's features. • Uses node pairs for query responses.
CAG	<ul style="list-style-type: none"> • Clustered AGgregation technique. • Cluster are formed using spatial correlation of the nodes. • Two phases: Query phase (forms clusters), Response phase (sends the data)
Upgraded CAG	<ul style="list-style-type: none"> • Clusters remain unchanged as long as the sensed values stay between a given threshold. • Add temporal correlation to the CAG clustering technique.

Figure 2.6: Clustering Protocols for Reactive Networks

2.4.1. TEEN: Threshold-sensitive Energy Efficient Sensor Network

Threshold-sensitive Energy Efficient Sensor Network (TEEN) protocol [Man1] is a clustering protocol designed to be used in reactive networks, where the nodes react immediately to sudden and drastic changes in the value of a sensed attribute. This approach is useful for time-critical applications, but not well suited for applications where the user needs to get data on a regular basis

In TEEN, the sensor nodes are considered to be stationary, homogeneous and beginning with the same initial energy, while the BS is considered to have constant power supply and no energy constraints, and located far away from the nodes. The communication between the BS and the SN is asymmetric

In this protocol, all nodes take turns becoming the CH for a time interval T , called the Cluster Period (Figure 2.7). Two new parameters are used for each sensed attribute during the cluster functioning: the *Hard Threshold* and the *Soft Threshold*. The *Hard Threshold* value defines the range of interest in which the attribute is important for the network. The *Soft Threshold* is a small change in the value of the sensed attribute

Figure 2.7: Time Line for TEEN

TEEN main focus is in the information aggregation, thus the cluster formation is very similar to LEACH. Sensor nodes sense the medium continuously, but the data transmission is done less frequently. When the cluster heads are to change, new values for the parameters *Hard Threshold* and the *Soft Threshold* are broadcasted.

While monitoring the environment, each node senses a set of attributes. If the value of the sensed attribute is greater than the *Hard Threshold*, the node stores the sensed value and reduces the number of transmissions. If the sensed value is greater than the *Hard Threshold* plus the *Soft Threshold*, the node is triggered and sends data to the CH.

The *Hard Threshold* tries to reduce the number of transmissions by allowing the nodes to transmit only when the sensed attribute is in the range of interest, while the *Soft Threshold* reduces the number of transmissions. A smaller value of the soft threshold gives a more accurate picture of the network, at the expense of increased energy consumption, but by managing carefully the thresholds values, the user can control the trade-off between efficiency and data accuracy.

TEEN needs to ensure that there are no collisions in the clusters. Since message transmission consumes more energy than data sensing, this scheme consume less energy than the one used in the proactive networks. However, it has an important drawback: if the thresholds are not reached, the nodes will never communicate.

2.4.2. APTEEN: Adaptive Periodic Threshold-sensitive Energy Efficient sensor Networks protocol

APTEEN (Adaptive Periodic Threshold-sensitive Energy Efficient sensor Networks protocol) is a variation of TEEN, designed as a hybrid protocol that changes the periodicity or threshold values used, to provide a periodic state view of the network.

APTEEN uses a combination of proactive and reactive network's features, and specially focuses on how to manage the data collected in the sensor nodes.

All nodes in the network are assumed to be homogeneous and to begin with the same initial energy, while the BS provides a direct path for the down link.

The cluster head selection in APTEEN is based on the mechanism used in LEACH-C. The cluster exist for an interval called the cluster period T , and then the BS re-groups clusters, at a time called the cluster change time

Once the cluster heads are decided, in each cluster period, the cluster head first broadcast the following parameters: attributes to be sensed, thresholds (hard and soft) for those attributes, transmission schedule (TDMA), and count time that is the maximum time period between two successive reports sent by a node.

APTEEN uses a modified TDMA schedule (Figure 2.8), where each node in the cluster is assigned a transmission slot, to avoid collisions. In this schedule, the nodes have two states: sleep and idle (listening for queries), the modification is that the slots for the idle nodes have to be larger than the slots for the sleeping nodes.

Figure 2.8: Time Line for APTEEN

For query responses, APTEEN uses node pairs. This implies adjacent nodes that sense similar data, but only one of them responds to a query; the other can go to “sleep” mode

and don't need to receive the query. These two nodes can take the role of handling queries alternately.

The CH aggregates all the data and sends it to its higher level CH or the BS. The BS extracts the queries and the answers and transmits them in down-link mode. However, the BS shouldn't transmit to the nodes when they are transmitting data to their CH in their slots. Thus, it is necessary to include the BS in the schedule and the BS needs to be assigned a separate slot. The BS has to calculate the length of the longest TDMA schedule among the clusters and make allowance for the transmitted data from the CH to reach it, after which it can transmit its own data. Different CDMA codes are used to avoid inter-cluster collision. A common CDMA code is employed for the up-link from the CH to the BS and the downlink from the BS to all sensor nodes.

The main drawbacks of the APTEEN approach are the overhead and complexity associated with forming clusters at multiple levels, the method of implementing threshold functions, and how to deal with attribute-based naming of queries.

2.4.3. CAG: Clustered AGgregation technique

While monitoring environmental features in a sensor field, nearby sensor nodes typically register similar values. This relation between position and measures is called *spatial correlation*. According to its authors, CAG (Clustered AGgregation technique) [Yoo1] is the first in-network aggregation algorithm exploiting spatial correlation, which trades a

negligible quality of result (precision) for a significant energy saving. CAG forms clusters of nodes sensing similar values (real-time sensor values).

CAG is appropriate for reactive WSN, because it acts when the user specifies a query for monitoring a specific attribute. The nodes in the sensor field are assumed to be stationary and positioned in a two dimensional grid, and all nodes are within a radio range R performing lossless communication. During their simulations, the authors assume that every single node participates only in one cluster, not in multiple clusters.

The CAG algorithm operates in two phases: query and response. During the query phase, CAG forms clusters. During the response phase, CAG transmits the value of the aggregated data within the cluster to the BS.

At the beginning of the algorithm, a *user provided error threshold*, τ parameter is introduced in the network. This parameter is used along with the reading of each node, to decide if that node must become a CH or not.

Cluster head selection policy applied in CAG is such that the first node becomes a CH if its sensor reading is outside of the tolerable error range for CR(Cluster head Reading). If the reading (sensed data) of a node is bigger than the CR (initially zero) plus the user provided error threshold, it becomes a CH, and thus, a parent for other nodes in the tree structure formed by the algorithm.

As said before, the clusters are built using a *user provided error threshold*, τ . Each node decides to join a cluster based on the cluster head sensor reading (CR) and its own reading (MR). If $MR < CR \pm CR \times \tau$, then the sensor is included in the same cluster. After the user specifies the attribute that want to monitor, the base station broadcast the query through the tree that is being formed at the same time. The clusters are formed when the forwarding tree is built. CAG forwarding tree is based on dynamic environmental phenomena, so the CH may change for every query and response cycle. This prevents cluster heads from becoming energy-draining bottlenecks.

Once all the nodes receive the query packet, the response phase starts, transmitting the gathered data through the previously built tree. Bridge nodes (intermediate nodes) do not contribute their sensor readings to the aggregate, but they are required to bridge the segments of the forwarding tree.

CAG achieves efficient in-network storage and processing by allowing a unified mechanism between query routing (networking) and query processing (application). Instead of gathering and compressing all the data (lossless algorithm), CAG generates synopsis by filtering out insignificant elements in data streams (lossy algorithm) to minimize response time, storage, computation, and communication costs. CAG is a lossy clustering algorithm because it uses only sensor values from the cluster heads to compute the aggregates.

According to the simulation showed in [Yoo2], as data becomes more correlated and the user given threshold gets larger, fewer transmissions are observed. As data becomes more correlated, lower relative error is observed. The relative error is guaranteed to be within the user provided threshold, when the data shows a certain level of special correlation.

Although the overhead and complexity of forming clusters and implementing threshold-based functionality are drawbacks of CAG, those are negligible because cluster is built during query forwarding; a step common to any query/response protocol in wireless sensor networks.

2.4.4. Upgraded CAG

Updated CAG algorithm [Yoo2] is an improvement to the CAG algorithm presented in [Yoo1], where the clusters are still formed from nodes sensing similar values within a given threshold (spatial correlation), but in this case, the clusters remain unchanged as long as the sensor values stay within a given threshold over time (temporal correlation). This fixed range clustering ensures that the performance of CAG become independent of the magnitude of sensor readings and network topology.

In this CAG extension, the authors included two modes of operations: interactive and streaming, to enable CAG to be used in different environments and for different purposes.

In the interactive mode the same weight is given to different clusters independent of their size. The interactive mode alternates query and response phases. This is appropriate for scenarios where the environment (network topology and data) changes dynamically, or the users change the approximation granularity or query attributes over time. In this scenario, a new forwarding tree is built each time a query is sent out. This frequently rebuilding the tree can be wasteful if the sensed data is almost the same over time. If data is unchanged, the cluster head nodes and the forwarding tree are likely to be the same.

