Self-Organization and Management of Wireless Sensor Networks

By

Muhammad Asim B.Sc., M.Sc

A thesis submitted in partial fulfillment of the requirements of Liverpool John Moores University for the degree of Doctor of Philosophy in Computer Science

School of Computing and Mathematical Sciences
Liverpool John Moores University, U.K.

2010

ACKNOWLEDGMENTS

I would like to take the opportunity to pay thanks to the people who guided and supported me throughout my PhD. Without their support and contribution, successful completion of my research would not have been possible.

I would like to express my gratitude to my first supervisor, Professor Madjid Merabti for his continuous support, encouragement, guidance and invaluable suggestions during this research. I would also like to say special thanks to my second supervisor Dr Hala Mokhtar for, her numerous helpful discussions and her valuable time to help me with various stages of my project, in particular, giving valuable suggestions and comments on my work.

My thanks also go to the researchers, administration staff and technicians in the School of Computing and Mathematical Sciences at Liverpool John Moores University for their support over the past years.

I would like to show my great appreciation to my beloved parents and wife for their support and guidance during all these years.

Last but not least, my father-in-law, Dr. Fazal Rehman, and my mother-in-law, without them, the work would not have been possible. Thus, I also want to thank them for their support and encouragement.

I dedicate this thesis to my son Mustafa Asim.

ABSTRACT

Wireless sensor networks (WSNs) are a newly deployed networking technology consisting of multifunctional sensor nodes that are small in size and communicate over short distances. These sensor nodes are mainly in large numbers and are densely deployed either inside the phenomenon or very close to it. They can be used for various application areas (e.g. health, military, home). WSNs provide several advantages over traditional networks, such as large-scale deployment, high-resolution sensed data, and application adaptive mechanisms. However, due to their unique characteristics (having dynamic topology, ad-hoc and unattended deployment, huge amount of data generation and traffic flow, limited bandwidth and energy), WSNs pose considerable challenges for network management and make application development nontrivial. Management of wireless sensor networks is extremely important in order to keep the whole network and application work properly and continuously. Despite the importance of sensor network management, there is no generalize solution available for managing and controlling these resource constrained WSNs. In network management of WSNs, energy-efficient network self-organization is one of the main challenging issues. Self-organization is the property which the sensor nodes must have to organize themselves to form the network. Self-organization of WSNs is challenging because of the tight constraints on the bandwidth and energy resources available in these networks. A self organized sensor network can be clustered or grouped into an easily manageable network. However, existing clustering schemes offer various limitations. For example, existing clustering schemes consume too much energy in cluster formation and re-formation.

This thesis presents a novel cellular self-organizing hierarchical architecture for wireless sensor networks. The cellular architecture extends the network life time by efficiently utilizing nodes energy and support the scalability of the system. We have analyzed the performance of the architecture analytically and by simulations. The results obtained from simulation have shown that our cellular architecture is more energy efficient and achieves better energy consumption distribution. The cellular architecture is then mapped into a management framework to support the network management system for resource constraints WSNs. The management framework is self-managing and robust to changes in the network. It is application-co-operative and optimizes itself to support the unique requirements of each application. The management framework consists of three core functional areas i.e., configuration management, fault management, and mobility management. For configuration management, we have developed a re-configuration algorithm to support sensor networks to energy-efficiently re-form the network topology due to network dynamics i.e. node dying, node power on and off, new node joining the network and cells merging. In the area of fault management we have developed a new fault management mechanism to detect failing nodes and recover the connectivity in WSNs. For mobility management, we have developed a two phase sensor relocation solution: redundant mobile sensors are first identified and then relocated to the target location to deal with coverage holes. All the three functional areas have been evaluated and compared against existing solutions. Evaluation results show a significant improvement in terms of re-configuration, failure detection and recovery, and sensors relocation.

Table of Contents

List of Abbreviations

ADC	Analogue Digital Converter
ALERT	Automated Local Evaluation in Real-Time
CDS	Clustered or Connected Dominating Set
CH	Clusterhead
CPA	Cluster Policy Agent
CPU	Central Processing Unit
GAF	Geographic Adaptive Fidelity
GTSNETS	Georgia Tech Network Simulator
ISM	Industrial, Scientific and Medical band
ISO	International Standards Organization
LCA	Linked Cluster Algorithm
LPA	Local Policy Agent
MIB	Management Information Base
MSC	Mobile Switching Centre
NM	Network Management
NMS	Network Management System
PC	Personal computer
PDA	Personal Digital Assistant
PM	Policy Manager
QoS	Quality of Service
SNMP	Simple Network Management Protocol
TOM	Telecom Operation Map
WBAN	Wearable Wireless Body Area Network
WCA	Weighted Clustering Algorithm
WSNs	Wireless Sensor Networks

List of Figures

List of Tables

1. Introduction

Humans always invent new technologies according to their needs to bring more flexibility to their daily lives. The history of networking is a good example of how computer networking has become more efficient and flexible day by day, e.g. the evolution from wired networks to wireless networks to bring more amenities and flexibility to users. Correspondingly, WSNs are a newly developed networking technology consisting of sensor nodes that are small in size, low-power, low-cost, and multifunctional. These tiny sensor nodes consist of sensing, data processing and communicating components, and communicate untethered over short distances.

A wireless sensor network is a new breed of sensory system and is often referred to as smart sensors. The size of a sensor node may vary, depending on the actual needs of the application. It can be equal to the size of a shoe box to a microscopically small particle. Similarly, the cost of a single device may vary from hundreds of euros (for the network of very small but powerful nodes) to a few cents (for large-scale networks made up of very simple nodes). Each node in sensor network is typically equipped with a radio transceiver or any other wireless communication device, a processing unit which can be a small micro-controller, sensing unit, and an energy source, usually an alkaline battery. In some application, a mobilizer is also required to move a sensor node to perform a particular task [Akyildiz 2002, Romer 2004].

There are two main components in Wireless Sensor Networks, Sensor nodes and Sink. Sink in sensor network can be a computer, laptop or a sensor node which would gather all information or data from sensor nodes and send it to users or forward to other networks e.g. ad hoc network or internet etc. In other words the functionality of sink in sensor networks is similar to server in traditional networks with little difference. Almost in all sensor networks data are routed towards the sink (base-station), hops close to that sink become heavily involved in packet forwarding and thus their batteries get depleted rather quickly [Akyildiz 2002, Bharathidasan

2002, Estrin 2001, Estrin 1999, Hariri 2005]. An example of a sensor network is shown in Figure 1.1.

Wireless sensor networks allow users to monitor areas from a long distance using their laptops or PCs. It provides opportunities for close-up observation with much higher fidelity and extends the scope of monitoring. Due to its small size and wireless communication capability, WSN can be placed as close and as dense as necessary to the phenomenon of interest. Also the positions of sensor nodes do not need to be engineered or predetermined, which allows random deployment in inaccessible terrains or disaster relief operations [Akyildiz 2002]. Wireless sensor networks share many common characteristics with existing ad hoc networks but there are also a number of differences that make wireless sensor networks a challenging subject.

The design, implementation, deployment and maintenance of such large scale wireless sensor networks are different from and are more challenging than traditional systems due to factors such as dynamic topology, energy and memory constraints, infrastructure less architecture, and the harsh environment in which wireless sensor networks are deployed.

A typical wireless sensor network consists of low-cost devices having limited memory, computational power, and energy and communication range. A well known sensor hardware platforms such as MICA mote [Anastasi 2004] and Tmote Sky [TmoteSky] are used for both research and commercial deployments. Motes have much lower processor speed, memory, link bandwidth, and energy supply than mobile PCs or PDA. A typical MICA mote, for example, uses an 8-bit, 8 MHz processor, and the comparable Tmote Sky sensor hardware platform employs a 16-bit, 8 MHz processor [Malasri 2008]. These low cost sensor devices have to rely on a limited supply of energy i.e. batteries. Replacing these energy sources in the field is usually not practical, and consequently, a WSN must operate at least for a given mission or as long as possible. Therefore, all the aspects of the node, from the sensor module to the hardware and protocols, must be designed to be extremely energy-efficient. Decreasing the energy dissipation by a factor or two can double the network lifetime.

Wireless sensor networks are also prone to several types of faults. These failures occur, mainly because of energy depletion, connectivity interruptions and environmental obstacles. For instance, a simple fault in sensor node is a fail-stop where a node stops working once it runs out of battery. However, before node completely shut down due to fail-stops, it may operate at a critical battery level where its processor can operate correctly but other components such as flash memory cannot and, thereby producing arbitrary behaviors during sensing or reprogramming. Failures in sensor networks due to energy depletion are continuous and with time increase. This often results in scenarios where a certain part of the network becomes energy constrained and stops operating.

Furthermore, wireless sensor networks can be used where wireline system cannot be deployed (e.g., a dangerous location or an area that might contaminate with toxins or be a subject to high temperature). Such harsh and dynamic environments also lead to different types of faults in wireless sensor networks. Therefore maintenance and control of such systems is essential to ensure efficient use of network resources for appropriate information gathering and processing. In other words, network management of highly dynamic, resource constraints, and complex large scale wireless sensor networks is extremely important and vital in order to keep the whole network and application working properly.

1.1 Wireless Sensor Nodes Hardware

A sensor node has four basic components, *a sensing unit, a processing unit, a transceiver unit and a power unit.* It can have additional components, depending on the nature of application such as a location *finding system, power generator and a mobilizer* [Akyildiz 2002]. Sensing unit is made up of two sub units: sensors and analog-to-digital converters (ADC). The sensors generate analog signals when observing phenomenon are converted into digital signals by the ADC and then fed into the processing unit. The processing unit is associated with a small storage unit and manages procedures that enable the sensor nodes to collaborate with other nodes. A transceiver unit connects the node to the network. Power unit is one of the most important components as it provides power to the nodes. All these sub units may need to be fit in match box size module. The required size may be smaller than a cubic centimeter. Apart from the small size, these nodes must consume extremely low power, operate in high volumetric densities, and have low production cost, be dispensable and autonomous, operate unattended, and be adaptive to the environment.

A common commercial hardware platform consists of processor cum radio boards commonly referred to as "motes". Each mote is a battery-powered device that consists of a power unit, a sensor unit, a two way industrial, scientific and medical radio band (ISM) transceiver unit, a processor that runs TinyOS-based code, an ADC unit, a logger memory capable of storing up to 100,000 measurements. A base station consists of a mote attached to a mote-interface-board that interfaces to a PC via the parallel port. Two types of motes namely, mica2dot and mica2 are shown in Figure 1.3 [Junas 2009].

1.2 Applications of WSNs

WSNs are different from traditional networks and present a new set of properties. Typically the communication structure of a traditional network will remain the same in all its applications while a WSNs structure will change according to its application. WSNs can be classified into two categories according to applications. The first category is that of indoor WSNs and the second is that of outdoor WSNs. Indoor WSNs can be implemented in buildings, houses, hospitals, and factories etc [Cerpa 2004, Gao 2005, Song 2008]. Outdoor WSNs can be implemented for marine, battlefield, soil, atmospheric monitoring; forest fire detection; meteorological or geophysical research; flood detection; bio-complexity mapping of environments; pollution studies; etc [Akyildiz 2002, Cerpa 2001, Piotrowski 2006]. Other applications of sensor networks can be found in smart environments, interactive museums [Cerpa 2004], car theft monitoring [Song 2008], inventory control, vehicle tracking and detection [Rabaey 2000], soil moisture monitoring, salinity level measurement, traffic control and road detection, aircraft and space vehicles to report excessive temperatures, tire temperature and pressure monitors on automobiles, and many others. The following are some WSN projects for different applications, including;

1) **FireWxNet:** It is a multi-tiered wireless system for monitoring weather conditions in rugged wild land fire environments. FireWxNet enables the fire fighter community to measure and view fire and weather conditions over a wide range of locations and elevations within forest fire [Hartung 2006].

2) **WBAN (Wearable wireless body area network):** The WBAN [Jovanov 2005] system consists of inexpensive, lightweight, and miniature sensors that can allow long-term, unobtrusive health monitoring with instantaneous feedback to the user about the current health status and real-time or near real-time updates of the user medical record.

3) **AlarmNet:** The AlarmNet system integrates heterogeneous devices, some wearable on the patient and some placed inside the living space. Together they perform a health-mission specified by a healthcare provider. Data is collected, aggregated, pre-processed, stored, and acting upon, according to a set of system requirements identified [Wood 2006].

4) **VigilNet:** is a real-time WSN, used for military surveillance. The general objective of VigilNet is to alert military command and control unit of the occurrence of interest in hostile region [He 2006].

1.3 Problem Definition

Wireless sensor networks have many challenges and difficulties in terms of limited energy and bandwidth, short life time, harsh environment, dense deployment, frequent faults, dynamic nature, redundancy, mobility, application specific and unattended operations. To address these challenges, protocols and architectures should be scalable, energy efficient and flexible to incorporate the highly dynamic nature of WSN.

The most energy consuming activity of sensor nodes is radio communication. To save energy consumptions two schemes are used: data aggregation and switching of redundant nodes into sleep mode. As sensor nodes are geographically close to each other, there is high correlation between the data they sense. Clustering of sensor nodes is an efficient solution for data aggregation and to control the mode of sensor nodes. However, existing clustering schemes have high cost for clustering and re-clustering. Because of the highly dynamic nature and frequent changes in nodes status, forming and reforming of clusters are needed frequently and thus consume much energy.

We believe that a static clustering scheme that is based upon location rather than any specific set of nodes is the first solution for such a problem. However, this static clustering should be flexible enough to allow frequent changes for both nodes and clusterheads (CHs). It should also be flexible to accommodate the different applications of WSNs (e.g. data aggregation is dependent on application, so the cluster size and aggregation levels should follow the application).

Another essential task for the optimum operations of WSN is the self management of the network. Generally speaking, network management consists of a set of functions and services to monitor network status, detect network faults and abnormalities, manage, control and help configure network components, maintain normal operation, and improve network efficiency and application performance [Sohraby 2007]. The unique characteristics of wireless sensor networks poses several challenges for network management and makes traditional management schemes impractical for wireless sensor networks.

A sensor network comprising tens of thousands of sensor nodes with energy and bandwidth restrictions, deployed in harsh, uncertain and dynamic environments. These conditions require the system to be adaptive in nature to changing connectivity and node failures; unattended operation requires configuration ("node setup" and "network boot up") to be done automatically and repeatedly; untethered for energy and communication requiring maximum focus on energy efficiency. Applications need to reconfigure and adapt themselves based on information scattered over the network. A self-managed WSN must know its environment and the context surrounding its activity and act accordingly. In other words WSNs need network monitoring and controlling or network management in distributed fashion to cope with the large scale of the network. Localized decisions reduce the number of messages exchanged.

The task of developing and deploying management system in environment that contain large number of energy constrained sensor nodes is not trivial. This task becomes more complex due to the physical restriction of the unattended sensor nodes. Despite the importance of wireless sensor network management, there is no generalized solution available for WSN management. WSNs and their application have been considered without considering an integrating management solution [Asim 2010b, Yu 2008].