The streaming mode adjusts the clusters locally as the data and topology change over time. In this case, the CH keeps transmitting streams of response for a query that is issued just once, thus the tree does not have to be reconstructed and the network changes its schema from reactive to proactive. This mode of operation is appropriate when the environment does not change as frequently and the query remains valid for a certain period of time. A node counting is performed within a cluster to assign weights to the CH values. The count is updated only when cluster adjustments happen. With the streaming mode, each member of the cluster sends a packet to the cluster head so that the CH can calculate the size of the cluster.

In streaming mode, due to the CH are fixed throughout the duration of the query, they can become an energy bottleneck. In this case, the authors propose the use of CH rotation technique to maximize the network lifetime, like the mechanism used in LEACH.

2.5. Summary

Several clustering approaches have been proposed for mobile sensor networks, due to the advantages of having a hierarchical structure to communicate the nodes and save energy in the WSN. Table 2.1 shows a comparison between the main characteristics on the mentioned protocols.

Table 2.1: Clustering Protocols for WSN

	Network Type		Sensors Mobility		Sensors Type		Clustering Process		Cluster Head Election			Clustering Formation		
	Proactive	Reactive	Stationary	Mobile	Homogeneous	Heterogeneous	Static	Dynamic	Fixed	Random	Attribute Based	Centralized	Distributed	Fixed
DCATT		Y	Y			P		P	Y			P		
LEACH	Y		Y	P	Y			Y		Y			Y	
LEACH-C	Y		Y	P	Y			Y		Y		Y		
LEACH-F	Y		Y		Y			Y		Y		Y		Y
TEEN		Y	Y	P	Y			Y			Y		Y	
APTEEN		Y	Y	P	Y			Y			Y		Y	
HEED	Y		P	P	Y			Y			Y		Y	
EECS	Y		Y		Y			Y			Y		Y	
Sensor Aggregates	Y		Y		Y			Y			Y		Y	
ACE	Y		Y		Y			Y			Y		Y	
CAG		Y	Y		Y			Y			Y		Y	
Upgraded CAG		Y	Y		Y			Y			Y		Y	
EEDC	Y		Y		Y			Y			Y	Y		
TASC	Y		Y		Y			Y			Y		Y	

Y
Yes
P
Partially

Reviewing the protocols presented in this chapter is possible to observe that all of them are concerned on how to prolong the WSN lifetime and how to make a more efficient use of the critical resources located at the sensor nodes, without decreasing the communication functionalities, but creating more intelligent clusters, minimizing the maximum number of nodes in a cluster, and minimizing clusters with only a single node (the CH).

The sensors mobility is only considered partially in some of the protocols reviewed in this chapter. The next evolutionary step for sensor networks is to handle the mobility

effect in all its forms. The cluster-based routing protocol that we propose in this thesis aims to cover all those mobility effects, using zone-base information and a cluster-like communication between nodes in the sensor field.

CHAPTER 3

CLUSTER-BASED ROUTING PROTOCOL FOR WIRELESS SENSOR NETWORKS

In this chapter, we describe the Cluster-Based Routing Protocol for Mobile Sensor Networks (CROMOS). First we give an overview of the protocol in Section 3.1. In Sections 3.2 to 3.7, we describe the Zones Definition, Cluster Head Election and Zone-Based Routing procedures, detailing the aspects of the protocol that allow managing the sensor's mobility. In Section 3.8, we present a summary of CROMOS protocol.

3.1. Cluster-Based Routing Protocol Overview

In this section, we present the assumptions we made related to the sensors involved in the functionality of our protocol and the main components that constitute the design of CROMOS.

3.1.1. Protocol Assumptions

In our proposed protocol, we make different assumptions to facilitate the design of the clustering mechanism and the routing protocol.

Assumption 1: All sensors in the network are homogenous.

The sensors are considered to have equal physical characteristics and the same available energy and transmission range at network deployment time. All sensors initially have the same resources when the network is deployed.

Assumption 2: All sensors in the network are location-aware.

To know their location in the network, the sensors can use a Global Position System (GPS) if we are working with “high-level resources” devices, or the sensors can use GPS-free location scheme (i.e. location based on triangulation) if we are working with “low-level resources” devices. More information about location schemes used in WSN can be found in [Sav1] [Ngu1] [Mos1] [Til1] [Nic1] [Bar1].

Assumption 3: The base station is stationary.

Some recent proposals related to mobility in WSN, like [Luo1] consider the option of a moving base station. However in this thesis we are not considering this option and we assume that the base station is stationary. Including a moving BS is an option that we plan to consider in future work.

Assumption 4: The nodes are mobile.

Node mobility is a very important aspect in the design of an effective routing algorithm for mobile wireless networks. For this reason, the focus of many recent researches has been put into mobility characterization. Research works like [Vya1] and [Sam1] study how it is reasonable to expect a high degree of correlation between future and past movement speeds and direction. Thus, while each node acts separately from all other

entities, its decisions for the next move are somehow affected by the context it is in and by the movements surrounding it. That dependence factor could give some amount of predictability into the next movement which the node will make. The basic consideration to raise the mobility patterns is that in each position, the mobile node is associated with a direction vector. This direction vector may be different for each position, but when the node moves, it is the deviation of the newly acquired direction vector from the old one which dictates the type of mobility pattern being followed by the nodes.

In general, there are three different mobility models to consider: random, semi-deterministic and deterministic. In the *random movement* the mobility is totally unpredictable and each new movement can be completely independent of the previous movement. In the *semi-deterministic movement* even though the individual nodes do not have a specified direction, it is possible to see a general pattern of a column evolving to it, in what is called a column model. In the *deterministic movement* the mobile nodes mostly move in the same direction and then the deviation of the direction vectors associated with any two positions would be zero.

CROMOS uses a *random movement model* based on (a) the selection of a random location inside the sensors field for each sensor to move to, (b) straight movement towards the selected location, and (c) a velocity that is uniformly distributed between 0 and a configurable maximum speed. Using this model for the protocol facilitates its adaptability for different kinds of wireless sensor applications.

CROMOS uses a parameter called the *Maximum Moving Velocity (MMV)* to determine the speed of each one of the sensors in the field and to calculate the **Mobility Factor**, which is a critical parameter for the protocol and acts as an indicator of a node's mobility. We will define and explain the Mobility Factor in Section 3.5.1.

3.1.2. Cluster-Based Routing Protocol for Mobile Sensor Networks Stages

CROMOS has three different stages: zones definition, cluster head election and zone-based routing. The *Zones Definition stage* is used to divide the sensor field into areas of the same size and to determine the zone where the sensor is currently located. The *Cluster Head Election stage* is used to determine which node is going to act as a cluster head to forward messages towards the base station. The *Zone-Based Routing stage* is used to discover a route and send data packets following the discovered route.

3.1.2.1. Zones Definition Stage

When considering how to divide the sensor field, we can take into consideration many mechanisms, most of them related with geometrical patterns that facilitate the definitions of aspects like boundaries between areas, coverage, positioning of sensors, etc. In [Che1], the authors use hexagonal areas based on the Voronoi cells proposed in [Aur1] and very similar to the structure used in cellular networks. In some other cases, a circular shape associated with the radius of the transmission signal is used to define the areas in the network. In our case, we use a **square shape**. We assume that regardless of the real shape of the monitored area, it is always going to be possible to circumscribe that area

into a big square (Figure 3.1) and then divide that square field into non-overlapping square zones.

Figure 3.1: Different Sensor Fields' Shapes Circumscribed Into Squares

The CROMOS routing protocol is based on the formation of non-overlapping square zones, all of them with the same size and a unique Zone ID.

The square zones are going to have the same size. Every area corresponds to a cluster region. In this thesis, we are going to use the term zone and cluster interchangeably, because they correspond to the same area of the sensor field. Each zone has a unique Zone ID (i, j) , corresponding to the coordinates of the origin point of the area that it covers in the plane (Figure 3.2).

Figure 3.2: Zone IDs

Each node is placed in a zone and can obtain its zone ID based on its location parameters. The node location can be obtained using GPS or a GPS-free scheme, depending on the node characteristics and capabilities.

3.1.2.2. Cluster Head Election Stage

The sensors in a common zone form a cluster and each cluster has one of the mobile nodes acting as cluster head. The function of the cluster head election stage in CROMOS is to select which node is the best to act as a cluster head during each period of the network lifetime. This selection is based on the mobility of the sensors, as we will explain in Section 3.5.

The cluster head acts as a gateway between the sensors in the cluster and the base station or, between the sensors in the cluster and other cluster heads.

The responsibilities of the cluster head include:

- Maintaining information about the member nodes in the zone,
- Receiving messages from the member nodes,
- Sending vicinity messages to its neighbors to create the routes to the base station,
- Forwarding data packets to its next cluster head hop along the path or to the base station,
- Handing-off the information to the new zone head when it moves out of the zone or when it demotes its role to member.

We use a mobility factor as an indication of the node's mobility. This is the average number of times a node moves from one zone to another during a specific period of time (See Section 3.5.1). The node with the smallest mobility factor becomes the next cluster head.

3.1.2.3. Zone-Based Routing Stage

CROMOS acts as a hybrid routing protocol. Inside every zone the communication between the nodes and the cluster head is proactive and is made directly to the cluster head that is located at a one-hop distance from the node. The communication among the zone heads and between the cluster heads and the base station is made in a reactive manner, based on a periodic timer that triggers the sending process on the cluster head or based on the amount of data the cluster head has to transmit.

In CROMOS, only the cluster heads act as routers for the network, which helps avoid the *broadcast storm effect problem* [NiS1] caused by using general flooding techniques to route messages in wireless networks.

CROMOS uses a vicinity advertisement process that allows the cluster heads to maintain information about their neighboring cluster heads and helps identify which of those cluster heads are closer to the base station than the current zone head. The cluster head uses this information to select the best next hop to forward the accumulated data to the base station. Once the cluster head forwards the data packets to the next cluster head and receives its acknowledgement for the reception of the packets, it assumes that the packets are responsibility of the receiving cluster head and clears the packets from its data cache in order to have more space for new data packets.

When a packet is to be sent from one node to another and if there is no current route available in its cache table, the source node stores the packet in its data cache until the next cluster head election and neighbor discovery processes are executed. Detailed explanations of these mechanisms are given in the following sections.

3.2. System Parameters

The following inputs are necessary to initiate the CROMOS protocol:

- a) **Origin:** the original position of the deployment area. This is always assumed to be (0, 0) which is the location of the base station in the sensor field.

- b) **Field width and height:** The width and height of the sensor field where the sensors will be deployed.
- c) **Maximum Transmission Range (MTR):** The maximum transmission range for each sensor in the field. All the sensors are considered to be uniform (i.e. have the same MTR).
- d) **Maximum Moving Velocity (MMV):** The maximum moving velocity of the sensors in the field.
- e) **Number of Sensors:** The number of sensors to be deployed in the field.

Every mobile sensor node maintains the following information:

- a) **Sensor ID:** Unique identification number of the sensor in the network.
- b) **Location:** Sensor's position (x, y) received from the GPS receiver
- c) **Zone ID:** Identification of the zone where the sensor is currently located.
- d) **Zone Head:** Reference to the sensor acting as a cluster head in the zone where the sensor is currently located.
- e) **Status:** Indicates the current sensor's status. The possible values for this attribute are: ACTIVE (Sending or Receiving Data) or DEAD.
- f) **Role:** indicates if the sensor is acting as a member or as a cluster head inside the zone. The possible values for this attribute are: MEMBER, CLUSTERHEAD, or POSSIBLECH (i.e. Possible Cluster Head).
- g) **Mobility Module:** this sensor's sub-module maintains the location, speed and mobility pattern of the node. It operates using a random waypoint where a mobile sensor node starts from its original location, randomly chooses a location in the

simulated area as the destination and moves in a straight line to that destination location at the speed that is uniformly distributed between 0 and MMV. When the mobile sensor node reaches the destination location, it chooses a new destination location and repeats the procedure again.

- h) **Sensing Module:** The role of this sensor's sub-module is to receive (sense) the stimulus (signal) generated by an external source to send it to the base station. The packets carrying the sensed data are considered to have a constant bit rate.
- i) **Battery:** Implements a simplistic battery model whose initial capacity is assumed to be always constant and decreases according to the energy consumed during the basic operations of the sensor node which basically involve sending and receiving messages.
- j) **Location Changes:** The number of times the location information of the node is updated.
- k) **Zone Changes:** The number of times the node moves from one zone to a new zone.
- l) **Mobility Factor:** calculated as the percentage of position changes that correspond to zone changes over the total location changes of the node.
- m) **Times CH:** Counter that indicates the number of times the sensor has acted as cluster head.

If the sensor is acting as a cluster head, it also maintains the following information:

- a) **Sensors In Zone:** A list of the sensors that currently are in the same zone and using this sensor as their cluster head.

- b) **Data To Transmit:** Collection of stored data sensed by the member nodes in the same zone and pending to transmit.
- c) **Neighboring CHs:** A list with references to the sensors that are acting as zone heads in the neighboring zones of this cluster head's zone.
- d) **Level:** Indicates how far the cluster head is from the base station. It defines the number of hops that a data packet will need to follow to get to the base station from the current CH.

Each sensor is deployed randomly in the field and has an initial position and a random destination that it is going to move to. Once the node reaches its destination, it determines a new destination to go to and starts the movement process again. The node moves at a constant speed which is uniformly distributed between 0 and the MMV.

3.2.1. Network Initialization

In the initialization stage, every node sets its parameters as shown in Table 3.1:

Table 3.1: Initialization Parameters

Sensor ID	A unique number from 1 to the maximum number of sensors in the sensor field.
Location	Random location between (0,0) and (Field Width, Field Height)
Zone ID	Calculated based on the sensor's current location (see Figure 3.3)
Zone Head	Reference to the sensor node that is acting as CH in the sensor's current zone. Initially, each sensor references to itself as its own cluster head until the cluster head election period is started.
Status	Set to ACTIVE. When the sensor consumes its energy, its status is changed to DEAD.
Role	Initially all sensors assume a MEMBER role. During the Cluster Head Election Process, the sensor node determines if it will act as a CH or as a CM until the next Update Time period.
Mobility Model	The sub-module is initiated and selects an initial random destination for the node. (See Figure 3.4)
Sensing Model	The sub-module is initiated and randomly defines when the sensor is sensing information and when is it not. The packets generated by the sensing sub-module have a constant bit rate of 100 bytes, with a 20 bytes header and an 80 bytes data payload.
Battery	Each node has an initial energy of 0.5 Joules.
Location Changes, Zone Changes, Mobility Factor, Times CH	Initially set to 0.
Sensor In Zone, Data To Transmit, Neighboring CHs, Routes Cache	Initially set to NULL.

The sensor's zone is calculated as shown in Figure 3.3 using the sensor's current location and is verified against the zones map that the system maintains to control the areas in the sensor field.

```

/*
 * Returns the Zone ID of the sensor's zone, depending on the position
 * where the sensor is located
 */
Locate_Zone(Zone_Size, zones_Map){

    /* Obtain coordinates of the zone where the sensor is located */
    Zone_X_Coordinate =
        Math.abs((Node_X_Coordinate-Sensor_Field-Origin)/zoneSize);
    Zone_Y_Coordinate =
        Math.abs((Node_Y_Coordinate-Sensor_Field-Origin)/zoneSize);

    /* Verify zone's ID in the network's Zones Map */
    Zone = Zones_Map.Locate_Zone_at(X,Y);

    if (zone found in Zone Map){
        return Zone;
    }else{
        return null;
    }
}

```

Figure 3.3: Locate Zone Method

The sensor's mobility is based on a series of objective destinations randomly selected inside the sensor field. Figure 3.4.


```

/*
 * Defines a Random destination for the node. This destination is
 * calculated based on the current position of the node and the max
 * value for the generated X and Y coordinates for the destination
 * point.
 */
Set_Random_Destination(){

    currentSpeed = Generate_Random_Speed(max_Speed);

    new_X_Position_In_Field =
    min_Field_X + Generate_Random_Distance_To_Move;
    new_Y_Position_In_Field =
    min_Field_Y + Generate_Random_Distance_To_Move;

    dest_Pos_X = new_X_Position_In_Field - cur_Pos_X;
    dest_Pos_Y = new_Y_Position_In_Field - cur_Pos_Y;

    distance_To_Move = Math.sqrt((dest_Pos_X * dest_Pos_X) + (
    dest_Pos_Y * dest_Pos_Y));

    if (new_X_Position_In_Field != cur_Pos_X) OR
    (new_Y_Position_In_Field != cur_Pos_Y) {
        dest_Pos_X = dest_Pos_X/distance_To_Move * current_Speed;
        dest_Pos_Y = dest_Pos_Y/distance_To_move * current_Speed;
    }
}

```

Figure 3.4: Definition of the Sensor's Next Target Location

3.3. Zones Creation Process

In order to calculate the zones, the system uses the following parameters: origin, MTR and MMV.