1.4 Project Aim and Objectives

The overall aim of this thesis is to design a generic and flexible hierarchical architecture for WSNs, and then use the hierarchical architecture to develop a self-management framework that monitors the sensor network with minimum overhead, collect the management data energy efficiently, and can adapt and reconfigure autonomously to cope with changes of node conditions, resources and network environment. Specifically, the objectives of this thesis are:

1) To establish the background related to clustering schemes, management issues and management requirements for WSNs. The literature review shall examine the different clustering and management solutions proposed in the area of wireless sensor networks.

2) To develop a general and flexible self-organizing hierarchical architecture for wireless sensor networks that extends the network life by efficiently utilizing nodes energy and supports the scalability of the system in a densely deployed sensor networks.

3) To map the self-organizing hierarchical architecture into a management framework that would allow the efficient self management process of the network. In this step we should define the managing and management entities, the different management roles, the management policy and the different tasks for managers.

4) To develop a configuration and re-configuration algorithm to provides services i.e. the self-organization and self-configuration of sensor nodes. wireless sensor networks are prone to network dynamics such as node dying, being disconnected, node power on or off, and new nodes joining the network and so the configuration management services should enable nodes to self-reconfigure themselves without knowing anything about network topology in advance.

5) To propose a new fault management scheme based upon the proposed cellular architecture to address sensor nodes failure and connectivity recovery in wireless sensor networks.

6) To develop a mobility management scheme based on the proposed cellular architecture to explore the motion capability to relocate sensors to deal with sensor failure or to respond to new events. We define the problem of sensor relocation and propose a two-phase sensor relocation solution: redundant sensors

are first identified and then relocated to the target location to heal coverage holes in the network.

1.5 Novel Contributions of the Thesis

We have designed and evaluated a new cellular self-organizing hierarchical architecture for wireless sensor networks that is used as a basis to design a self-management framework to monitor the network with minimum overhead, collect the management data and finally manage the network in an efficient way.

In this section, we discuss our main contributions that compose the proposed management architecture:

- **The hierarchical cellular scheme:** We have developed a cellular clustering scheme [Asim 2008a] to partition the network into square shaped cells to extend the network life time. This involves nodes organizing themselves into cells and identifying a leader, or a manager, for each cell. The cellular architecture is designed for a densely populated sensor networks. The proposed grid is then extended to a hierarchical architecture with cells organized in groups in different layers. This design minimizes the communication messages, eliminates the redundancy of transmitted data, and thus conserves energy. Our cell and group formation algorithm consumes less energy as it is based upon the actual or virtual coordinates of the nodes. We simulated our proposed algorithm and compared it to existing work. The scheme shows better results with regards to the life time of the network.

- **A sensor network management framework:** We mapped the cellular architecture into a hierarchical management framework for wireless sensor networks. We have proposed a generic n-tier hierarchical framework for wireless sensor networks. The number of hierarchical levels is based on application type and number of nodes. The management information is collected and processed at each level of hierarchy, and only forwarded to the upper level on request or by some special event. The proposed management framework facilitates the distribution of control over the entire network. It saves energy and reduces network contention by enabling locality of communication. We have identified five main management functionalities for WSN: configuration management; fault management; mobility management; power management; and performance management. This thesis will mainly focus on configuration, fault and mobility management. Security and performance management will not be discussed in this thesis.

- **A configuration management algorithm:** WSNs are highly dynamic because of frequent node dying, node being disconnected, node power on or off, and new nodes joining the network. To address this challenge, we have developed a configuration and re-configuration algorithm [Asim 2010a] to energy-efficiently re-organize the network topology of WSNs. Our algorithm performs network re-configuration and maintenance in a distributed fashion and consumes significantly low energy. Experiments were performed to elucidate the characteristics of the proposed re-configuration mechanism.

- **A new fault management scheme:** We have used our cellular architecture and developed a new fault management scheme for wireless sensor networks [Asim 2008b, Asim 2009]. The purpose of the proposed system is to detect the fault and if possible recover from the faulty state of the network. The grid based architecture permits the implementation of fault detection and recovery in a distributed manner and allows the failure report to be forwarded across cells. The faulty nodes are detected and recovered in their respective cells without affecting overall structure of the network. The proposed failure detection and recovery scheme has been compared to existing related work and proven to be more energy efficient.

- **A new mobility management scheme:** Mobile redundant sensor nodes can move to repair coverage holes caused by node failures or non uniform deployment of sensor nodes. Utilization of redundant mobile nodes plays an important role in prolonging network life time. We define the problem of sensor relocation and propose a two-phase sensor relocation solution: redundant sensors are first identified and then relocated to the target locations. We utilized our hierarchical cellular architecture to quickly locate the closest redundant sensors with low message complexity, and used cascaded movement to relocate the redundant sensor in a timely, efficient and balanced way. Simulation results verify that the proposed solution outperforms existing solutions in terms of relocation time and total energy consumption.

1.6 Thesis Structure

The remainder of this thesis is organized as follow:

In Chapter 2, "Background", we discuss WSN characteristics and their main challenges. We investigate existing clustering schemes and analyze their problems. This chapter explains in detail sensor networks management, including sensor network management design issues, management architectures and management services.

In Chapter 3, "A Hierarchical Cellular Self-organizing Scheme for Wireless Sensor Networks", we explain the proposed cellular hierarchical architecture, the formation of the cellular architecture, the hierarchical clustering design objectives, and then analytical evaluation of the proposed architecture. Finally, performance evaluation has been performed through simulations.

In Chapter 4, "A Self-management Framework for Wireless Sensor Networks", we explain in detail the proposed hierarchical management framework for wireless sensor networks. This chapter describes the management hierarchy, management roles, role assignment and the management process. It also discusses the management functional units of our proposed framework.

In Chapter 5, "Configuration Management of Wireless Sensor Networks", we present a re-configuration algorithm to deal with network dynamics. This chapter describes a detailed description of the re-configuration algorithm to deal with different modes of the sensor node. It describes the cell merging procedure in detail. Performance of the proposed algorithm is evaluated through simulations.

In Chapter 6, "A Fault Management Scheme for Wireless Sensor Networks", we discuss in details the new fault management scheme for wireless sensor networks. This chapter describes the motivation for the design decisions, the different phases, and simulation results to evaluate the scheme performance.

In Chapter 7, "Mobility Management of Wireless Sensor Networks", we present a sensor relocation algorithm to find redundant mobile nodes in the network and relocate them to target location in a timely and energy efficient manner. This chapter provides a detail description of the different phases of proposed sensor relocation algorithm, and evaluate its performance by simulation.

In Chapter 8, "Conclusion and Future Work", we conclude our thesis by summarizing the findings that we have achieved so far. This chapter concludes our PhD project by providing an overall summary, comparison with existing approaches, contribution to knowledge and future plans.

2. Background

A Wireless sensor network is a network comprised of numerous small independent sensor nodes densely distributed over the region of interest for collecting information or monitoring and tracking certain specific phenomena from the physical environment. Each node in a sensor network is typically equipped with a radio transceiver or other wireless communication device, a small micro controller, and an energy source, usually a battery. The sensor nodes self organize to form a wireless network and data from the nodes is relayed to neighboring nodes until it reaches the desired destination for processing. Depending on the nodes geographical positions, their transceiver coverage patterns, transmission power levels, and co-channel interference levels, a network can be formed on the fly without relying on the presence of any fixed network infrastructure. The sensor networks topology changes as sensor nodes migrate, "disappear" (failure or depletion of battery capacity), or adjust their transmission and reception characteristics.

The wireless sensor network has proven to be useful in the different fields of applications, where traditional networks have failed. WSNs offer characteristics like limited power consumption, ability to withstand harsh environmental conditions, ability to cope with node failures, mobility of nodes, dynamic network topology, heterogeneity of nodes, large scale of deployment and handling unattended operations. These unique characteristics enable WSNs to be used in applications such as temperature control, humidity control, vehicular movement, lightening conditions, pressure, soil make up, noise levels, the presence or absence of certain kinds of objects, mechanical stress levels on attached objects, the current characteristics such as speed, direction and size of an object. The ad hoc nature and deploy-and-leave vision make it even more attractive in military applications and other risk-associated applications, such as catastrophe, toxic zones, and disasters [Akyildiz 2002, Bharathidasan 2002, Culler 2004, Lewis 2002].

WSNs represent a relatively new research area that has a large number of complicated research challenges in management, mobility, routing, security and many others. In chapter 1, we discussed the importance of self-organization and self-management in wireless sensor networks. In this chapter, we first discuss some existing clustering solutions for the self-organization of wireless sensor networks. We then present some of the key issues that differentiate wireless network management from that of traditional network management system. We also discuss some design issues and requirements for proposing a new management framework for WSNs. Finally, we present and analyze some existing management solutions for WSNs.

2.1 Classification of Wireless Sensor Networks

In this section we will briefly discuss the different classifications of WSNs.

2.1.1 Homogenous vs. Heterogeneous Sensor Networks

In terms of the component nodes, the sensor network can be classified into two categories: *homogenous sensor networks and heterogeneous sensor networks*. In homogenous sensor networks all sensor nodes are identical in their capabilities and functionalities with respect to the various aspects of sensing, communication and resource constraints. On the other hand, in a heterogeneous sensor networks, two or more different types of nodes with different capabilities execute different functions. For example; some sensor nodes may have larger battery capacity and more processing capability and some may aggregate and relay data; other nodes may only perform the sensing function and not relay data for other nodes in the network. The deployment of homogenous sensor network is simpler and easier, while a heterogeneous network is more complex and its deployment is more complicated because different types of nodes must be dispensed carefully in specified areas [Ilyas 2004].

2.1.2 Static vs. Mobile Sensor Networks

In static sensor networks, there is no motion among communication sensors (the observer and the phenomenon). For example; a group of sensor nodes are spread for temperature sensing, and there is no movement of sensor nodes. In mobile sensor networks, either the sensor nodes themselves, or the phenomenon are mobile [Chen 2007b]. Mobility and its effects on the sensor network have been emerged as an important requirement for wireless sensor networks. Mobile sensor nodes can be used to improve network security and network coverage holes. Mobile nodes can deliver energy to static sensor nodes. However, movement of sensor nodes has many special needs i.e. movement in sensor networks involves communication and can be very expensive in terms of energy. Mobility in WSN would also require network reconfiguration.

2.1.3 Event-based vs. Query-based Sensor Networks

In event-based sensor network applications like forest fire detection, one or more sensor nodes detect an event and report it to a base station or monitoring station. However, in query-based sensor networks applications like inventory tracking in a factory warehouse, sensors remain silent until they received a request from the monitoring station. In both cases, sensor nodes are generally deployed in large numbers-placed mostly random-either close or inside the phenomenon to be studied [Carle 2004].

2.1.4 Flat vs. Hierarchical Sensor Networks

There are two basic sensor network architectures, flat and hierarchical, that specify how sensors are grouped and how sensor information is routed through the network. In flat architecture, sensor nodes have almost the same communication capabilities and resource constraints and the data is routed sensor by sensor. However, in hierarchical architectures, sensor nodes are grouped in clusters and each cluster is represented by a clusterhead node [Lopez 2008]. A clusterhead is responsible for its cluster and may perform different operations i.e. data aggregation and routing. A common example of hierarchical clustering is Leach algorithm [Heinzelman 2000].

2.2 Characteristics & Requirements of WSNs

This section discusses some unique characteristics and requirements challenges of WSNs, which need to be taken into account when designing protocols and architectures for WSNs. These include production cost; transmission media; limited power consumption; hardware constraints; harsh environment; dense deployment and scalability, node failure and fault tolerance, unattended operation and self-management.

2.2.1 Production Cost

A sensor network consists of a large number of sensor nodes and the cost of single node justifies the overall cost of the network. Thus, the cost of each sensor node has to be kept low to justify its deployment over traditional sensors.

2.2.2 Transmission Media

Nodes in a sensor network are linked by a wireless medium. These links can be formed by radio, infrared, or optical media. To support global operation of these networks, the chosen transmission medium must be available worldwide.

2.2.3 Limited Power

A sensor node is a microelectronic device, equipped with a minimum power source. In some application scenarios, replenishment of power resource is impossible. Therefore sensor node life time is strongly dependent on the battery life time. Expiration of a battery causes failure of the sensor node, which on the other hand causes significant topology changes and might require rerouting of packets and reconstruction of the network. That is the reason where researchers are still focusing on the design of power aware algorithms and protocols for sensor networks.

2.2.4 Hardware Constraints

The sensor nodes face a number of hardware constraints. Due to its small size a sensor node has limited computational capabilities and is built up with limited memory storage. Radio communications is a major energy consumer. The limited hardware capabilities compel to develop algorithms which do not require immense computational and storage resources.

2.2.5 Harsh Environment

Sensor nodes are densely deployed either inside the phenomenon or very close to it. Therefore a sensor node has to operate unattended in remote geographic areas. They may be working in the interior of large machinery, at the bottom an ocean, in a biologically or chemically contaminated field, in a battlefield beyond the enemy lines, and in a home or large building [Akyildiz 2002, Culler 2004].

2.2.6 Dense Deployment and Scalability

Sensor nodes may be deployed in large numbers to study a phenomenon. The number may increase up to millions, depending on the nature of the application. Thus any scheme proposed for sensor networks must address the scalability issue. The sensor network density can be calculated as flow [Akyildiz 2002]:

$$\mu(R) = (N\pi R^2)/A$$

Where N is the number of scattered sensor nodes in region A, and R the radio transmission range. Basically, $\mu(R)$ gives the number of nodes within the transmission radius of each node in region A.

2.2.7 Node Failure and Fault Tolerance

Fault tolerance is the ability to sustain sensor networks functionalities without any interruption due to sensor nodes failure. Sensor nodes may fail or be blocked due to environmental interference, lack of power or physical damage. A failure of any sensor node should not affect the overall performance of sensor network [Akyildiz 2002, Hariri 2005]. That is why reliability or fault tolerance is an importance issue and algorithms and protocols must be designed to address the level of tolerance required by sensor networks. In [Agre 2000] the fault tolerance (or reliability) of a sensor node is modeled by the flowing equation: $R_k(t) = \exp(-\lambda_k t)$ where λ_k and t are the failure rate of a sensor node k and the time period, respectively. In the same time, fault tolerance is related to the environment where the sensor network is deployed. For example, the fault tolerance needed in tracking of animals movements is not the same needed in battlefield surveillance.

2.2.8 Unattended Operation and Self-management

Wireless sensor networks are usually deployed in the harsh operational environment where the physical presence of human administrators is impractical. Applications and systems of these networks are thus expected to operate with the minimum aid or supervision. Thus, WSN must have self-management capabilities to manage the network resources, and it should be robust to changes in network states while maintaining the quality of services.

From the characteristics of WSNs, it is clear that there are large numbers of challenges that should be addressed to improve performance, life time and reliability of WSN. Among these challenges we address the scalability, dynamity, self-organization and self-management issues.

2.3 Self-Organization of Wireless Sensor Networks

In WSNs, large numbers of sensor nodes are densely deployed either inside the phenomenon or very close to it. Also the position of sensor nodes does not need to be engineered or predetermined, which allows random deployment in inaccessible terrains or disaster relief operations. Another unique feature of sensor networks is the on board processing and co-ordination. Instead of sending the raw data to the nodes responsible for fusion, they use their processing abilities to carry out simple computation and transmit only the required and partially processed data. On the other hand, this also means that sensor networks protocols and algorithms must possess self-organizing capabilities. Self-organization is the process of autonomous formation of connectivity, addressing and routing structures.

Self-organization (or self configuration) has been a significant research topic in wireless networks. Self-organization involves abstracting the communicating entities into an easily controllable network infrastructure. Clustered or connected dominating set (CDS), grid, tree, or mesh based organization are key terms in self organization. A self-organized wireless node can be grouped or clustered into an easily manageable network infrastructure [Kochhal 2003].

Grouping sensor nodes in clusters has been widely persuaded by the research community in order to achieve scalability in wireless sensor networks. Network nodes are first grouped into clusters and then a leader (clusterhead) is selected in each cluster to represent the cluster at a higher level. The same clustering scheme can be applied to the cluster leaders to form a hierarchy [Yu 2008]. A clusterhead is the leader of the cluster, and is often required to organize activities in the network i.e. data aggregation.

Apart from supporting network scalability, clustering offers numerous advantages [Abbasi 2007]. It can localize the route setup within the cluster to reduce the size of routing table stored at the individual node. It can also conserve communication bandwidth since it limits the scope of inter-cluster interaction to clusterheads and avoids redundant message exchange among sensor nodes. Clustering algorithms vary in their objectives and are set in order to facilitate meeting the application requirements. There are several key attributes that a designer must carefully consider before developing any clustering scheme for wireless sensor networks. These will be considered in the section.

2.3.1 Clustering Design Philosophy

In this section, we will summarize some important requirements for clustering in wireless sensor networks.

- **Cost of cluster formation:** Although clustering plays an important role in organizing sensor network topology, there are many resources such as communication and processing tasks needed in the creation and maintenance of the clustering topology. Such costs as the required resources are not being used for data transmission or sensing tasks. Also, due to highly dynamic nature of WSNs, forming and reforming of clusters are needed frequently and thus consume much energy. Therefore, it is very important that a clustering scheme should not consume much energy in clustering formation and re-formation.

- **Selection of clusterheads and clusters:** The clustering concept offers various benefits for WSNs. However, when designing for a particular application, designer must carefully examine the formation of clusters in the network. Depending on the application, certain requirements for the number of nodes in a cluster or its physical size may play an important role in its operation. This prerequisite may have an impact on how clusterheads are selected in the application [Dechene 2006].

- **Uniform energy consumption:** Transmission in WSNs is more energy consuming compared to sensing, therefore the clusterhead which performs the function of transmitting the data to the base station consumes more energy compared to the rest of the nodes. Clustering schemes should ensure that energy dissipation across the network should be balanced and the clusterhead should change when its energy drops below a threshold value.

- **Data aggregation:** One major feature of WSNs is the ability for data aggregation to occur in the network. In a densely populated network there are often multiple nodes sensing similar information. Data aggregation allows the differentiation between raw sensed data and useful information. In-network processing makes this process possible and now it is fundamental in many sensor network schemes. As such, the amount of data transferred in the network should be minimized.

- **Network dynamics:** Aside from the few schemes that utilize mobile sensors, most of the network architectures assume that sensor nodes are stationary. Sometimes it is extremely important for an application to support the mobility of sensor nodes. Node mobility would make clustering very challenging since node membership will dynamically change, forcing clusters to evolve overtime [Abbasi 2007].

- **Network coverage:** Node deployment in WSNs is either uniform or random depending on the application. In fixed network deployment, the network is deployed on predetermined locations whereas in random deployment the resulting distribution can be uniform or non uniform. In such case, it is important for a clustering scheme to ensure an entire area is fully covered and each node belongs to one cluster.

2.3.2 Existing Clustering Schemes

Many clustering algorithms in various contexts have been proposed for wireless sensor networks [Amis 1999, Chatterjee 2002, Chen 2007a, Gupta 2003b, Lin 2000, Zhang 2003]. Clustering algorithms differ with respect to the metrics they use for cluster control such as energy, hops, life time calculations, distance from the clusterhead and also the type of controls such as centralized or decentralized [Venkataraman 2005]. A survey on clustering algorithms in sensor networks can be found in [Abbasi 2007, Dechene 2006]. Clustering schemes can be classified into heuristics schemes, weighted schemes, hierarchical schemes and grid-based schemes [Dechene 2006].

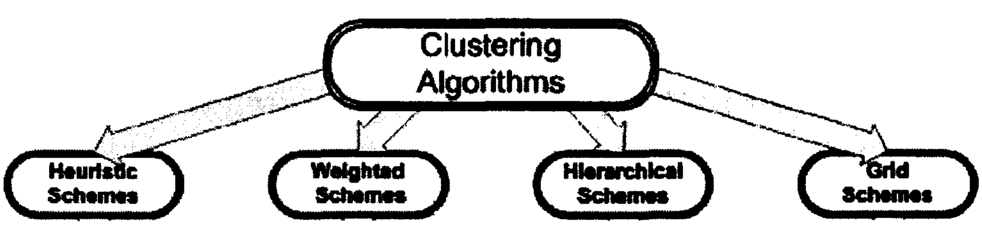

Figure 2.1: Classification of clustering schemes

1) **Heuristic clustering**

Heuristic clustering scheme usually has one or both of the following goals in solving a problem:

- Finding an algorithm with reasonable run-time (time needed to set up clusters is affordable); and/or
- With finding the optimal solution

This means that heuristic algorithms leads to reasonable performance and is not based on particular metrics [Dechene 2006]. Linked Cluster algorithm (LCA) [Baker 1981] is a heuristic clustering scheme that was initially developed for wired sensors, but later implemented for wireless sensor networks. LCA mainly focuses on forming an efficient network topology that can handle the mobility of the sensor nodes in the network. In this algorithm, each node is assigned a unique identifier and the selection of the CH based on the highest identity among all nodes within 1-hop. The main drawback of LCA is that it may elect an excessive number of clusterheads. This limitation of LCA was enhanced by modifying LCA to form Linked cluster algorithm 2 (LCA2) [Ephremides 1987], which selects the node with the lowest ID among all nodes that is neither a CH nor is 1-hop of the previously selected CHs. LCA2 consists of covered and non-covered nodes. A node with a CH as a neighbor is considered covered. CHs are selected starting from the node having the lowest ID among non-covered neighbors.

2) **Weighted clustering**

These clustering schemes rely on weights to select CHs. Weighted clustering algorithm (WCA) [Chatterjee 2002] select clusterheads based on the number of neighboring nodes, transmission power, mobility and battery life. It uses weights associated with nodes to elect CHs. A node with the highest weight among its one-hop neighbor is elected as a CH. These weights are generic and can be defined based on the application. When a node looses connection with its CH, the election procedure is invoked to find a new clustering topology. This re-election approach is energy consuming and not suitable for energy constrained wireless sensor networks.

3) Hierarchical clustering

Leach [Heinzelman 2000] is one of the most popular clustering algorithm, and a number of clustering algorithms are derived from this scheme. The main objective of leach is to guarantee a certain network life time while minimizing energy consumption. This is achieved by ensuring that all nodes die at the same time by rotating the role of clusterhead periodically among the nodes of the cluster. The disadvantage of Leach protocol is its random selection of clusterheads. In random selection of clusterhead, there exists a probability that a node with low energy is selected as a clusterhead. When this node dies, the whole cluster becomes dysfunctional. Also, Leach protocol offers no guarantee about uniform placement of CHs in a system. Therefore, there is the possibility that the elected CHs will be concentrated in one part of the network. Hence, some sensor nodes will not have any CHs in their vicinity. Furthermore, rotating the role of clusterhead consumes much energy.

An extension of Leach protocol has been proposed in [Heinzelman 2002]. It uses a centralized approach for the formation of clusters. The algorithm begins from the base station where each node sends its location information along with their energy level to the base station. The clusterheads are selected randomly but the base station ensures that a node with less energy does not become a clusterhead. The main disadvantage of this approach is that it is not feasible for large networks because the nodes which are far away from the base station may have difficulty in sending their status to the base station.

The clustering architecture proposed in [Chen 2007a] is based on hierarchical management of sensor nodes. This study presents an algorithm for self-organization mechanism of high-level nodes, contesting member nodes by multi-hop to form hierarchical clusters, and applying the '20/80 rule' to determine the ratio of headers to member nodes. Clusterheads or high level nodes periodically broadcast a 'cover request' (CREQ) periodically. CREQ is delivered to all the sensor nodes in the network. The low-level nodes select the clusterhead using the minimum-hop-count.

Load balance clustering has been proposed to balance the load on clusterheads [Gupta 2003b]. It incorporates two types of nodes: gateway nodes which are less energy constrained nodes (clusterheaders) and sensor nodes which are energy constrained. The less energy constrained gateway nodes maintain the state of sensors as well as multi-hop route for base station. The gateway nodes are less energy constraint and static than the rest of the network nodes and they are also fixed for the life of the network. Therefore, sensor nodes close to the gateway node die quickly while creating holes near gateway nodes. Also, when a gateway node die, the cluster is dissolved and all its nodes are reallocated to other healthy gateways. This consume more time as all the cluster members are involved in the recovery process.

In [Banerjee 2001] authors proposed a multi-tier hierarchical clustering algorithm. In the proposed scheme, any node in the network can initiate the cluster formation process. Initiator with least node ID will take the precedence, if multiple nodes started cluster formation process at the same time. The algorithm is based on two phases: Tree discovery and cluster formation. The tree discovery phase is basically a distributed formation Breadth-First-Search tree rooted at the initiator node. The cluster formation phase starts when a sub-tree on a node crosses the size parameter, k. It considers logical radius of clusters instead of geographical radius, which can reduce wireless transmission efficiency because of large geographical overlaps between clusters.

***4)* Grid clustering**

The following are routing protocols rather than clustering schemes. However, we present them here because they have some similarity to our work.

GROUP [Yu 2006] is a grid clustering routing algorithm, in which one of the sinks (termed the primary sink), dynamically and randomly builds grid clusters. In GROUP, all sensor nodes are divided into several clusters dynamically. One node is selected as the clusterhead in each cluster. The CHs are arranged in a grid-like manner. This algorithm is developed purely for routing purpose. The data queries will be transmitted from sinks to all nodes via clusterheads.

Geographic Adaptive Fidelity or GAF [Xu 2001] is a localized, gird-based routing algorithm that concentrates on energy consumption to increase network life. It is specifically proposed for routing in WSNs. GAF uses location information to divide the network area into virtual grids, and each node associates itself with this virtual grid. A virtual grid is defined such that, "for two adjacent grids A and B, all nodes in A can communicate with all nodes in B and vice versa". In each grid, nodes determine which of them will sleep, and which nodes will remain active for a certain period of time. It is also important to balance the network load and therefore, sleeping nodes turn on their radios periodically and trade places with the active nodes. In GAF, nodes are in sleeping, discovery or active state. All nodes begin in discovery state where they send and receive discovery messages to find other nodes in the grid. After some time nodes enter into active state and sets a timer as for how long it will stay active. Once this timer expires, nodes will then go back to discovery state. If a node determines that it is a redundant node for the routing protocol, it will enter into sleeping state for a specific period of time. Nodes are ranked according to their remaining energy level. GAF achieve a good load balancing by employing node ranking strategy. As in [Frye 2007], the main drawback of GAF is that it guesses at connectivity instead of directly measuring, and thus requiring more nodes to remain active than may be necessary.

Significant attention has been paid to clustering strategies and algorithms in wireless sensor networks; however there is still much to be done. In this section we surveyed existing clustering scheme and discussed their advantages and disadvantages. Most existing schemes consume significant energy in cluster formation and re-formation and do not minimize energy associated with clusterhead selection process. Heterogeneous clustering scheme requires clusterheads to be carefully placed in the network to contribute towards the performance of the application. This is not feasible for applications that need the random deployment of sensor nodes in a harsh environment, where human intervention is not possible. Most clustering schemes assume that the network is based on stationary sensor nodes and does not support node mobility. We therefore contend that there is still a need of a new clustering scheme to address all the problems in existing clustering schemes for wireless sensor networks. After choosing a perfect clustering scheme, the next step is to provide efficient management for WSNs.

2.4 WSN Management

The second area that we address in this thesis is the self-management of WSNs. In this section we start by defining network management in general, and survey some traditional management schemes. We then define and analyze existing solutions for WSN management.

2.4.1 Network Management

A computer network generally consists of three components: physical devices, including links (wireless or wired link), network nodes (hub, bridge, switch, or router), terminals and servers; protocols; and information that is being carried, including applications. The physical devices collaborate with network protocols to forms the underpinning support for the applications. Protocols are used to transport information efficiently, preferably in a correct, secure, reliable, and understandable manner. They consist of a set of software residing at physical devices. However, the physical devices and protocols are not sufficient to support effective operation of network communication. Network management (NM) tools and techniques are also required to help provision of network services and ensure cooperation of entities in the network [Sohraby 2007]. Generally speaking, network management consists of a set of functions to monitor network status, detect network faults and abnormalities, manage, control and help configure network components, maintain normal operation, and improve network efficiency and network performance. To perform these tasks, a managing entity collects real-time information through an agent, analyzes the information, and applies control based on the information.

In other words, network management is the process in which different network entities (which represent managed devices) provide information about their state to a managing entity, which then reacts to this information by executing one or more actions such as logging, reset or repair. Managed network devices may send information to their manager on their own, either periodically or when certain triggers are fired such as exception, or upon instructed by the manager [Bapat 2006]. There are various reasons that a network must have an efficient management system such as [Sohraby 2007]:

1) A network normally consists of many heterogeneous devices and software entities, and some may fail to operate. It is then the responsibility of network management to determine when, where and why the fault had occurred and how to restore these entities.

2) A distributed system optimization requires NM to collaborate in the process. For example, in some networks, congestion control through admission control, by changing routes, or through device upgrade occurs by NM functions.

3) For most networks, NM functions can be used to collect and analyze the behavior of user interaction during network interface, which is sometimes very important in planning the long-term evolution of network capacity and its performance.

2.4.1.1 Network Management Model

Conceptually, management systems are based on a simple model. In this model, management is interaction/cooperation between two major entities: the managing entity and the managed entity as shown in figure 2.2. The entity represents a management platform, a management system, and/or a management application. The managed entity represents the managed resources.