The zone size is defined using the sensor's MTR, which is used as the diagonal L of the square zone that can be covered using the sensor's transmission range (Figure 3.5). The zone size can use two different values, depending on the length of the desired path that we want to manage in the routing stage of the protocol. We work with a larger zone size if shorter paths are desired. However, if larger paths are better, then we work with a smaller zone size. The length and width corresponding to the zone sizes are calculated using the formulas in Eq.(3-1) and Eq.(3-2), where w denotes the length of the zone's size:

$$\begin{aligned}
 L^2 &= (2w)^2 + (2w) \\
 L^2 &= 8w^2 \\
 L &= 2\sqrt{2}w \\
 \frac{L}{2\sqrt{2}} &= w \\
 \text{Zone's area}_1 &= \left(\frac{L}{2\sqrt{2}} \right) \times \left(\frac{L}{2\sqrt{2}} \right)
 \end{aligned}$$

Eq.(3-1): Zone Size₁

$$\begin{aligned}
 (2L)^2 &= (3w)^2 + (3w) \\
 4L^2 &= 18w^2 \\
 \sqrt{2} \times \sqrt{2} \times L &= 3\sqrt{2}w \\
 \sqrt{2}L &= 3w \\
 \frac{\sqrt{2}L}{3} &= w \\
 \text{Zone's area}_2 &= \left(\frac{\sqrt{2}L}{3} \right) \times \left(\frac{\sqrt{2}L}{3} \right)
 \end{aligned}$$

Eq.(3-2): Zone Size₂

The zone size determines the neighboring sensors, either within a zone, or between the sensors of a zone and those in any of the possible neighboring zones around it.

Figure 3.5: Zone Size and Neighboring Zones in the Cluster Field.
(a) Zone Size₁, (b) Zone Size₂

In the first case (Figure 3.5a), where zone size = zoneSize₁, any two sensors in two neighboring zones are within the transmission range of each other and can communicate directly. This means, that a sensor in any position of a zone can communicate directly with any other sensor in its neighboring zones. At the same time, we have smaller zones that will require a higher number of cluster heads in the sensor field and also, longer paths during the routing stage of the protocol (larger number of hops).

In the second case (Figure 3.5b) a sensor in the center of a zone can communicate directly with any other sensor in its 8 neighboring zones. This size produces larger zones and shorter routing paths (smaller number of hops), which can lead to a lower end-to-end delay, but also can potentially decrease the efficiency of the protocol when the transmitting sensor is not located in the center of the zone.

Among the information maintained by every sensor node in the cluster field, the following aspects are related with its location in the zones: node id, location (x, y), zone id. Each sensor can get its Zone ID (i, j) by using the formulas in Eq.(3-3) and Eq.(3-4), where (x_1, y_1) is the sensor's current location, (x_2, y_2) is the sensor field's origin and z is the zone size being used.

$$i = \left\lfloor (x_1 - x_2) \div z \right\rfloor$$

Eq.(3-3): Zone ID's i Coordinate Calculation

$$j = \left\lfloor (y_1 - y_2) \div z \right\rfloor$$

Eq.(3-4): Zone ID's j Coordinate Calculation

3.3.1. Considerations for the Information Update

Update mechanisms of information can be classified as time-based, distance-based or event-based. Time-based mechanisms update information periodically, distance-based mechanisms update information whenever a node moves for a certain distance and event-based mechanisms update information when an event occurs. In CROMOS we use a combination of time-based and event-based update mechanisms to control the operations to be performed by the sensors.

A general update timer is used to control the period used by the sensors to calculate their new positions and to verify if they are still in the same zone as before or if they have changed to a new zone.

Figure 3.6: Update Time Scenario

To determine the value of the update timer, let us consider the situation presented in Figure 3.6, where the zone head is in one corner of the zone and the member node is in the opposite corner of the zone. They may move in opposite direction at the maximum speed. To guarantee that any member node is in the transmission range of its zone-head, the value of the update timer is given by Eq.(3-5):

$$Update\ Timer = \frac{MTR - \sqrt{2} \times Field_width}{2 \times MMV}$$

Eq.(3-5): Update Timer Value

According to this, we obtain equations Eq.(3-6) and Eq.(3-7) for the zone sizes presented in Eq.(3-1) and Eq.(3-2), respectively:

$$Update\ Timer = \frac{MTR}{4 \times MMV}$$

Eq.(3-6): Update Timer Value for Zone Size₁

$$Update\ Timer = \frac{MTR}{6 \times MMV}$$

Eq.(3-7): Update Timer Value for Zone Size₂

We can see from the previous equations that the update timer is determined by the MMV, the MTR and the zone size being used. With the same MTR, when the zone size is larger, the update timer is shorter. The update timer functioning in the WSN may help avoid the *Ping-Pong effect*, which is caused by frequent movements of a sensor back and forth between two zones. However, this does not solve the ping-pong effect and we do not consider any other mechanisms to solve it, since it is not critical for CROMOS operations.

The second mechanism used to update information in CROMOS is event-based. When a cluster head needs to send information towards the base station, it generates an event that triggers the transmission processes in the network.

3.4. Cluster Head Election Process

In this section we explain the process used to select the sensor that will act as the CH on each zone of the sensor field. The criterion use for this selection is the Mobility Factor of each one of the sensors in the zone.

3.4.1. Mobility Factor

The **Mobility Factor** is an indication of the node's mobility. It is defined as the average number of times a node moves from one zone to another during a specific period of time. This factor is used in selecting a node to be a cluster head. The mobility factor is defined as follows. Eq.(3-8):

$$\text{Mobility Factor} = \frac{\text{Zone Changes}}{\text{Location Changes}}$$

Eq.(3-8): Mobility Factor

By definition, the Mobility Factor is always less than 1. A node with low mobility factor means it either moves with a low speed or moves locally in a specific region. In this last case, the node's mobility factor is 0. The node with the smallest mobility factor becomes the next zone-head, as we will explain in the next section where we describe the processes of the protocol.

The zone head is within the transmission range of all the zone-heads in its neighboring zones. The zone head plays a very important role in CROMOS, since it is responsible for forwarding route discovery requests to neighboring zones, forwarding data packets to its next cluster head hop along the path and maintaining information about the member nodes.

3.4.2. Cluster Head Election Criteria

The cluster head election routine is shown in Figure 3.7. To initiate the cluster head election process, all sensors consider themselves as a possible cluster head (Role =

POSSIBLECH). The sensor compares its mobility factor with the mobility factor of the other sensor in its same zone. If the sensor finds a sensor with a smaller mobility factor, it sets its own role as MEMBER in the zone and waits for a Cluster Head advertisement from another sensor in its same zone. Once it receives the advertisement, it stores a reference to the elected cluster head for further use.

If the sensor finds that its mobility factor is the smallest one among the sensors in the zone, it sets its role as CLUSTERHEAD and becomes the cluster head in the zone because it is the most stable sensor available. It also increases its TimesCH counter. After this step the sensor announces its new role to the other sensors in the zone, receives their messages announcing their membership to the cluster and adds a reference to them to its *Sensors In Zone* list.

If during the cluster head election two nodes have the same Mobility Factor, the protocol checks which one of them has acted as cluster head the least amount of times and selects that node as the new cluster head. If there is also a tie in the number of times the nodes had been cluster heads, and one of the nodes was the cluster head in the previous cycle of the protocol, the other node is selected as cluster head. Otherwise, the next criterion is to simply select the node with the highest sensor ID.


```

/* To detect if the sensor should act as Cluster Head or Member, it
 * has to compare its Mobility Factor, its TimesCH and its ID
 * attributes against other nodes in the same zone.
 */
Cluster_Head_Election(){
    Sensor1 = This_Sensor;
    if (Sensor1.Role != CLUSTERHEAD)
        Sensor1.set_Role(POSSIBLECH);
    while (Sensors in this sensor's zone){
        Sensor2 = next_Sensor_in_Zone;
        if (Sensor1.MobilityFactor > Sensor2.MobilityFactor){
            Sensor1.set_Role(MEMBER);
            break;
        }else{
            if (Sensor1.MobilityFactor == Sensor2.MobilityFactor)
                if (Sensor1.Times_CH > Sensor2.Times_CH){
                    Sensor1.set_Role(MEMBER);
                    break;
                }else{
                    if (Sensor1.Times_CH==Sensor2.Times_CH)
                        if (Sensor1.Role == CURRENTCH){
                            Sensor1.set_Role(MEMBER);
                            break;
                        }else{
                            if (Sensor1.ID < Sensor2.ID){
                                Sensor1.set_Role(MEMBER);
                                break;
                            }
                        }
                }
        }
    }
    if ((Sensor1.Role==POSSIBLECH) OR (Sensor1.Role==CURRENTCH)){
        Sensor1.set_Role(CLUSTERHEAD);
        Sensor1.increment_Times_CH_counter;
        Send Cluster_Head_Elected message to other sensors in Zone;
    }else{
        Reset Sensors In Zone;
        Reset Neighboring CHs;
    }
}
}

```

Figure 3.7: Cluster Head Election Process

Keeping a list of zone members in the cluster head allows CROMOS to receive messages directly from those nodes and allows the cluster head to act as their gateway to the base station. It can be argued that keeping this list consumes memory space in the cluster head which can be a problem due to the exponential growth of information with the network size and the small memory capacity available in the sensors. For example, sensor nodes such as the Telos and MicaZ have notoriously small memory sizes: the MSP430 microprocessor has only 10 KB of RAM, while the Atmega 128L has just 4 KB. However, because CROMOS only maintains a reference to the nodes in the list, then the size will not be difficult to manage for the cluster heads.

3.4.3. Zone-Head Maintenance

If a zone head updates its position and detects that it is leaving the zone that it is serving as a cluster head, it performs the following actions:

- Checks if it has data pending to be sent. If it does, the sensor sends that data to the Cluster Head in the new zone that it is entering, to ensure that it will be sent towards the base station.
- Changes its role to MEMBER and keeps following the normal processes of the protocol.

3.5. Neighbor Discovery Process

CROMOS relies on the communication links existent among the cluster heads in the sensors field's zones. The mechanism used to know about the cluster heads that are

reachable from every cluster head in the field is the Neighbor Discovery Process. The routine for this process is shown in Figure 3.8. Using the MTR value, each node creates a list of the nodes that are at a distance less than or equal to its maximum transmission range.

```

/* This method defines the CHs that are going to get the
 * broadcasted Vicinity Message because of being in the CH's
 * transmission range.
 */
Define_Neighboring_CHs(){
    while (CHs in the list){
        Calculate distance to other CH;
        If (distance to CH < MTR)
            Add CH to neighboring CH list.
    }
}

```

Figure 3.8: Neighbor Discovery Process

3.6. Zone-based Routing Process

As we explained in Section 1.7, when designing a routing protocol for wireless environments, it is necessary to satisfy requirements like: distributed routing, minimum delay in the route discovery process, loop-free routes, minimum overhead, routes reconfiguration and scalability.

Some of the design criteria used in CROMOS to fulfill these requirements are:

- a) Only the cluster heads act as routers in the network,

- b) The routes creation process is made by sending vicinity messages from the base station's neighboring zone heads to the most exterior zone heads in the sensor field.
- c) Instead of storing complete paths towards the BS, each CH stores a list of the CH in the previous level closest to the BS and selects one of those CHs as its next hop in the routing process.

3.6.1. Zone-based Routing Operation

Route discovery allows any mobile node to dynamically discover a route to the base station in the network, whether the base station is directly reachable within the node's wireless transmission range or reachable through one or more intermediate cluster heads.

3.6.1.1. Overview of Route Discovery Procedure

The route discovery procedure is executed in a coordinated way among the cluster heads in the sensor field, and it is performed at the same time that the nodes obtain the information about its neighboring cluster heads.

In CROMOS, instead of storing complete routes, the CHs store information about the neighbors in the previous level towards the BS. The levels definition process is initiated by the cluster heads that are located closer to the base station and can reach it directly (single-hop routes). These cluster heads are labeled as "Level-1" cluster heads. Level-1 cluster heads send a Vicinity Message to their neighbors, which are labeled as "Level-2" cluster heads. Once level-2 cluster heads receive the vicinity messages, they will repeat

the process with their own neighbors, and the process will continue until the most exterior zone heads in the sensor field are reached.

Using only the CHs with the previous defined level as the possible CHs to be used as the next hop to forward messages, ensures the creation of loop-free routes and avoids the *count to infinity* effect that the routing loops may cause, since the CH are not going to use CHs that have the same level or a higher level than they do.

The second advantage of using levels is that the CHs are able to detect when they are out of reach of the base station. If a CH doesn't have any sensor in the previous level of the network, it stores the data until the next neighbor discovery process is performed. If after this process the sensor does not have any neighbors in the previous level of the network, the CH dies and gets excluded from the following sensing and network operations in the WSN.

An additional advantage of not collecting complete routes towards the base station is avoiding the exchange of route request (RREQ) and route reply (RREP) messages between neighboring CHs. This saves energy on the CHs, increases their lifetime inside the network and reduces the overhead of the CROMOS protocol.

3.6.1.2. Description of Zone-based Routing Protocol

CROMOS acts as a proactive routing protocol. Inside every zone the communication between the sensor nodes and the cluster head is made directly to the cluster head that is

located at a single-hop distance from the sensor node. When a member node senses data, it immediately creates a data packet and sends it to its corresponding zone head.

Once the cluster receives the data packet from the cluster member, it checks its list of CHs in the previous network level. If there is only one CH in the previous level towards the base station, the cluster head sends the data to that sensor node. If there are two or more possible CHs in the previous level, the zone-head chooses the *best node*, and sends the data to that CH.

The selection of the best previous node is calculated based on the cluster head mobility factor and CH zone's density (i.e. the number of nodes that belong to the cluster head's zone). Figure 3.9 shows the previous CH election process.

```

/* This method selects the best previous node that a CH has to use
 * in order to send the data towards the BS.
 * The criteria to select the node are:
 * a) the sensor has the least MobilityFactor
 * b) If 2 or more sensors have the same MobilityFactor, then use
 * the sensor's Density which means to select the CH that is
 * covering more sensors in the sensor field.
 */
Select_Previous_CH(){
    /* Stores the smallest Mobility Factor among
     * the sensors being analyzed */
    smallest_MF=-1;
    /* Stores the highest density (number of sensors)
     * among the sensors being analyzed */
    highest_Density=0;

    Iterator iter = this.getSensorsInPrevLevel().iterator();
    while (Sensors_in_Previous_Level){
        node = next_Sensor_in_Previous_Level;
        if (node.Role==CLUSTERHEAD){
            if (smallestMF==-1){
                smallest_MF = node.MobilityFactor;
                selected_Node = node;
            }else{
                if (node.MobilityFactor<smallestMF){
                    smallest_MF = node.MobilityFactor;
                    selected_Node = node;
                }else{
                    if (node.MobilityFactor==smallestMF){
                        if (node.Density > highest_Density){
                            highest_Density = node.Density;
                            selected_Node = node;
                        }
                    }
                }
            }
        }
    }
    return selected_Node;
}

```

Figure 3.9: Previous CH Selection

Once a cluster head has sent the data to the next CH and receives the acknowledgment message, it assumes that it is the next node's responsibility to forward the messages towards the base station. It relies on the routing protocol design and "forgets" about the messages already sent.

3.6.1.3. Route Maintenance

Since CROMOS does not store routes, it does not deal with normal route maintenance processes. However, CROMOS has two different mechanisms to try to avoid the link breakage in the network. The first mechanism occurs when a CH moves from its current zone to a new one and still has data packets to send towards the BS. In this case, the CH passes along the messages to its new CH (in the new zone it belongs to), for that CH to deal with the transmission of the packets.

The second mechanism is used when the CH doesn't have any sensor in the previous level of the network and stores the data until the next neighbor discovery process is performed. This allows the CH to recover from the previous connectivity loss and, if it has new neighbors in the previous level, this allows continuing the transmission of the packets towards the BS.

3.7. Summary

In this chapter we provided a detailed description of our cluster-based routing protocol for mobile sensor networks. We explained how to partition the network area into zones

and the strategies to select a cluster-head in each zone. We presented how the protocol updates the sensor's information and how those sensors react to the topology changes due to mobility. We also explained how the protocol selects the best CH to forward the data packets according to its mobility factor and zone density criteria.

CHAPTER 4

PERFORMANCE EVALUATION

In this chapter we evaluate the main characteristics of our protocol and study their impact on the performance of CROMOS. A packet-level simulator was developed using the Java programming language to evaluate the performance of CROMOS. The GUI for our simulator allows following the sensor's movement inside the sensor field and allows identifying which sensors are acting as cluster heads and which ones are acting as members at each step of the simulation. Figure 4.1 shows a sensor field with 16 defined zones. The sensors shown as circles (•) are the cluster heads and the sensors shown as squares (▪) are the members. The sensors are shown as triangles (▲) when they are executing the cluster head election process to determine if they must act as a cluster head or as a cluster member inside their current zone.

Figure 4.1: Simulator's GUI

4.1. Simulation Model

In order to evaluate the behavior of CROMOS in different WSN configurations, simulation was used to determine the effect of the clustering and routing components included in the CROMOS protocol over the WSN. The simulations were evaluated in terms of data sent by the sensors/data received by the BS, ratio energy dissipation and network lifetime.

4.1.1. Sensors Deployment

We assume that the nodes are deployed randomly in the sensor field. The density of nodes in the sensor field (i.e. amount of nodes per zone) is a critical aspect of our

protocol. At low sensor densities, the WSN will have coverage gaps. These gaps are caused by the presence of sensors that do not have any reachable neighbor in order to route the messages towards the base station. Thus, the minimum number of sensors deployed in the sensor field will depend on the Maximum Transmission Range of the sensors and therefore on the zone size being used for the simulation.

To illustrate this concept, we consider a coverage area for the number of zones that can be covered for a single sensor, and we examine two cases:

Case 1: zone size 1. In this case the coverage area for a sensor will be 16 zones.

Case 2: zone size 2. In this case the coverage area will be 9 zones.

In both cases, to cover the entire sensor field, we will need at least $n + (n - 1)$ sensors, where n is the number of coverage areas, and assuming that the sensors are distributed in such a way that they are within the transmission range of at least 2 other sensors or one sensor and the base station. So the density for our protocol should be equal or greater to $n+(n-1)$ sensors. This must be determined prior to deciding the number of sensors to deploy inside the sensor field.