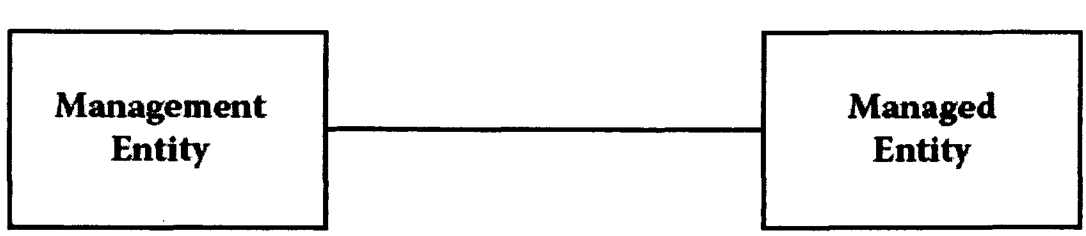

Figure 2.2: Management basic model

In order to communicate with the managed resources, there is a need of an intermediate component called management agent or managed agent. The manager–agent model is very common, and is used in describing the interaction between the management entity and the managed entity at a high level as shown in figure 2.3.

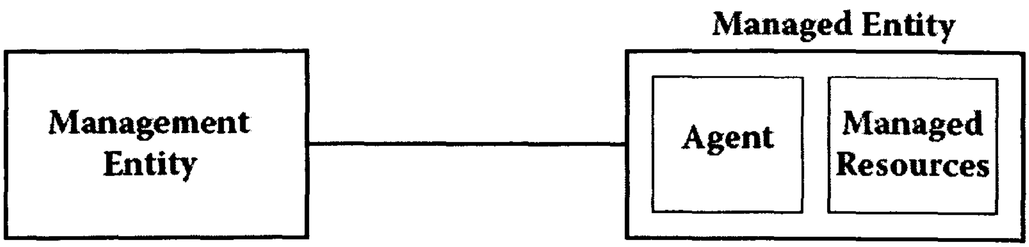

Figure 2.3: Manager-agent model [Morreale 2009]

Managers are software systems, responsible for the communication with managed entity through agents, to retrieve information about their state, the storage of the obtained information and own activities in adequate databases, and the provision of the stored management information to administrators (through user interfaces).An Agent is a software module of a certain network component (e.g. a bridge / router) entrusted with the supervision, configuration, and control of the entities of resources and the connection with the related manager and the transmission of the requested information to the manager. The network components are modeled as managed entities, and can be accessed via a virtual information database, called the management information base (MIB) [Meer 2003].

The management communication is based on request-reply paradigm. The manager will request from the agent specific management information about the managed entity; and the managed entity, through the agent, will reply with a message containing the information requested. The request-reply mechanism is considered a synchronous communication mechanism, i.e. the manager expects an answer from the agent in a limited time frame before taking any action. If the reply message is not received, a retransmission request is initiated by the manager. Another communication mechanism between the agent and manager is called notification. The notification is an asynchronous mechanism initiated by the agent that communicates important changes to the manager in case of managed resource status changes and requires either manager attention or intervention [Morreale 2009].

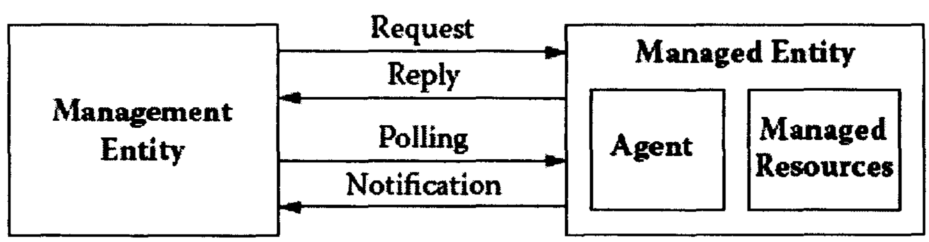

Figure 2.4: Manager-agent communication model [Morreale 2009]

2.4.1.2 Traditional Networks Requirements and Management Schemes

Some common traditional management requirements that have been discussed at [Morreale 2009] are presented as follows:

- Ability to monitor and control end-to-end network and computing systems components.
- Remote access and configuration of managed resources.
- Ease of installation, operation, and maintenance of the management systems and their applications.
- Secure management operations, user access, and secure transfer of management information.
- Ability to report meaningful and important management-related information.
- Real-time management and automation of routine management operations.
- Flexibility regarding systems expansion and ability to accommodate various technologies.
- Ability to back up and restore management information.

Some of the existing traditional management schemes are discussed below:

1) **Simple network management protocol**

The simple network management protocol (SNMP) [Case 1990] is in broad use today. It consists of three major components: a network management system (NMS), managed elements and agents. NMS is a set of application that control and monitor managed elements. The managed elements are the network devices that are required to be managed. Examples of managed elements include routers, switches and hosts. SNMP agents run on each managed element, which collects and stores management information in Management information base. Agents translate the management information into a form compatible with SNMP MIB. NMS has the ability to request management information from the network agents and present to the network manager. Even though SNMP is limited in terms of being reactive as opposed to proactive and some scalability concern , it remains the most widely used network management protocol because it enables the network manager to collect real-time information, analyze the information, find and solve network problems , and plan for network growth. However, it only manages network elements and does not support network-level management.

2) **Telocom operation map**

The Telocom operation map (TOM), proposed by TeleManagement Forum [TOM 2000], is a management model that provides a layered architecture for management and administration. Each layer has a different management function and set of management objects. TOM can be used to manage most tasks, from the underlying physical network element to the entire network, as well as the services provided.

Neither SNMP nor TOM is designed particularly for wireless sensor networks. However, one can utilize the simplicity of SNMP and the layered framework of TOM to design effective and efficient network management architecture for wireless sensor networks [Sohraby 2007].

2.4.1.3 Functional Areas of Traditional Network Management

Traditional wired network management includes five fundamental areas as identified by the international standards organization (ISO) [Lee 2006a]. These five areas are:

- Fault management
- Configuration management
- Accounting management
- Performance management
- Security management

Configuration management is the process of monitoring and controlling network devices. Usually an inventory of all network devices along with their current configuration is maintained. This is achieved by collecting information from all the devices on a periodic basis, either manually or automatically. Performance management is a very important part of the network management model and is used to ensure that network performance remains at acceptable level. It consists of accessing and monitoring network devices and links in order to determine utilization and gather regular performance data such as packet loss rate, link utilization, network response time, and so forth. Performance monitoring is an importance step in identifying problems before they occur.

Security management deals with control access to network resources and includes managing network authentication, auditing, and authorization. The main goal of security management is controlling access points to critical or sensitive data that is stored on the network devices. Fault management is used to detect, log and alert system administrators of problems that might affect the system operations. The purpose of this area of network management involves finding the problem, isolating the problem, and fixing the problem if possible. Fault should be reported in some manner such as an email message to the network administrator, log file, or an alert on the network management system. Accounting management monitors and assesses the usage of data and/or resources for the purpose of billing. This information can be used to generate metrics and quotes.

2.4.1.4 Network Monitoring

Networking monitoring is the information collection process of network management. Network monitoring is used to collect useful information from various parts of the network so that the network can be managed and controlled using the collection information. Network monitoring is an essential component of managing network and is important for detecting network anomalies. Anomaly detection is the process of determining when system behavior has deviated from normal behavior.

Network monitoring can be broken down into three stages. The first of these stages involves in the process of collecting information about the network. In the second stage, collected information is transformed into useful detection metrics. These new metrics should capture information about the behavior of the network. Finally, the third stage of network monitoring assesses network behavior in order to determine abnormal events. This function is also termed as anomaly detection [Bunke 2006].

As discussed in [Bunke 2006, Cecil 2006], network monitoring can be categorized into active and passive monitoring. The active monitoring approach relies on the capability to inject test packets into the network to collect measurements between at least two end points in the network. The traffic generated by such testing is in addition to the usual traffic load on the network. As such it creates extra load traffic in the network. Active monitoring techniques use tools such as ping to measure delay and loss of packets in the network, and are often used for the characterization of the internet, since they can be used when administrative control of the network is not centralized, and hence direct access to network elements is not possible. Conversely, passive monitoring does not inject traffic into the network or modify the traffic that is already on the network. The passive approach uses devices to watch the traffic as it passes by. It gauges the traffic flow in and out of a single device and can examine encapsulated headers to derive behavior related to the network layer and above. Any packet sniffing program can be used to achieve passive monitoring. Devices such as routers containing SNMP agents is the most commonly used passive monitors.

Although the goals of management in wireless sensor networks are similar to that of traditional networks such as internet or cellular networks, there are several distinctions between the two due to the unique characteristics of wireless sensor networks. In traditional network systems, network elements are installed and configured by the technicians. These networks are designed to accommodate a diversity of applications. Technicians manage network components and resources, and make sure that the network provides all the desirable services. Further, the network follows a well-established plan to utilize network resources. The overall goal of traditional network management is to promote productivity of network resources and maintain the quality of the service provided. Nevertheless, many of the traditional management concepts discussed above are applicable for WSNs management.

2.4.2 Sensor Networks Management

A lot of existing traditional network maintenance and management designs have been proposed in the context of wired network, not only for the internet, but also for cellular networks, where the connection from the base station to the mobile switching centers (MSCs) are wired. These networks are provisioned with enough resources to support the network information gathering required for management. Furthermore, these networks are perpetually powered so that they do not have to worry about energy or network life time. Accordingly, the traditional network maintenance and management approaches are impractical for resource constrained wireless sensor networks.

WSNs management are mainly concerned with monitoring and controlling node communication in order to optimize the efficiency of the network, ensure network operates properly, maintain the performance of the network, and control large numbers of nodes without human intervention [Lee 2006a]. A sensor network management system collects different information from the network (i.e. battery levels, communication power, network topology, link state and the coverage) and can perform a variety of management control tasks such as: switching node on/off (power management), controlling wireless bandwidth (traffic management), and performing network reconfiguration in order to recover from a node failure. Furthermore, a sensor network management system should support self - forming, self – organize, and especially self-configure in the event of failures. A discussion on WSN management has been presented in [Ruiz 2004b] and [Wang 2003a].

One primary goal of WSN management is to be autonomous. This term derives from the human autonomic nervous system, which control key functions without conscious awareness or involvement. An autonomic system is composed of interrelated autonomic elements. These elements are responsible for the management of hardware and software resources that build the IT infrastructure and autonomic managers that supervise and control these resources. The autonomic manager provides self-management services through monitoring, analyzing, planning and executing modules. WSN management must be autonomic, and must be capable of self-configuration, self-healing, self-organization and self optimization.

2.4.3 Sensor Network Management Design Issues

The unique characteristics and restrictions of WSNs make the management approach different enough from the traditional wired networks. It is necessary to take those unique features into account when proposing efficient management architectures for WSNs. This section discusses some design issues and requirements for proposing efficient management architecture in WSNs.

- **Energy efficiency:** One of the crucial design challenges in WSNs management is energy efficiency. As sensor nodes are operated on battery, keeping the nodes active all the time will limit the duration that battery last. Also, individual sensor nodes use a small battery as a power source and replacing or recharging of these batteries in remote locations is not practical. In some cases, solar cells can also be used as a source of energy but they provide limited power. Therefore, it is very important to tackle energy efficiently at all levels of sensor network management.

- **Robustness and fault tolerance:** WSNs are prone to network dynamics such as nodes dying, becoming disconnected, powering on or off, and new nodes joining the network. A management system must be resilient to network changes and reconfigure the network when needed.

- **Lightweight Operation:** Lightweight operation is the third important requirement to be considered by a management system. Sensor nodes in WSNs are generally operating with very tight resources. Traditional distributed approaches are normally heavyweight and therefore not feasible for WSNs. Thus, a system should be able to run on sensor nodes without consuming too much memory or energy or interfering with the operation of sensor nodes. Therefore, Lightweight operation prolongs network lifetime [Lee 2006a].

- **Scalability:** Generally WSNs are assumed to contain hundreds or thousands of sensor nodes. However the number of sensor nodes depends on the application and in some circumstances it might reach millions [Bulusu 2001]. Management architectures must support scalability of the network. Any increase in network nodes should not affect the overall performance of the network.

- **Minimal Data Storage:** A sensor node is equipped with only limited memory or storage space. Therefore, the data model used must be extensible and able to accommodate information required to perform management tasks, but also consider the memory constraints of sensor nodes.

- **Mobility:** Mobility is generally viewed as a major hurdle in the management of large scale wireless sensor networks. In fact, a hierarchical clustering and addressing scheme of the type used in the internet could be easily applied to a static sensor network (without mobile nodes) to manage routing. However, node movement imposes frequent hierarchical address changes, followed by update broadcast to the entire network. This is a very resource consuming proposition that can easily congest the entire network. Most of the network architectures assume that sensor nodes are stationary. However, the mobility of either base stations or sensor nodes is sometimes necessary in many applications [Ye 2002].

The protocol or algorithm for sensor network management should be robust to node mobility. Many applications of sensor networks or sensor network deployment have node mobility; some by design and some just by the nature of application (nodes may shift or move accidentally).

2.4.4 Existing WSN Management System Architectures

Sensor network management systems can be classified according to their network architecture into 3 categories: centralized, distributed, or hierarchical [Lee 2006a].

1) Centralized management system

In centralized management system, the base station acts as a central manager and controls the entire network. It collects information from all nodes and performs complex management tasks. The central manager has a global view of the network and with unlimited resources; it can provide accurate management decisions by reducing the processing burden on resource-constrained sensor nodes in the network. However, it incurs a high message overhead (energy and bandwidth) for data polling, and this limits scalability of the network. Nodes closer to the central manager will exhaust their energy much faster for forwarding messages to (or from) the others. Finally, if a network is partitioned then nodes that are unable to reach the central server are left without any management functionality. Some examples of centralized system includes: Sympathy [Ramanathan 2005a], BOSS [Song 2005] and SNMS [Tolle 2005].

2) Distributed management system

Distributed management system performs management tasks by employing multiple manager stations. Each manager is then responsible for a sub network and able to communicate directly with other managers to perform management task. Distributed management has lower communication costs than centralized management, and provided better reliability and energy efficiency [Lee 2006a]. Distributed system include: Node energy level management [Boulis 2003], App-Sleep [Ramanathan 2005b] and sensor management optimization [Perillo 2003].

A variation of distributed management is mobile agent based framework. Agents are used to distribute tasks in the network. Unlike centralized, these approaches reduce network bandwidth consumption by local processing and prevents network bottlenecks by reducing processing at the central station. A common example is MANNA [Ruiz 2003].

MANNA considers three management dimensions: function areas, management levels, and WSN functionalities (see figure 2.5). Similar to SNMP, MANNA consists of five traditional management functional areas, fault, configuration, performance, security, and accounting management. But configuration management in MANNA has more important role, where all other functions depend on it. The management levels in MANNA are similar to TOM: network element, network element management, network management, service management, and business management.

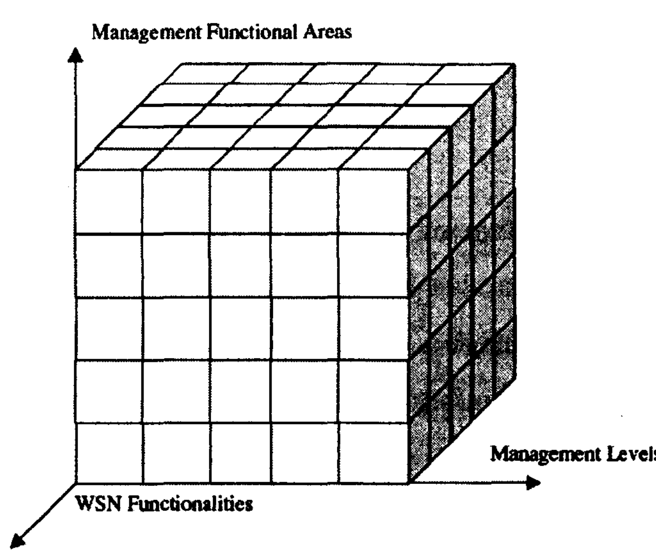

Figure 2.5: Management functions in MANNA [Sohraby 2007]

MANNA is an agent based management system, which creates a manager located externally to the wireless sensor network and has a global vision of the network and can perform complex operations that would not be possible inside the network. This approach is focused on event driven WSNs and is a policy-based management system. Management activities take place when sensor nodes are collecting and sending data. Every node will check its energy level and send a message to the manager/agent when there is a state change. The manager can then obtain the coverage map and energy level of all sensors based upon the collected information. This system provides two main management services: coverage area maintenances and failure detection services. The central manager uses the topology map and energy map to build a coverage area for sensing. However, this approach requires an external manager to perform the centralized diagnosis and the communication between nodes and the manager is too expensive for WSNs. Some other prominent examples of mobile agent based approaches are: Sectoral Sweeper [Erdogan 2003] and Mobile Agent-Based Power Management [Ying 2005].

There are some disadvantages of agent-based approaches. First, there is a need of special nodes to act as agents and perform management tasks. Secondly, the human managers need to place these agents 'intelligently' to cover all the nodes in the network. Third, the agent-based approaches introduce delay when retrieving nodes status as managers have to wait for the agent to visit the node in order to retrieve its status [Lee 2006a].

3) **Hierarchical management system**

A hybrid between centralized and distributed is hierarchical management system. It uses intermediate managers which do not communicate with each other directly. Each manager passes information from its management area to its upper higher management level, and also disseminates management functions received from the high-level manager. For examples, in TopDisc [Deb 2001] and STREAM [Ramanathan 2005b], common nodes coordinate to elect a manager among themselves to act as a distributed manager, and construct a hierarchical cluster based architecture.

A topology discovery algorithm, TopDisc [Deb 2001] makes use of a clustering mechanism to find the network topology. It creates clusters among the nodes and identifies clusterhead in order to report the network topology. Cluster heads report the topology information to the monitoring nodes or base station. In TopDisc, the clusters are created by finding the set coverage with greedy approximation algorithm. The algorithm begins by the monitoring node broadcasting a topology request message. The request message is propagated throughout the WSN. TopDisc consists of two different node coloring approaches. The first coloring scheme uses a three coloring approach. TopDisc is scalable as it uses only local information. However, it does not guarantee a certain distance between CHs i.e. some CHs get too close to each other and do not cover an optimal number of non CHs nodes.

Mobile agent-based policy management [Ying 2005] scheme is a good example of distributed hierarchical systems. It aims at providing effective management scheme in the form of pre-defined management policies. These management policies and rules are enforced by mobile agents, in order to keep the wireless sensor networks running in a normal, stable and reliable way with high efficiency. Each rule consists of conditions and management operations to be executed when the conditions are satisfied. Figure 2.6 shows the hierarchical architecture of this management system.

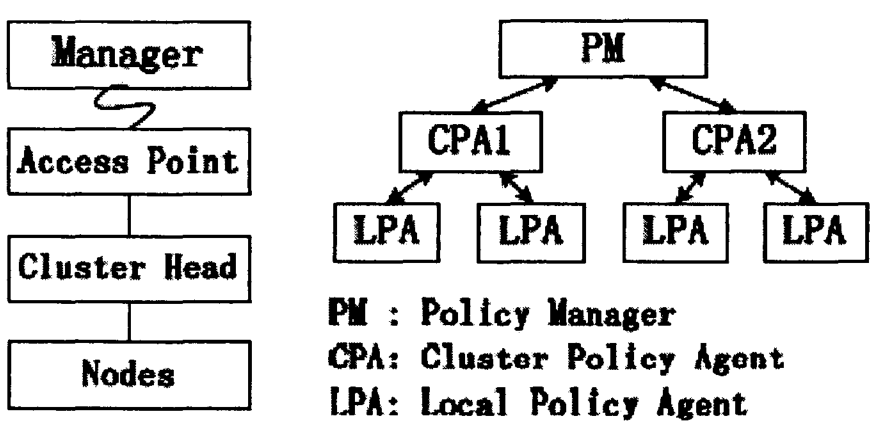

Figure 2.6: Hierarchical architecture of policy management [Ying 2005]

As discussed in [Lee 2006a], the system consists of 3 levels: Policy Manager (PM) at the highest level, Cluster Policy Agent (CPA), and Local Policy Agent (LPA). The PM manages multiple CPAs and adaptively reconfigures the network (locally or globally) when network conditions change. Multiple LPAs are managed by a CPA. An LPA manages a sensor node and also enforce local policies by analyzing network dynamics, performing configuration, monitoring, filtering and reporting. Policies propagated from PM to CPAs, CPAs to LPAs, or from CPAs to LPAs. The advantage of this system is that policies agents organize in hierarchy can be used to perform network management function either locally or globally. New management functions can be injected into the system by the end user. However, large numbers of messages are exchanged to form the management hierarchy.

To summarize, in this section we presented an overview analysis of existing management schemes, so as to find out and summarize their advantages and disadvantages. As discussed earlier existing management solutions for WSN can be categories into centralized, distributed or hierarchical. Centralized management schemes incurs high message overhead in terms of bandwidth and energy. Also, they are not scalable with the growth of the network. Distributed or hierarchical management solutions though more efficient for WSNs, but consume much energy to form the management hierarchy.

2.5 Summary

In this chapter we discussed the ongoing research efforts and projects in the area of wireless sensor networks self-organization and self-management. We also presented a survey on current clustering schemes for the self-organization of wireless sensor networks. We first discussed some of the key issues that differentiate wireless sensor network management from that for traditional networks. We then presented a brief overview of WSN management, management challenges for WSNs and discussed WSNs management architectures.

Most existing management solutions fall short of matching the characteristics of wireless sensor networks and cannot effectively support their applications. We believe that a hierarchical cluster-based framework is a technique that helps to design energy-efficient and scalable management systems for WSNs. It conserves node energy by aggregating redundant sensor data in the network. Clustering involves arranging the nodes into groups, and identifying a leader node for each clustering. This design eliminates redundancy of transmitted data and conserves node energy by reducing the number of communication messages. However, existing clustering schemes are not efficient and consume much energy for cluster formation and reconfiguration. So we believe that a systematic approach for clustering would be a promising solution for self-organizing and self-managing WSNs.

3. A Hierarchical Cellular Self-organizing Scheme for Wireless Sensor Networks

In this chapter we present a new cellular self-organizing hierarchical architecture [Asim 2008a] for wireless sensor networks wherein sensor nodes are arranged into a multilayer architecture. We propose a novel n-tier hierarchical framework for wireless sensor networks. However, the number of hierarchical levels is based on application type and number of nodes. Existing hierarchical scheme for WSNs are based on fixed parameters and therefore can be used for specific applications. However, our generic cellular hierarchical framework allows us to define a number of parameters i.e. number of hierarchical levels, cluster size. These parameters can be defined based on the application requirements. For example, by defining the correct cell size for a particular application helps in the optimal distribution of managing sensor nodes across the network and provides the maximum coverage of sensor nodes. Our cellular architecture addresses various limitations of existing clustering schemes that we discussed in chapter 2. It extends the network life by efficiently utilizing nodes energy and supports the scalability of the system in densely deployed sensor networks. Our aim is to use the cellular architecture for managing WSNs.

3.1 Background

One of the crucial design challenges in wireless sensor networks is scalability. A sensor network may consist of hundreds, thousands, or even millions of inexpensive wireless sensor nodes that may be placed either regularly or irregularly. Designing and operating such a large size network would require scalable architecture and management strategies. Moreover the number of WSN applications is increasing due to their unique characteristics. Future applications will be highly dense, e.g. whole countries and cities will be monitored for various purposes using WSNs. Therefore scalability is a core issue and this can affect the performance of any proposed protocols especially when we also take into account the resource limitations of WSNs.

As described in chapter 2, an efficient way to tackle scalability is by dividing the network into small groups called clusters. This involves organizing nodes into groups or clusters and identifying a leader, or clusterhead, for each cluster. Clustering can distribute the management strategies across the network to further enhance the network operation and prolong the battery life of the individual sensors and the network life time. Furthermore, a clusterhead can aggregate the collected data to decrease the number of relayed packets.

A wireless sensor network may consist of large number of nodes. Proficient organization of these sensor nodes has crucial effect on bandwidth resources, traffic load, dynamic topology, etc. of a wireless sensor network. Hierarchical or tree based clustering is an energy efficient way to administer these sensor nodes. The organization of these sensor nodes could be in a single hierarchy with few hundred nodes or multi-level hierarchies with thousands of nodes organized in several levels [Abbasi 2006]. We analyzed a number of hierarchical clustering schemes in chapter 2, and highlighted their advantages and disadvantages. Existing hierarchical clustering schemes for WSNs offer promising improvements over conventional clustering; however there is still much work to be done. Most existing clustering schemes consume too much energy in group formation and re-formation. Normally, cluster formation consists of two phases: clusterhead election and assignment of nodes to clusterheads. Many attempts have been made to minimize the energy associated in clusterhead selection process [Abbasi 2007], and for achieving a desirable distribution of clusterheads [Younis 2004]. However, none of them offers optimal clustering in terms of energy efficiency to reduce the overhead associated not only with clusterhead selection process, but also with nodes association to their respective clusterheads.

As discussed in chapter 2, some cluster based networks are heterogeneous and consist of different type of nodes i.e. load balancing clustering scheme [Gupta

2003b], where some nodes are less energy constrained than others. In such type of networks the less energy constrained nodes are chosen as leader nodes or clusterheads. The problem arises when the network is deployed randomly and most of the clusterheads are deployed in a particular area of the network. This results in an uneven distribution of managing nodes. Therefore, heterogeneous cluster based sensor networks require a careful management of the clusters in order to avoid the problems resulting from unbalanced clusterhead distribution.

Most existing works on clustering in wireless sensor networks treat a network as geography – unaware graph. However, geographic-unaware clustering can cause a number of problems such as: the communication links between a cluster leader and its cluster members are long, the geographic overlap between neighboring clusters is large, and routing traffic load is unbalanced across different clusters. Consequently, this reduces the overall life time of the network as well as the communication quality and efficiency of the network. Many multi-hop wireless sensor network applications i.e. temperature sensing and environmental monitoring, are inherently geographic aware. Thus, reflecting geography in the underlying network structure optimizes system performance. Therefore, in order to improve efficiency, scalability, save energy and improve communication quality, geographic aware radius of cluster should be taken into account in clustering algorithms [Zhang 2003].

To summarize, prolonging network lifetime, scalability, load balancing and incorporating network dynamics are important requirements for many ad–hoc sensor networks [Younis 2003]. Many solutions used hierarchical (tiered) clustering architectures to address these requirements [Banerjee 2001, Chen 2007a, Heinzelman 2000]. Hierarchical architectures differ in terms of cluster formation, clusterhead selection process, number of hierarchies, and type of nodes etc. There are various limitations offered by existing hierarchical architectures i.e. energy consumed in cluster formation; optimal distribution of clusterheads in the network; managing sensor nodes at different levels of the hierarchy; energy consumed to form a management hierarchy and data aggregation. Normally the clusterhead is responsible for aggregating the collected data and then forwarding it to the base station. However, this operation can be too energy intensive.

We address these challenges by proposing a distributed cellular architecture that partitions the whole network into a virtual grid of cells. Our vision is that static clustering scheme based upon location is a flexible solution to accommodate different types of applications. The proposed algorithm is then extended to a hierarchical architecture with nodes organized into different layers.

3.2 Hierarchical Clustering Design Objectives

By analyzing existing schemes, we found that there are very important issues related to clustering in WSNs that need to be addressed. We therefore outlined the following design objectives for proposing a new clustering scheme for WSNs.

- A clustering scheme should ensure that energy dissipation across the network is balanced and that clusterheads are optimally distributed across the network.
- A clustering scheme should not consume much energy in cluster formation/reformation.
- For traffic optimization and energy efficiency, a clustering scheme should adapt a layer based data aggregation process to reduce the clusterhead overload.
- A clustering algorithm should achieve a balance distribution of nodes among clusters.
- A clustering scheme should cover all the deployed sensor nodes and no nodes should be left uncovered after clustering.

3.3 Energy Model & Assumptions

The sensors are assumed to be capable of reporting their remaining energy and operating in an active mode or a low power standby mode. It is also assumed that sensors can act as a relay to forward data from another sensor. In our proposed architecture, network management and organization are energy aware and rely on the knowledge of energy reserve at each sensor node. In our architecture we used the radio model proposed in [Younis 2006] which is one of the most used models in the wireless sensor networks.

The key energy parameters for communication in this model are the energy/bit consumed by the transmitter electronics ($\alpha 11$), energy dissipated in the transmit op-amp ($\alpha 2$), and energy/bit consumed by the receiver electronics ($\alpha 12$). Assuming a $1/dn$ path loss, the energy consumed is:

$$Etx = (\alpha 11 + \alpha 2\ dn) * r \qquad and \qquad Erx = \alpha 12 * r$$

where Etx is the energy to send r bits and Erx is the energy consumed to receive r bits. Table 3.1 summarizes the meaning of each term and its typical value.

Term	Description
$\alpha 11$, $\alpha 12$	Energy dissipated in transmitter and receiver electronics per bit (Taken to be 50 nJ/bit).
$\alpha 2$	Energy dissipated in transmitter amplifier (Taken = 10 pJ/bit/m2.
r	Number of bits in the message.
d	Distance that the message traverses.

Table 3.1: Communication energy model parameters

Communication in each cell is one-hop transmission to cell manager. Also, one-hop communication between Cell-heads and Cell-head to Group-head is unrealistic because of physical constraint i.e. geographic location. Thus, communication between Cell-head to Cell-head and Cell-head to Group-head involves multi-hop transmission with minimum transmission range for connectivity.