4.1.2. Energy Model

Energy consumption in mobile WSN includes two parts: movement and communication. In CROMOS we assume the sensors are mounted on a moving device which has its own power source and therefore, we exclude movement from the energy consumption operations.

For the energy consumption due to communication, we need to consider the different activities that make the sensor consume energy during the simulation:

- Cluster head election
- Neighbors discovery
- Sending information

Each one of these operations consists basically of the reception and transmission of messages between the member sensors and its CH or among the cluster heads.

We assume a contention-free and error-free communication environment, and therefore sensors do not have to retransmit any data (i.e. try to transmit the same data many times). The sensors use radio-frequency (RF) communication, which allows the packets conveyed in the sensor network to be small and with low data rates (i.e. generally less than 1 Hz) [Aky1], and also allows a high frequency reuse due to short communication distances. The radios have power control and can expend the minimum required energy to reach the intended recipients.

We implement the First Order radio Model proposed in [Hei1] and used in other papers like [Lin1], [Kam1] and [You1]. This model (Figure 4.2.) is composed by two different sub-modules: a transmitter/receiver circuitry (Elec) and a transmitter amplifier (Amp).

Figure 4.2: First Order Radio Model

A sensor node consumes E_{elec} (nJ/bit) in transmitter or receiver circuitry and E_{amp} (pJ/bit/m²) in transmitter amplifier.

A sensor node expends energy $E_{Tx}(k, d)$ or $E_{Rx}(k)$ in transmitting or receiving a k -bit message to or from distance d , given by the formulas in Eq.(4-1) and Eq.(4-2).

$$E_{Tx}(k, d) = E_{Tx-elec}(k) + E_{Tx-amp}(k, d)$$

$$E_{Tx}(k, d) = E_{elec} \times k + E_{amp} \times k \times d^2$$

Eq.(4-1): Energy Expended to Transmit a k -bit Message with a Distance d

$$E_{Rx}(k) = E_{Rx-elec}(k)$$

$$E_{Rx}(k) = E_{elec} \times k$$

Eq.(4-2): Energy Expended to Receive a k -bit Message

The values used for E_{elec} and C_{Amp} are:

Table 4.1: Radio Model Values

Operation	Energy Dissipated
Transmitter Electronics ($E_{\text{Tx-elec}}$) Receiver Electronics ($E_{\text{Rx-elec}}$) ($E_{\text{Tx-elec}} = E_{\text{Rx-elec}} = E_{\text{elec}}$)	50 nJ/bit
Transmitter Amplifier	100 pJ/bit/m ²

The load of a node is the power that node consumes to transmit and receive data. The higher the load is, the shorter the lifetime of the node [Luo1]. Taking this into account and by considering the parameter values presented in Table 4.1, our CROMOS protocol tries to minimize not only the transmission distances but also the number of transmit and receive operations for each message, since these are high energy cost operations. Transmission cost is a good estimate of energy cost in WSN because radio transmissions consume far more energy than any other operation in a node. A protocol is energy efficient if it minimizes the accumulative energy consumption for fulfilling its task (e.g. minimum-energy broadcast).

4.1.3. Mobility Model

We use two different speed ranges for our simulation experiments. The first range, between 0 and 5 m/s considers sensors with low mobility, like the sensors attached to a turtle in [Tur1]. The second range is between 6 and 10 m/s and considers sensors with a medium mobility, like sensors attached to a running person. A sensor randomly chooses

an initial location in the sensor field, then randomly chooses a direction and a target point to move to. When the sensor reaches the targeted location, it randomly selects a new direction and a new target location and keeps repeating the process until it runs out of energy to function.

4.1.4. Traffic Model

There are two different traffic packets in our simulation: control packets and data packets. Control packets for the clustering process include vicinity messages, cluster head advertising messages and join messages. Control packets for the routing process are neighbor discovery messages and level definition messages. For simplicity, we define the length of these control packets to be 10 bytes. Data packets have a fixed size of 512 bytes. These data packets are generated randomly, one every time the mobile node senses data in the field.

4.2. Simulation Results

In our simulations, numerous experiments are conducted in order to evaluate the performance of CROMOS. We change different input parameters such as the number of nodes in the network, the MTR and MMV for the sensors and the zone size. We make these changes to evaluate their impact on the network performance.

Our protocol evaluation is based on the simulation for 100 to 200 wireless mobile sensors forming a WSN, with increments of 25% each time. The sensors are moving over a

square flat space of 300m x 300m. For simplicity, we assume the probability of signal collision and interference in the wireless channel can be ignored. Each simulation experiment is run until all nodes reach their minimum energy point, which is 80% lost of its initial energy, or until the nodes are out of the transmission range of all other nodes in the network. Different values are used for the Maximum Transmission Range and Maximum Moving Velocity of the nodes, in order to evaluate their impact over the zone sizes used and over the clustering and routing processes used to transmit the messages towards the base station during the network lifetime.

We evaluate the performance of CROMOS in terms of: network lifetime, dissipated energy and packets delivery ratio. Network lifetime is the time span from the sensors deployment to the moment when the network gets disintegrated due to the energy consumption of the nodes or due to their inability to connect to other sensors required to send data towards the BS.

The second parameter we evaluate in our simulations, and closely related with the network lifetime, is the average of dissipated energy in the network. This parameter shows the average dissipation of energy per node in the network. Since the nodes have limited energy, they use up this energy during the course of simulation. The energy decreases every time a node sends or receives data or control messages within the network.

We use the energy utilization rate η to evaluate the efficiency of energy consumption, which is defined as the ratio of the total energy consumed when the first node dies to the initial total energy. A high value of η implies that energy consumption is distributed well across the network.

In sensor networks, power efficiency is an important performance metric directly influencing the network lifetime [Aky1]. In the best case, we would like each sensor to have enough energy to perform its functionality as long as possible. However, in WSN due to the existent energy constraints, this constitutes a very big challenge not only for routing, but for any other operation performed by the sensors.

In order to be able to send messages towards the BS, we do try to avoid any coverage gaps in the network. We therefore also consider the time when the first node dies inside the network, as the beginning of the network disintegration. A node is considered dead if it has lost 80% of its initial energy. Once a node runs out of energy, it is considered dead and can no longer transmit or receive data. However, since the nodes that remain alive can become CH and close the gaps formed by the dead sensor, we also monitor the number of nodes alive after each cycle of the simulation. The total number of nodes alive gives an idea of the area coverage of the network over time.

The third parameter used to evaluate CROMOS' efficiency is the ratio between the number of packets sent by the sensors and the number of successfully received packets at

the BS. The more data the BS receives, the more accurate its view of the remote environment will be.

4.2.1. WSN with Small Zones and Low Velocity

For this set of simulations we first use only the mobility factor as the criterion for the selection of the CH during the clustering process and for the selection of the next hop node during the routing process.

Our second set of simulations for this group uses a combined criterion for the selection of the CH during the clustering process and for the selection of the next hop node during the routing process. In our simulation we combine the sensor's mobility factor and the remaining energy available in the sensor to evaluate their eligibility as CH. We use weights of 67% for the mobility factor and 33% for the remaining energy. Figures 4.3 and 4.4 show the results obtained from these simulation sets.

(a)

(b)

(c)

Figure 4.3: Simulation Results for WSN with Small Zones and Low Velocity

From Figure 4.3 (a), we can see that the number of messages received at the base station is proportional to the number of sensors as expected. There is a growth of the number of packages received, which means that if all sensors are functional, the number of cumulative messages grows. In the last cycles of the graph, we can observe the flattening of the curve to a maximum cumulative number of messages, which indicates the expiry of the network due to sensors death. In all our experiments, sensor death and curve flattening appear to happen at the same time, independent of number of sensors.

The behavior of CROMOS related to the number of packets that traverse the network and reach the base station shows that in spite of the incorporation of sensors mobility, it does not decrease the network performance in terms of number of messages at the BS.

Figures 4.3 (b) and 4.3 (c) show that the number of sensors does not impact the total energy state of the system versus time. Energy consumption is proportional to the number of sensors (hence number of messages transmitted). The number of nodes alive at each cycle shows that energy drain on the cluster heads is proportional to the number of sensors (hence number of messages) in the WSN, resulting in consistent performance.

Figure 4.4 shows the differences obtained when combining the Mobility Factor with a 67% weight and the Remaining Energy with a 33% weight, as the criteria for the CH election and the previous forwarding hop election. By trying to take into account the remaining energy and the mobility factor, in the cluster head and next hop election, it causes three negative consequences: the networking lifetime is decreased significantly (approximately 40 percent) versus the simulations where we use only the mobility factor, the number of packets received at the base station decreases and the percentage of lost messages increases for higher node systems. Some of the reasons that produce these effects are:

- If the higher mobility factor sensors are elected cluster heads due to their higher energy, this causes network discontinuity because of their greater movement.
- If the sensors with higher mobility remain only as members, they are more efficient in transmitting data to the cluster heads, since they only have to find out which one is

their new cluster head and send the data to that new cluster head, but they can continue to work without further inconvenience.