In this work, the network model is based on the following assumptions:

- The sensor nodes are assumed to be homogenous i.e. they possess the same processing power and initial energy. We consider that all the nodes in the network are equal in resources and no node should be more resourceful than any other node.
- Two nodes can communicate with each other directly if they are within the transmission range.
- Sensor nodes are assumed to know their locations or relative position through location techniques such as the recursive position estimation [Albowitz 2001] or virtual co-ordinate system [Gautam 2009].
- All sensors transmit at the same power level and hence have the same radio range.
- The sensor is assumed to be capable of reporting its remaining energy and operating in an active or a low-power stand-by mode.

3.4 The Proposed Hierarchical Cellular Architecture

In this section, we describe our proposed cellular-based hierarchical architecture to meet the unique requirements of wireless sensor networks. Let us consider a continuous distribution of sensor node on a 2D plan and divide sensor nodes into virtual cells of equal radius with minimum overlap between neighboring cells to achieve a cellular structure as shown in Figure 3.1. More specifically, sensor network nodes configure themselves into a virtual grid structure, in which the network nodes are partitioned into several cells each with a radius that is tightly bounded with respect to a given value radius R and zero overlap between neighbouring cells. A cell can be considered as a special kind of clustering. However it is more systematic and scalable.

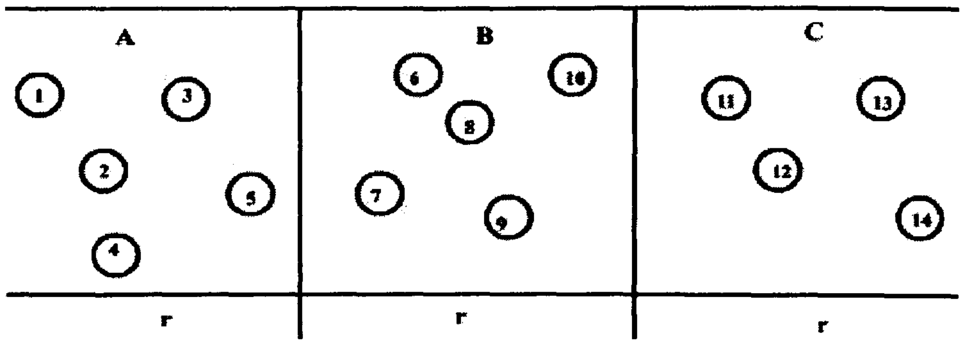

Figure 3.1: Division of the network into a virtual grid

The aim of using a cellular architecture is to associate every node with one cell. Guaranteeing that each node belongs to one and only one cell helps reducing network energy consumption and supports scalability. Data from neighbouring sensor nodes are often correlated in wireless sensor networks but the end user needs only a high-level aggregation of the data that describes the events occurring in the environment. Because the data correlation is strongest between sensor nodes close to each other, we chose to use a cellular infrastructure as the basis for our cellular architecture. This allows sensor nodes to aggregate similar packets locally and reduces the number of transmission.

A grid-based architecture is feasible in a network in which nodes are relatively regularly deployed. One node in each cell is distinguished as the Cell-head, to represent this cell in the network. All Cell-heads in the network form an upper level grid and the remaining nodes form a lower level grid. A set of virtual cells are aggregated to form a large virtual group, which might consist of nodes from hundreds to thousands in number. A Group-head is appointed for each group, and is responsible for managing and organizing sensor nodes in its group. Following the same process, Group-heads from different groups form another virtual grid structure towards the base station. We propose an n-tier hierarchical clustering for wireless sensor networks. However, the number of hierarchical levels is based on application type and number of nodes. Figure 3.2 depicts the overall cellular hierarchical architecture.

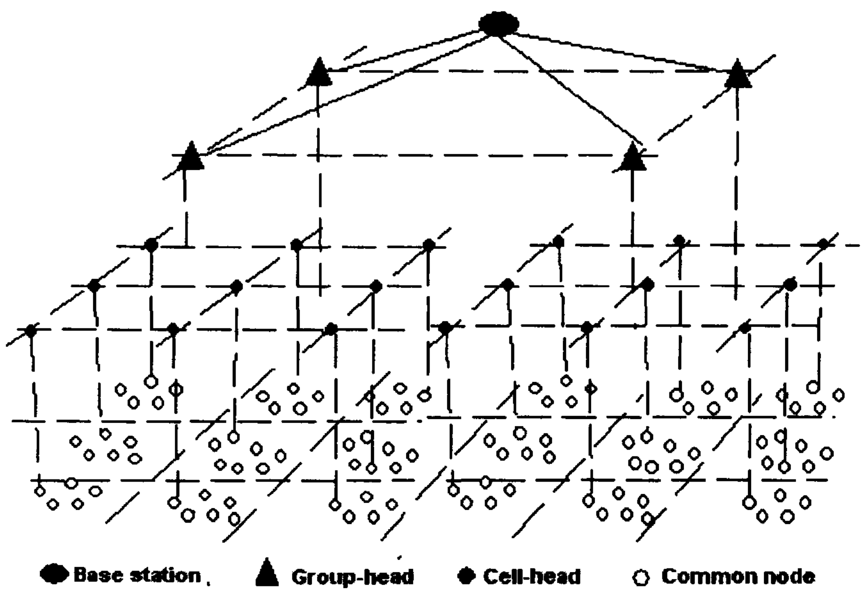

Figure 3.2: Cellular based hierarchical Architecture

3.4.1 Formation of the Cellular Architecture

The formation of cellular architecture is divided into two phases: cell formation phase and cell-head and group-head selection phase.

1) Cell formation phase

For simplicity, we assume that the sensors have no movement during cell formation phase. The steps for establishing and deployment of cellular grid are as follows:

We assumed that each sensor node in the network knows its location or relative position through location techniques or using virtual co-ordinate systems. The geographic area of the wireless sensor network is partitioned into two dimensional virtual grids and each cell has its unique co-ordinate identifier (x, y). As shown in Figure 3.3, each sensor node can calculate in which cell it currently dwells based on its location information and using the following equation.

$$Cell\_idx = (x - X\min)/d \quad and \quad Cell\_idy = (y - Y\min)/d$$

Where *Cell_id* is the co-ordinate identifier (x, y), *Xmin* and *Ymin* are the x and y co-ordinate of the node with minimum co-ordinates in the network. x and y are the co-ordinates of current node.

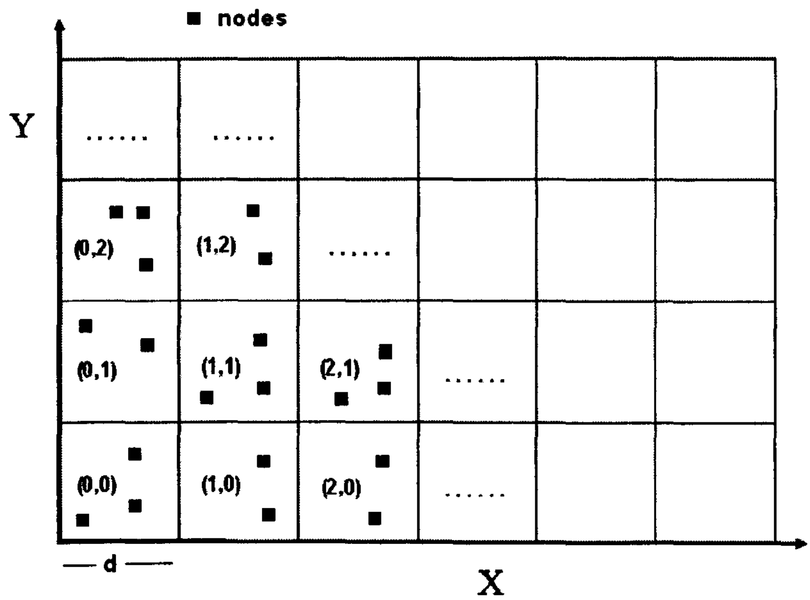

Figure 3.3: The deployment of virtual grid

The size of the cell (i.e., *d*) is made available to the sensors during network initialization. The cell size *d* is based on application type and can vary. In order to have one-hop communication between cell members and the Cell-head, we are assuming radio range r equal to $d\sqrt{2}$. The nodes appeared on the border will always select the cell with fewer nodes.

The next stage is a discovery phase, where each node discovers its set of cell members. Every node n picks a small radius ($d\sqrt{2}$) and broadcast a *hello message* around a radius r. Every node within the radius r reply back with a *hello message*. A *hello message* consists of the node ID, location of the node and a cell id. A hello message will be dropped, if heard by a node belonging to a different cell.

***2)* Cell-head and group-head selection phase**

As discussed above, sensor nodes exchange hello messages to discover their cell members. If a node *i* hears from a node *j*, the node *i* first checks if node *j* belongs to the same cell. It then compares its co-ordinates with node *j* co-ordinates and stores its id as a Cell-head id, if node *j* co-ordinates are higher than itself. Once the nodes are organized into a cellular architecture, a node declares itself as a Cell-head if it has the highest co-ordinates of all its cell members. Figure 3.4 describes the flow chart of Cell-head selection algorithm.

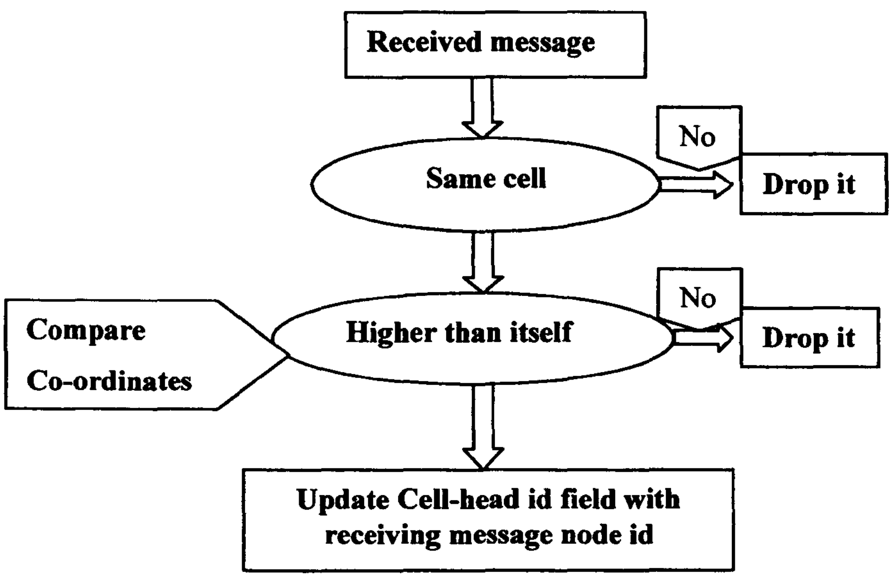

Figure 3.4: Cell-head selection algorithm

This criterion for the selection of cell-head is for initial deployment only. After that, selection of Cell-head is based on available residual energy. There is one –hop communication between Cell-head and its member but in some scenarios multi-hop communication is required i.e. communication between cell-head and group-head. The Cell-head role itself does not change but Cell-head does change inside the cell. The Cell-head selection algorithm is performed in all network cells.

Initially, all the nodes are assigned the same rank. After going through various transmissions, the node energy decreases. If the node energy is greater than or equal to 50% of the battery life, it is ranked as high and becomes a promising candidate for the Cell-head role. If the node energy becomes less than or equal to 20% of battery life, it is ranked as low node and becomes liable to put to sleep. The nodes having battery between 20% and 50% or higher are suitable candidates for routing and sensing.

A cellular based architecture helps in identifying redundant nodes across the network. The Cell-head is responsible for collecting information of its cell members, and determining the existence of redundant sensors based on their location. . For redundant sensors located on the boundary of the cells, Cell-heads coordinate to make decisions. The Cell-head can also monitor its cell members and initiate a relocation process in case of new event or sensor failure. Redundant nodes may send to a low computational mode to conserve energy. The cell size can be other criteria to identify redundant nodes i.e. restricting the cell to have a total number of S nodes. *S is* a user-defined parameter, which can be adjusted to meet the required Cell-head density. If a cell size is above the threshold value *S,* then some nodes can be sent to sleep mode to adjust the cell size. The low energy nodes are replaced by awaking other sleeping redundant nodes in their respective cells or moving mobile sensor nodes to that area. This helps to achieve a gradual reduction in the overall network energy.

A set of virtual cells are aggregated to form a large virtual group, which might consist of hundreds of sensor nodes. A Group-head is appointed for each group, and is responsible for managing and organizing sensor nodes in its group. Cell-heads of a particular group coordinate and exchange Group-head selection messages. A Cell-head with the highest residual energy is selected as a Group-head for that group. Figure 3.5 demonstrates the flow chart of Group-head selection algorithm.

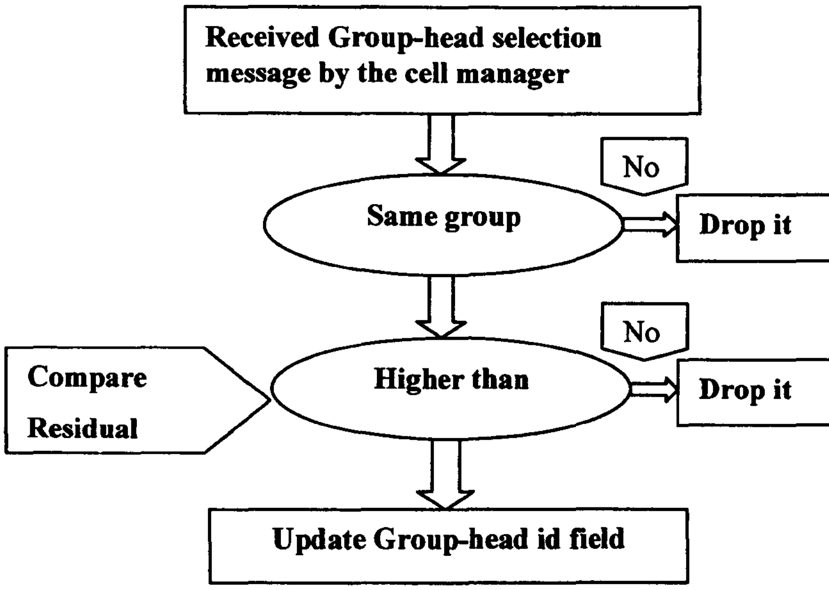

Figure 3.5: Group-head selection algorithm

Following the same process, Group-heads from different groups form another virtual grid structure towards the base station in the management hierarchy. In order to cover all sensor nodes across the network, our proposed cellular architecture optimally distributes cell and group-heads across the network. This approach maximizes the network life time by providing the maximum coverage of the sensor nodes. Our cell and group formation algorithm consume less energy as they are based upon the actual or virtual coordinates of the node.

3.4.2 Performance Optimization

Our proposed algorithm is based on message filtering to lessen the redundant message exchange during Cell-head and Group-head selection phase. To demonstrate the idea, we consider the following message format.

Addr	GROUP	MS_TYPE	....	DATA	....

Figure 3.6: Message format

We apply the MS_TYPE field in the data packet as in Figure 3.6, which distinguish messages from different sensor nodes in the management hierarchy. It contains message types i.e. Cell-head selection message, Group-head selection message. The 'GROUP' field, containing the value of group id, and is applied to distinguish and avoid receiving messages from other groups. The 'DATA' field is actually the structured message packet (i.e. both cell-head selection message and group-head message) and contain fields as shown in Table 3.2.

Group_id	The group id
Cell_id	The cell manager id
Timestamp	The message sending out time
Curr_energy	The current node battery enery
Src	Source address
Des	Destination address
Hop_cn	Record the communication hop

Table 3.2: Message attributes

We apply three stages of message filtering to lessen the redundant broadcast in the network for energy conservation i.e. the message type, and timestamp.

The message type stage first adopts the 'GROUP' field to quickly determine whether the received message belongs to the same group of current node. If not, the message will be dropped to avoid unnecessary message re-broadcasting. It then checks the 'MS_TYPE', distinguishing data packets from cell-head selection and group-head selection messages. After retrieving the value of 'cell_id', the node decides whether it belongs to the event cells (e.g. destination cell) to process the message. Otherwise, it will re-broadcast the message. A sensor node might receive multiple copies of the same message forwarded by different intermediate nodes. To avoid redundant rebroadcast, we apply the value of 'timestamp' field in the second stage to determine whether the receiving message has been handled previously. If the receiving message is a new one, it will be processed and forwarded to the neighbouring nodes. On the contrary, that message will be dropped to lessen the network traffic and conserve the node energy.

3.4.3 Cell Size and Level of Hierarchies

In order to identify redundant sensor nodes and to minimize the overhead of intra-cell communication in a sensor network, the appropriate design of cell size is crucial. Cell size is affected by factors such as the transmission range of the transmitter, or the transmission power and the sensing range of the sensor nodes. Varying the cell size in the network affects the lifetime of the network. Below we will discuss the impact of smaller and larger cell sizes on the network.

Smaller cell size results in the following:

- Number of hierarchical levels increase
- Data delay packet tends to increases because smaller cell size means larger average hops from sensor nodes to the base station
- Cell-head overhead decreases because smaller cell size means smaller number of sensor nodes
- A very small cell size will also lead to wasted resources, as the transmit power and the receiver sensitivity allow a minimum distance between the nodes to be covered

Larger cell size results in the following:

- Number of hierarchical levels decrease
- Energy consumption increases because larger cell size means cell-head consume more power for communication
- Data packet delay decrease as average hops from sensor nodes to the base station decreases
- Overhead on cell-head increases as larger cell means greater number of sensor nodes
- If the cell size is too large, it will lead to early partitioning of the network as some sensor nodes may not be in communication range of each other.

In our proposed cellular architecture, cell size is a user-defined parameter, which can be adjusted to meet the required Cell-head density. Also, to keep the hierarchical structure efficient, load for each clusterhead should be equivalent. Thus, the cluster size is a key parameter to achieve balanced load among clusters.

Cell-head density will be defined according to application requirements. Appropriate cell-head density plays an important role in maximizing the performance of the network. However, for most sensor networks application, it is important to support fast delivery of important and urgent data. For example, consider a sensor network deployed to sense the temperature in a forest. An abnormally high temperature in a particular location may be an indication of a fire. As a result, such messages have to be transferred to the base station as fast as possible, not being delayed or lost. Also, maximizing cell-head density may put extra burden on cell manager for certain operations i.e. data aggregation. Therefore, it is extremely important for the performance of sensor networks to carefully define cell size and cell-head density.

One important challenge for WSNs is aggregation of redundant data. Similar packets from multiple nodes can be aggregated so that number of transmissions would be reduced. This technique is used to achieve energy efficiency and traffic optimization. However, this operation can be too energy intensive. We therefore, used a layered data aggregation process. By using this organization, data from leaf node towards the clusterhead is aggregated at each layer of the hierarchy and thus avoid the clusterhead overload and offer more energy saving. Most network architectures assume that sensor nodes are stationary but sometimes it is deemed necessary to support the mobility of the nodes. In cellular architecture, since the network is partitioned into logically separate cells, it can easily keep track of mobile nodes i.e. node joining/leaving a cluster, topology of the cluster and node capabilities.

3.5 Analytical Evaluation

In this section our proposed architecture is evaluated analytically and compared to existing clustering solutions. We define the following criteria: load balancing, energy consumed for clustering and re-clustering, coverage and data aggregation.

- **Load balancing:** Load balancing is an important issue in WSNs where CHs are picked from available sensors. In such case, even distribution of managing sensor nodes becomes crucial for extending the network life time since it prevents the exhaustion of the energy of CHs at high rate and prematurely making them dysfunctional.

  In heterogeneous sensor networks (i.e. load balancing clustering scheme [Gupta 2003b]), some highly energy constrained sensor nodes are used for load balancing. As discussed earlier these type of networks results in an uneven distribution of CH nodes during random deployment. Sensor nodes close to the CH die quickly while creating holes in the network and decrease network connectivity. Therefore, heterogeneous cluster based sensor networks require a careful management of the clusters in order to avoid the problems resulting from unbalanced clusterhead distribution. To utilize the nodes to their maximum lifetime, our cellular based clustering employs the use of load balancing. Our approach does not rely on specific nodes with extra resources but assign tasks due to their optimal capabilities. Nodes are ranked according to their available energy. Therefore, the selection of a Cell-head is based on the available energy. The basic idea of this design is to encourage nodes to be more self-organized and extend the network life time for as long as possible.

- **Energy consumed in cluster formation:** As discussed earlier, cluster formation normally consists of two phases: clusterhead election and assignment of nodes to clusterheads. Many attempts have been made to minimize the energy associated with cluster formation process. However, none of them offers optimal selection of CH in terms of energy efficiency, but also with node association to their respective CHs. Some schemes are based on periodic broadcasting of messages to all the active sensor nodes of the network to create clusters i.e. autonomic self-organizing algorithm [Chen 2007a]. This is not an energy efficient approach as too many cluster formation messages are flooded across the network. Our approach avoids flooding of initial discovery messages and cells are formed in a localized distributed fashion. It further reduces message exchange redundancy by employing message filtering. The clusterhead selecting algorithm in Leach [Heinzelman 2000] is not energy efficient, because it does not take the residual energy of the nodes into account. CHs in Leach [Heinzelman 2000] are selected randomly and may result in some part of the network being uncovered. Managing nodes are selected based on available residual energy and in on-demand fashion in our proposed cellular scheme.

- **Coverage:** The proposed scheme guarantees that each node will belong to only one cluster as the choice is based upon its co-ordinates. This helps in covering all sensor nodes across the network and maximizing the network life time by providing the maximum coverage of the sensor nodes. In the hierarchical clustering algorithm [Banerjee 2001], authors consider only logical radius of the cluster instead of geographical radius, which can reduce wireless transmission efficiency because of large geographical overlaps between clusters. Leach [Heinzelman 2000] offers no guarantee about the placement or number of clusterhead nodes. Both [Gupta 2003b] and autonomic algorithm [Chen 2007a] has lower cover loss ratio but do not guarantee the coverage of every node in the network.

- **Data Aggregation:** Aggregation of redundant data in sensor networks helps to achieve energy efficiency and traffic optimization. Normally the clusterhead is responsible for aggregating the collected data and then forward it to the base station. However, this operation can be too energy intensive i.e. Leach [Heinzelman 2000], when it sends data to the base station, it is in the form of one-hop routing. In Leach, CH can transmit data directly to the sink node. However, CH nodes can be at a large distance from the sink node and the nearest CH nodes may overload. Managing nodes in our hierarchical cellular scheme can perform data aggregation at different levels of the hierarchy, and thus avoids the clusterhead overload and offer more energy saving.

3.6 Experimental Evaluation of the Cellular Architecture

In this section we evaluate the performance of our proposed algorithm. We used GTSNETS [Riley 2003] as simulator platform. Georgia Tech Sensor Network Simulator (GTSNETS) is a simulation tool that enables the development and evaluation of algorithms for large-scale WSNs. The design of GTSNETS matches closely with the design of actual network protocol stacks and other network elements. Further, GTSNETS was designed from the beginning to run a distributed environment, leading to better scalability. It also supports the simulation of network control systems having sensing, control and actuation capabilities which have been lacking in other sensor network simulators. GTSNetS is a fully-featured sensor network simulation tool. It provides each sensor node a simulated battery in order to measure the energy consumption. Moreover, GTSNETS is distributed under the GNU General Public License and is freely available [Ould-Ahmed-Val 2005].

We used the same radio model as discussed in section 3.3. The available energy per sensor nodes is assumed to be 2J (2000 mJ) in the initial time. The energy dissipation parameter *Eelec* is assumed to be 50 *n*J/bit, and the amplifier energy is 10 pJ/bit/m^2. The experiment assumed that channel allowed collision and that packets could be dropped in the medium. Sensors are given IDs in random fashion. All nodes are considered equal and no preference is given to any sensor.

Parameters	Value
Number of nodes	40 to 500
Node initial energy	2J (2000 mJ)
Energy dissipation	50nJ/bit
Amplifier energy	10 $pJ/bit/m^2$
Transmission range	50 to 80 m
MAC protocol	IEEE 802.11
Nodes deployment	Random

Table 3.3: Parameters for simulation

Experiments were performed to elucidate the characteristics of the proposed mechanism using the following performance metrics.

- **Distribution of sensor nodes:** Experiments were performed to measure the balanced distribution of sensor nodes among cells
- **Energy consumption for cell formation:** Experiments were performed to measure the average energy consumption in cell formation
- **Average cover loss ratio:** Experiments were performed to measure the total number of nodes not covered after clustering

1) Distribution of sensor nodes

In order to measure the balanced distribution of sensor nodes among cells, 500 nodes were randomly distributed in a 400x400 square meter area. In WSNs, uneven cluster size results in unbalanced data traffic load among clusters. Clusters that have more members than others suffer from congestion and data loss which negatively affect the accuracy of the collected data. In addition, the clusterheads of such clusters exhaust their energy earlier than others, thereby reducing the network lifetime. Figure (3.7) depicts the number of nodes per cell in our scheme.

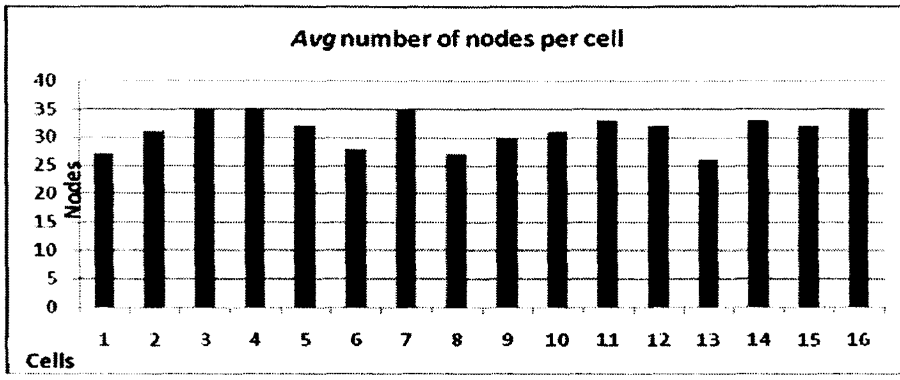

Figure 3.7: Avg number of nodes per cell

It can be observed from figure (3.7) that our proposed scheme achieves a balanced distribution of sensor nodes among cells, thereby increasing the network life time and balancing cell-heads.

2) Energy consumption for cell formation

To evaluate the performance of our proposed algorithm, we compare our scheme with autonomic self-organizing architecture [Chen 2007a]. Number of sensor is varied from 40 to 120, which are randomly deployed over 150 X 150 square meter area. Autonomic architecture is based on a 3 levels hierarchical architecture, where low level nodes are managed by high level nodes, forming a hierarchical management system. Our proposed architecture is also a hierarchical architecture, where lower level nodes are managed by the higher level nodes. This supports our efforts to compare our proposed mechanism with the autonomic architecture.

Figure (3.8) and figure (3.9), depicts the energy drain during cell formation (cluster formation). It can be observed from the graphs that the energy drain in our algorithm is lesser than the other one. Autonomic algorithm addresses the cluster formation from high level nodes (headers) through contests with low level nodes using minimum hop count as a primary metrics. Managing nodes broadcast a 'cover request' (CREQ) periodically. The CREQ messages are delivered to all of the active sensor nodes of the network. The lower level nodes select the cluster header using the minimum_hop_count method. The nodes then forward the CREQ to cover nearby nodes in its radio range, once they have accepted a header. This is not an energy efficient approach as too many CREQ messages are flooded across the network. Also, high level nodes send CREQ messages periodically to cope with self-organization of the network.

Our proposed architecture is an energy efficient approach towards cluster formation as proven through graphs. During discovery phase, nodes with in a cell exchange hello messages. A hello message consists of the node ID, location of the node and its cell id. A hello message will be dropped, if heard by a node belonging to a different cell. Each node well known its cell members, node with highest co-ordinates becomes the Cell-head. Our approach avoids flooding of initial discovery messages and cells are formed in distributed fashion.

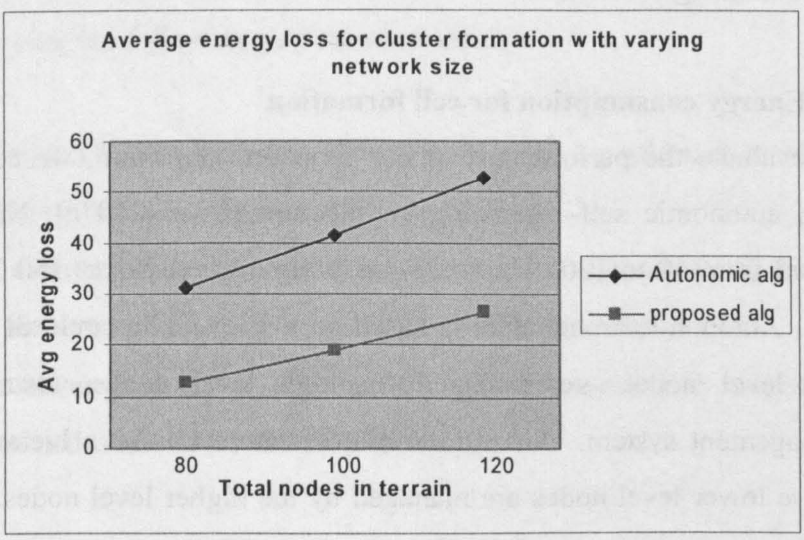

Figure 3.8: Average energy loss for cluster formation with varying network nodes

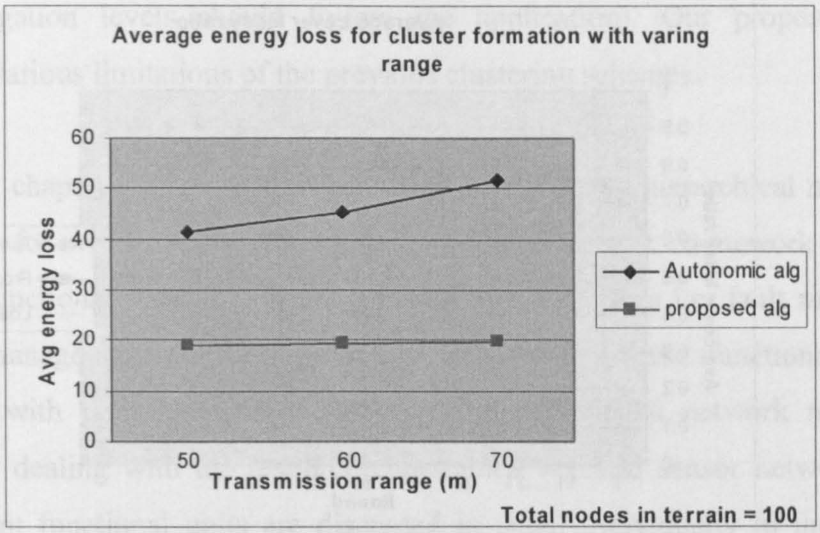

Figure 3.9: Average energy loss for cluster formation with varying radio range

3) Average cover loss ratio

We compared our work to autonomic algorithm to measure the total number of nodes not covered after clustering. Autonomic algorithm performed experiments to measure the total number of nodes covered after clustering. 100 - 300 nodes were deployed randomly in an environment with an area in the range of 100 x100 to 300 x 300 square units. They proved through simulation that their algorithm has lower cover losses than load-balanced algorithm. The load-balanced algorithm adopted only one hop to cover its member. Cover loss occurred in the load-balanced algorithm when the member nodes are out of radio range. Also, autonomic algorithm employed multi-hop to cover its member nodes but it treats the network as geographically unaware. This can result in a problem that long distance sensor nodes may not receive CREQ request messages and stay uncovered. However, as depicted in figure (3.10), our proposed algorithm offer 100% cover ratio as it's based on geographical boundary and can cover all the deployed nodes.