- If the sensors with higher mobility are elected cluster heads, they will consume additional energy because of dealing with the normal cluster head networking operations.

(a)

(b)

Nodes Alive

(c)

Figure 4.4: Simulation Results for WSN with Small Zones and Low Velocity, Combining Mobility Factor and Remaining Energy

4.2.2. WSN with Big Zones and Low Velocity

In our second set of simulations we modify the zone size to use bigger zones (zoneSize2).

Figure 4.5 shows the effects of this change in the simulation results.

(a)

Figure 4.5: Simulation Results for WSN with Big Zones and Low Velocity

In this case if the transmitter is not in the center of the zone, this creates gaps in the network since some cluster heads are not reachable to send or receive data. This causes higher lost packets and a smaller number of packets arriving to the base station. The update period is smaller which results in the sensors performing network operations more frequently and having less time and energy for sensing. For the larger zone, there are more frequent updates which should consume more energy than the less frequent updates in the small zone system, for equal number of messages transmitted. However, the graphs show the small zone networks consuming more energy, which can be explained by a higher number of data packet transmissions compared to the large zone networks.

The gaps in the larger zone are very significant. The larger zone systems are less efficient for transmitting data. This also influences the life of the nodes. The small zone nodes have a shorter lifespan because they are more efficient at transmitting data. In the large zone system the nodes remain alive longer, but the nodes trapped in the uncovered areas among the zones retain their energy but are less efficient because they are not being functional within the network.

4.2.3. WSN with High Velocity

CROMOS performs well at low velocities. To study the effect of a velocity increment, we run simulation experiments using a higher velocity. The results for these experiments can be seen in Figure 4.6. As can be observed, the higher velocity of the nodes causes a higher packet loss and a lower amount of total messages received at the base station.

If we use the equation for the update timer, the loss of efficiency would be even more pronounced due to the smaller resultant value. We would have to perform the network operations more frequently than it is possible in our simulator. To try to overcome the short update period, we used a fixed update period of 20 and resulted in a higher level of messages getting lost on their way towards the BS due to the lost of connectivity between the members and the CHs.

(a)

(b)

Nodes Alive

(c)

Figure 4.6: Simulation Results for WSN with High Velocity

Figure 4.7 shows the different packet loss rates we obtain during our simulations. This packet loss increases with the following factors: higher density, increased velocity and larger zones.

Figure 4.7: Packet Loss rates

4.3. Discussion of performance of CROMOS

In this section we discuss the impact of the parameters evaluated in the previous simulations over the total performance of CROMOS.

4.3.1. The Effect of Clustering

CROMOS has the advantage of being distributed and self-configuring. Both the clustering process and the routing process in CROMOS are performed by the mobile

nodes in a distributed manner. Also, the existence of a CH in each one of the zones of the sensor field, allows the cooperative transmission of messages towards the BS. By guaranteeing the existence of this unique CH per zone, CROMOS minimizes the total energy spent by the member nodes, since they only send data to their correspondent CH. In our simulation all the mobile nodes in the network may sense and generate data to send to the base station while they are acting as member inside the clusters.

While in some other clustering protocols as LEACH [Hei1] there is no guarantee as to the number or placement of CH nodes within the network, in CROMOS these values are known from the beginning of the protocol execution which facilitates the data transmission and improves the functionality of the routing protocol.

Unlike static clustering where the CHs are fixed, the rotation of the role inside the zone increases the time the nodes stay alive. Dynamic clustering allows the network to stay alive for a longer time than static clustering. Static clustering performs poorly because the CH nodes die quickly ending the lifetime of all nodes belonging to those clusters, because they have lost communication with the base station. The network lifetime is relatively short. With dynamic clustering more data can be delivered to the BS and the network lifetime expands.

4.3.2. The Effect of Mobility

CROMOS consider the mobility of the nodes while determining the clusters and the routes used to forward packets, thus choosing a more stable path and reducing the impact

of mobility on the network performance. By calculating the probability of the sensor moving out of their zones, CROMOS reduces the amount of time and messages required to maintain the network clusters and communication. CROMOS considers the nodes' mobility and makes them select a new CH to send the data to every time they change from one zone to another. If the node changing zones is a cluster head, the data that it still has to transmit is sent to its new assigned CH in order for that CH to look for a new path to send the data towards the BS. Since the data remains in one of the nodes within the network, the probability of data loss is reduced.

4.3.3. The effect of Nodal Density

The experiments Section 4.2 also studied the effect of density on the performance of CROMOS. We chose the number of nodes in the network to be 100 to 200 nodes. In flat-routed WSN, we know as the number of nodes increases, more nodes could act as routers, increasing the numbers of links existent in the network and increasing the complexity of routing. Since more nodes would take part in the route discovery procedure and broadcast RREQ messages to neighboring nodes, the overhead of routing would become large.

In CROMOS, as the number of nodes increases, the performance does not degrade as in flat-routed WSN. This is because of the clustering technique used to make only the CH act as routers. The number of CHs is more dependent on the number of zones than the number of nodes, when the average number of nodes in each zone is at least one (a well-

distributed WSN). This shows the scalability of CROMOS, as it adapts to high density WSN.

When the number of nodes in the network is high (sensor density increases), the probability of a zone to be empty is decreased, thus the probability of having holes inside the network is decreased and in turn the probability of losing data is decreased too. Since CROMOS only uses some of the mobile nodes as routers every time (the CHs), the overhead in routing does not increase with the increase of number of nodes as in a flat routed WSN.

4.3.4. The Effect of Zone Size

We compare the performance when partitioning the network into smaller zones (zone size= Zone Size₁) or larger zones (zone size= Zone Size₂). CROMOS' behavior is consistent for both zone sizes. However, in larger zones nodes are less likely to move out of their zone frequently than in a smaller zone, thus reducing the overhead of information update.

4.3.5. The Effect of Using Levels

The cluster heads in CROMOS do not maintain full routes to the BS. Instead, they define CH levels that indicate how far that zone head is from the BS. This minimizes the amount of data the CH have to store and also minimizes the number of messages exchanged between the CHs to find a path towards the BS.

The normal behavior for routing protocols is to exchange request messages (RREQ) and reply messages (RREP) among the nodes to obtain the different routes available to send data from a source to an objective point of the network. However, this messages exchange can be consume a large amount of resources especially in WSN where resources like power and storage capacity are limited. This is the reason for CROMOS to use a level definition process instead of a routes collection process. By defining levels, each CH only needs to store maintain a list of the sensor located in the previous level from the BS, instead or maintaining a list of routes than can grow exponentially depending on the number of neighbors that each zone head may have.

By using multiple hops to send the data towards the BS, we also save energy in the CH, since the distance of their receiver is smaller and therefore they need less energy to transmit the data.

4.4. Summary

In this chapter, we evaluated the performance of CROMOS and showed how the clustering scheme determined by the definition of zones inside the sensor field helps defining the number of cluster heads we have in the network and reduces the energy dissipation in the nodes by limiting the distances they have to cover when sending messages towards the base station.

Considering the mobility of the sensors as the main factor to select the cluster head inside the zone helps increasing the reliability of the network and decreasing the link breaking on the paths toward the sink.

We used the network lifetime, dissipated energy and packets delivery ratio as the performance indicators to evaluate the benefits of CROMOS, and we showed how aspects like mobility, nodes density and the zone sizes affect the behavior of the network.

CHAPTER 5

CONCLUSIONS

In this thesis we propose a Cluster-based routing protocol for mobile sensor networks (CROMOS) for WSN with low mobility sensors. The main characteristics of CROMOS are:

1. We use a fixed zone-based partition scheme to partition the network and a clustering scheme to save energy in the sensors. The algorithm is simple. The overhead in zone maintenance and cluster-head election is low.
2. CROMOS uses self-definition of the mobility of the sensor nodes using local data. We define the mobility factor of mobile sensor nodes according to localized data, which can be collected and estimated at each node. The definition of mobility factor considers both the moving speed and the localization of the sensors.
3. The mobility factor of sensor nodes is the main criterion used to select the best node to act either as CHs, or as previous node when routing data packets.
4. The update of node's information is time-driven, which means that only when the update timer expires the sensor nodes update their information and exchange the neighboring information and level definition needed to route packets towards the BS.
5. The update-timer is flexible, which is a value related to the mobility of the nodes and the zone size. This can keep the zone maintenance valid with a lower communication overhead.

CROMOS shows a good performance in WSN using smaller zone sizes and low speed sensors.

5.1. Future work

Some aspects we consider to improve the functionality of CROMOS in future research include:

1. In sensor networks where there is a large correlation within the network, data aggregation can be performed by the CH to reduce the amount of messages actually transmitted towards the BS. If all the nodes within a cluster are sensing the same event, the actual and effective data will contain the same information and there is no loss in quality by sending effective or aggregate data rather than actual data. Another option is for the CH not to aggregate the data, but to select the most significant data to send to the BS (i.e. select the strongest event or the highest or lowest value).
2. Increase the energy utilization rate by considering the residual energy on the CH and using it as a weighted parameter for the election of the next node to send the messages towards the BS. This will increase the node's lifetime.
3. Changing the transmission criterion for the CH to transmit messages only when its data cache has reached a percentage of its maximum capacity, will save energy by decreasing the number of data packets transmission.

BIBLIOGRAPHY

[Aky1] Akyildiz I.F., Su W., Sankarasubramanian, Cayirci E., *“Wireless Sensor Networks: A Survey”*, Georgia Tech Technical Report, December 2001.

[Alk1] Al-Karaki J.N., Kamal A.E., *“Routing Techniques in Wireless Sensor Networks: A Survey”*, Department of Electrical and Computer Engineering, Iowa State University, December 2004.

[Arb1] Arboleda L.M., Nasser N., *“Comparison of Clustering Algorithms and Protocols for Wireless Sensor Networks”*, IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), Ottawa, May 2006.

[Aur1] Aurenhammer F., *“Voronoi Diagrams - A Survey Of A Fundamental Geometric Data Structure”*, ACM Computing Surveys 23, pp. 345- 405, September 1991.

[Bar1] Barbeau M., Kranakis E., Krizanc D., Morin P., *“Improving Distance Based Geographic Location Techniques in Sensor Networks”*, 3rd International Conference on ADHOC Networks and Wireless, Vancouver, British Columbia, July 2004.

[Ben1] Benenson Z., Gärtner F., Kesdogan D., *“User Authentication in Sensor Networks (Extended Abstract)”*, Germany, 2004.

[But1] Butler Z., Corke P., Peterson R. and Rus D. “*Networked Cows: Virtual Fences for Controlling Cows*”. WAMES 2004, Boston, USA, June 2004.

[Cha1] Chan H., Perrig A., “*ACE: An Emergent Algorithm for Highly Uniform Cluster Formation*”, Proceedings of the First European Workshop on Sensor Networks (EWSN), January 2004

[Che1] Chen WP., Hou J. C., Sha L. , “*Dynamic Clustering for Acoustic Target Tracking in Wireless Sensor Networks*”, 11th IEEE International Conference on Network Protocols (ICNP'03), pp. 284-294, 2003.

[Fan1] Fang Q., Zhao F., Guibas L., “*Lightweight Sensing and Communication Protocols for Target Enumeration and Aggregation*”, Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC), p. 165-176, 2003.

[Hei1] Heinzelman W.B., “*Application Specific Protocol Architectures for Wireless Networks*”, PhD thesis, Massachusetts Institute of Technology, June 2000.

[Hei2] Heinzelman W., Chandrakasan A., Balakrishnan H., “*Energy-Efficient Communication Protocol for Wireless Microsensor Networks*”, Proceedings of the 33rd Hawaii International Conference on System Sciences, January 2000.

[Ibr1] Ibriq J., Mahgoub I., "***Cluster-Based Routing in Wireless Sensor Networks: Issues and Challenges***", Proceedings of the 2004 Symposium on Performance Evaluation of Computer Telecommunication Systems, 2004.

[Int1] Intanagonwiwat C., Govindan R., Estrin D., "***Directed diffusion: A scalable and robust communication paradigm for sensor networks***", Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MobiCOM '00), Boston, Massachusetts, August 2000.

[Jua1] Juang P., Oki H., Wang O., Martonosi M., Peh L-S. and Rubenstein D. "***Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with Zebranet***", ASPLOS, San Jose, CA, Oct. 2002.

[Kam1] Kamimura J., Wakamiya N., Murata M. "***Energy-Efficient Clustering Method for Data Gathering in Sensor Networks***", First Annual International Conference on Broadband Networks, BROADNETS, San Jose, CA, Oct. 2004

[Lin1] Lindsey S., Raghavendra C.S. "***PEGASIS: Power-Efficient Gathering in Sensor Information Systems***", IEEE Aerospace Conference, 2002.

[Liu1] Liu C., Wu K., Pei J., "***A Dynamic Clustering and Scheduling Approach to Energy Saving in Data Collection from Wireless Sensor Networks***", Second Annual

IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON 05), Santa Clara, California, September 2005.

[Luo1] Luo J., Hubaux J-P., ***“Joint Mobility and Routing for Lifetime Elongation in Wireless Sensor Networks”***, *IEEE INFOCOM 2005*, Miami, March 2005.

[Mai1] Mainwaring A., Polastre J., Szewczyk R., Culler D. and Anderson J. ***“Wireless Sensor Networks for Habitat Monitoring”***, In ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'02), Atlanta, GA, Sept. 2002.

[Man1] Manjeshwar A., Agrawal D.P., ***“TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks”***, Proceedings of the 15th International and Distributed Processing Symposium, 2001.

[Man2] Manjeshwar A., Agrawal D.P., ***“APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks”***, Proceedings of the International and Distributed Processing Symposium, 2002.

[Mos1] Moses R., Krishnamurthy D., Patterson R., ***“A self-localization method for wireless sensor networks”***, *EURASIP Journal on Applied Signal Processing*, 2002.

[Ngu1] Nguyen X., Jordan M. I., Sinopoli B., “*A kernel-based learning approach to ad hoc sensor network localization*”, ACM Transactions on Sensor Networks, Vol. 1, No. 1, pp. 134–152, August 2005.

[Nic1] Niculescu D., Nathi B. “*Ad hoc Positioning System (APS)*”, INFOCOM 2003, San Francisco, CA.

[NiS1] Ni S.Y., Tseng Y.C., Chen Y.S., Sheu J.P., “The broadcast storm problem in a mobile ad hoc network”, Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'99), 1999.

[Rag1] Raghavendra C.S., Sivalingham K.M., Znatti T.F., “*Wireless Sensor Networks*”, Kluwer Academic Publishers, 2004.

[Sam1] Samal S., “*Mobility Pattern Aware Routing in Mobile Ad Hoc Networks*”, MSc. dissertation, Virginia Polytechnic Institute and State University, May 2003.

[Sav1] Savvides A., Han C.C., Strivastava M.B., “*Dynamic fine-grained localization in Ad-Hoc networks of sensors*”, Proceedings of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2001), Rome, Italy, July 2001.

[Sch1] Schurgers C., Srivastava M., *"Energy Efficient Routing in Wireless Sensor Networks"*, MILCOM 2001.

[Sor1] Soro S., Heinzelman W.B., *"Prolonging the Lifetime of Wireless Sensor Networks via Unequal Clustering"*, Proceedings of the 5th International Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (IEEE WMAN '05), April 2005.

[Til1] Tilak S., Abu-Ghazaleh N. B., and W. Heinzelman, *"Infrastructure tradeoffs for sensor networks"*, in Proceedings of the ACM International Workshop on Wireless Sensor Networks and Applications Workshop, 2002.

[Tur1] TurtleNet - University of Massachusetts Amhersts,
<http://prisms.cs.umass.edu/dome/index.php?page=turtlenet>

[Vya1] Vyas N., Mahgoub I., *"Location and Mobility Pattern Based Routing Algorithm for Mobile Ad Hoc Wireless Networks"*, 2003 International Symposium on Performance Evaluation of Computer and Telecommunication Systems.

[Vir1] Virrankoski R., Savvides A., *"TASC: Topology Adaptive Clustering for Sensor Networks"*, Proceedings of the Second IEEE International Conference on Mobile Ad-Hoc and Sensor Systems, MASS 2005, Washington DC, ENALAB Technical Report 110401, November 2005.

[Wan1] Wang K., Abu-Ayyash S., T.D.C. Little, Basu P., "***Attribute-Based Clustering for Information Dissemination in Wireless Sensor Networks***", Proc. 2nd Annual IEEE Communications Society Conf. on Sensor and Ad Hoc Communications and Networks (SECON 2005), Santa Clara, CA, September 2005.

[Yem1] Ye M., Li C., Chen G., Wu J., "***EECS: An Energy Efficient Clustering Scheme in Wireless Sensor Networks***", National Laboratory of Novel Software Technology, Nanjing University – China, Department of Computer Science and Engineering, Florida Atlantic University, USA, 2205.

[Yoo1] Yoon S., Shahabi C., "***Exploiting Spatial Correlation Towards an Energy Efficient Clustered AGgregation Technique (CAG)***", IEEE International Conference on Communications.

[Yoo2] Yoon S., Shahabi C., "***An Experimental Study of the Effectiveness of Clustered AGgregation (CAG) Leveraging Spatial and Temporal Correlations in Wireless Sensor Networks***", submitted to ACM Transactions on Sensor Networks. USC (University of Southern California) Computer Science Department Technical Report 05-869.

[You1] Younis O., Fahmy S., "***HEED: A hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks***", IEEE Transactions Mobile Computing, June 2004.