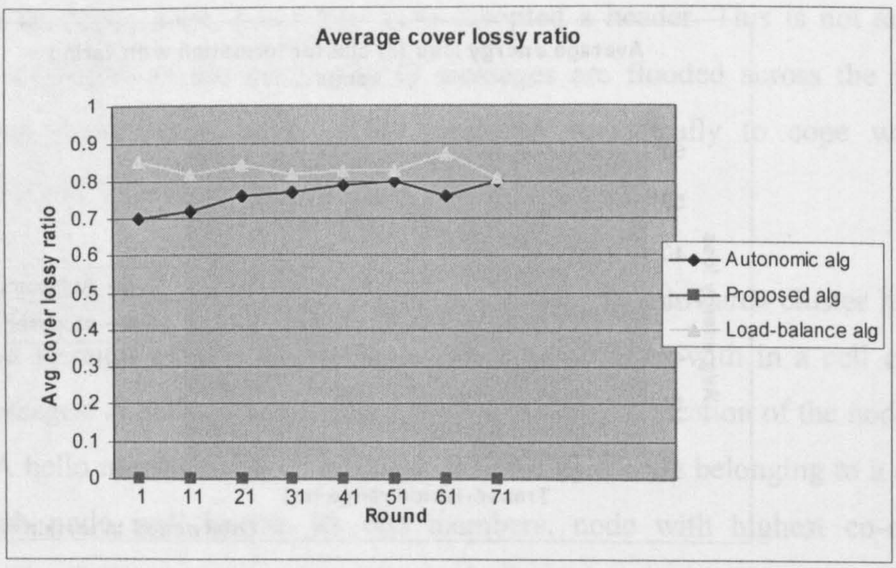

Figure 3.10: Average cover loss ratio

3.7 Discussion

Designing clustering algorithms for WSN needs to consider many issues and challenges that have never been addressed in traditional networks. The most power-consuming activity of a sensor node is typically radio communication; this applies to transmission and reception, and also to listening for data. Hence, radio communication must be kept to an absolute minimum. This means that the amount of network traffic should be minimized. In order to reduce the amount of traffic in a network, we can build clusters of sensor nodes [Hansen 2006]. Many clustering algorithms in various contexts have been proposed for wireless sensor networks. However, existing clustering schemes have high cost for clustering and re-clustering. Because of the highly dynamic nature of wireless sensor networks, forming and reforming of clusters are needed frequently and thus consume much energy. We therefore developed a static hierarchical clustering scheme that is based upon location rather than any specific set of nodes. This scheme is flexible enough to accommodate the different application of WSNs. Our generic cellular hierarchical framework allows us to define a number of parameters i.e. number of hierarchical levels, cluster size. These parameters can be defined based on application requirements. (e.g. data aggregation is dependent on application, so the cluster size and aggregation levels should follow the application). Our proposed scheme addresses various limitations of the previous clustering schemes.

In the next chapter we map the cellular architecture into a hierarchical management framework for wireless sensor networks. The management framework consists of different functional units for different management services i.e. fault management, mobility management and configuration management. These functional units are integrated with each other to provide an energy efficient network management system for dealing with the resource constrained wireless sensor networks. These management functional units are discussed in detail individually in the following chapters. In chapter 5, we discuss and evaluate the configuration management unit. Chapter 6 presents the fault management unit. In chapter 7, we discuss the mobility management.

3.8 Summary

In this chapter, we proposed a distributed hierarchical cellular architecture for WNSs. The aim was to achieve scalability, save energy and efficiently distribute management tasks across the network. Our proposed scheme optimally distributes managing nodes to achieve load balancing. Redundant data can be aggregated at different layers to achieve energy efficiency and traffic optimization. Redundant sensor nodes can be sent to sleep mode to save energy. The results obtained from the simulation have shown that our clustering architecture is more energy efficient and achieves better energy consumption distribution.

4. A Management Framework for Wireless Sensor Networks

A sensor network is a distributed system, where a set of N sensor nodes operate collaboratively towards a common goal in a resource and time constrained environment. The maintenance and control of such systems is vital to ensure efficient use of resources for appropriate information gathering and processing; since most of these networks must operate in an unsupervised environment. Despite the existence of management frameworks for well-resourced wired networks, there is no reason to believe a priori that such frameworks will apply to the resource-constrained environments of typical environmental sensor networks.

In this chapter we map the cellular architecture (discussed in chapter 3) into management architecture and propose a novel framework to support the network management system design for resource constraints wireless sensor networks. This self-managing framework can be use as a generic management solution to support different wireless sensor network applications. It provides different functional units for different management services i.e. fault management, mobility management, and configuration management. Current management schemes for WSNs consume too much energy in exchange of management messages. However, the cellular architecture enables sensor nodes to perform management tasks individually or in combined fashion, and reduce in-network communication and traffic for conserving the network energy. The management framework describes different management roles, management policies and different management tasks.

4.1 A Management Framework

We use an n-tier hierarchical framework for wireless sensor networks. However, the number of hierarchical levels is based on application type and number of nodes. To demonstrate the idea, we consider a 4-tier hierarchical framework to support network management in WSNs. The proposed management framework is flexible to accommodate the different applications of WSNs. The appliance of hierarchical structure is to specify different management roles and efficiently distribute network management tasks across the network. Instead of heavily relying on few central management entities (e.g. cluster-head nodes) or small portion of nodes, we encourage sensor nodes to evenly and efficiently share the management burdens for battery-energy conservation.

As described in chapter 3, the sensor network nodes configure themselves into a virtual grid structure, in which the network nodes are partitioned into several cells. The proposed cellular algorithm is then extended to a hierarchical architecture with nodes organized into different layers. In general, the performance and activities of low-level nodes is monitored and measured by the higher-level nodes as shown in Figure 4.1, which forms the hierarchical management structure in sensor networks.

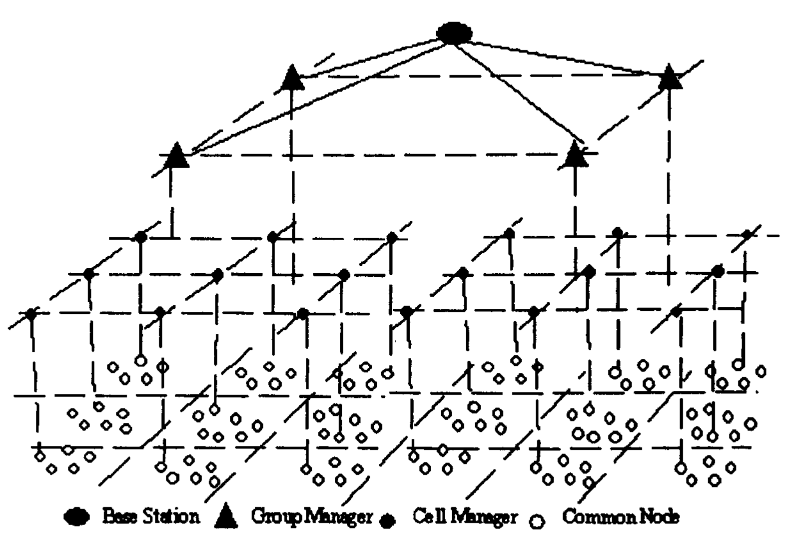

Figure 4.1: Hierarchical management framework

A cell manager is selected within each cell, and takes responsibilities for its cell management. Cells are aggregated into different groups, and each group is represented by a group manager. Cell managers shift parts of management tasks from group manager, and energy-efficiently monitor and control their one-hop communication cell members. Management information is collected and processed in each level of hierarchy, and only forwarded to the upper level on requests or by some special event. This design minimizes the communication messages, eliminates the redundancy of transmitted data, and thus conserves energy. Base stations lie at the top of the hierarchy, and are responsible for the overall network management. The managing nodes (group managers and cell managers) provide different levels of management capabilities to monitor common nodes. Base stations can also select and group a subset of sensor nodes to perform specific tasks such as communication or coordinated computation tasks. Thus, the participating sensor nodes only require to co-ordinate with their neighboring nodes. This minimizes a large amount of redundant data instead of always routing management messages back to the base station.

The cell managers within the same group represent a virtual grid structure towards their group manager as shown in Figure 4.1. Instead of frequently flooding keep-alive messages across the group and polling information from hundreds of thousands nodes, the group manager contacts its cell managers in the virtual grid structure to track the cell condition of its group nodes. Following the same process, group managers from different groups form another virtual grid structure towards the base station in the management hierarchy. The base station relies on group managers to track the residual status of the sensor network. At the top of the management hierarchy, the base station has the overview of the sensor network by accumulating the received topology information from the group managers. Thus, it has sufficient information to direct the group re-formation or group merging actions if the working nodes of certain groups have dropped to a critical level.

4.2 The Management Hierarchy

First, we introduce our management hierarchy. We assume a homogenous network where all the nodes in the network are equal in resources and no node should be more resourceful than any other node. We classify four management roles in the network. In general, the performance and activities of low-level nodes is monitored and measured by the higher-level nodes as shown in Figure 4.2, which forms the hierarchical management structure in sensor networks.

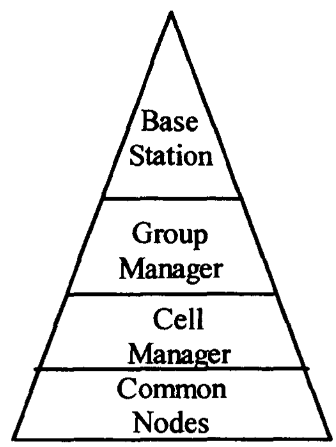

Figure 4.2: The hierarchical management architecture for WSNs

1) Common node

Common node is the basic unit / element to support the performance and activities of the sensor network. It is primarily responsible for its own management activities. They organized themselves into cells, sense and relay real-life measurements toward their monitoring nodes. They rarely involve themselves into group management tasks for conserving their limited battery energy. The process of data dissemination (routing) is separated from the process of data discovery. The common node has the ability to perform both these roles (one at a time). Some responsibilities of common nodes are as follows:

- Authenticate its cell manager
- Maintain connectivity to its monitoring node
- Respond to the monitoring node's command

2) Cell manager

A node is elected among a small amount of sensor nodes within a certain group. It is adopted to shift some management burdens (such as: network topology formation) from the group manager. It is more energy-efficient for cell managers to locally monitor and manage a small number of sensors. In addition, cell manager has a quick / fast response towards events occurred in the network. Some responsibilities of cell managers are as follows:

- A cell manage can act as a relay for the traffic generated by the sensors in its cell or perform aggregation/fusion of collected sensors data.
- Detect faulty nodes in its cell
- Represent the cell and send warning messages to its group manager
- Allocate data transmission slots (Schedule transmission)
- Send redundant nodes to low computational mode

3) Group manager

A node is elected among a group of cell managers, and is responsible for management of its group. It reports the residual energy of its group (including the network connectivity and sensing coverage rate) to the base station. Group manager has the knowledge and capability to handle some tasks in its group without consultation from the base station.

- A group manager can act as a relay for the traffic generated by the cells in its group or perform aggregation/fusion of collected sensors data.
- Monitor cells in-terms of energy
- Managing cell re-configuration and node mobility (will be discussed in chapter 5 and 7).

4) Base station

Base station acts as a gateway between sensing application and the WSNs. It collects the aggregated information from the group managers and integrates them into an information model. Based on such information model, the base station controls and balances the entire wireless sensor network by specifying and dispatching high-level management specification into the network. It usually contacts few nodes (e.g. group managers) to track the required information.

4.3 Network Monitoring, Diagnosis and Recovery

A management system must be able to determine the correctness of the sensing application during its deployed life time. It requires the propagation of network diagnostic information to a designated node for processing and analysis. Furthermore, the system must be able to store certain events for real-time or post-mortem analysis [Koliousis 2007]. We adopt a hierarchical layered-based system to gather network information. This is important to reduce the communication overhead imposed by the diagnostics management mechanism over the managing nodes.

Monitoring of sensor nodes can be active monitoring or passive monitoring. Active monitoring involves the existence of periodic messages and results in an implicit detection of a failure. On the contrary, passive monitoring triggers the alarm when an event occurs. Our management system exploits both active and passive measurements to detect failures in the networks (depending on application type). In active monitoring, sensor nodes periodically send keep-alive messages to their cell managers to confirm their existence. If the cell manager does not receive the update message from a sensor node after a pre-specified period of time, it may believe that the sensor is dead. In periodic monitoring model, nodes themselves notify the managing nodes of their residual energy (if it's below the required threshold value). Some applications may prefer passive monitoring as they introduce no additional bandwidth overhead. However, active monitoring can be more useful for some applications to determine the root cause of a failure. Passive monitoring plays an important role in our proposed reconfiguration algorithm i.e. for registering node status change, new nodes drifting in to the network and cell merging.

In our proposed framework, diagnostic information is stored by managing nodes at different levels of the hierarchy i.e. cell manager hold information only of its cell members and group manager store diagnostic information only about its group cell managers. Diagnostic information is aggregated at each level of the hierarchy to reduce the management traffic. Our proposed management framework provides a flexible platform to support various application-specific data aggregation schemes. Base station collects the aggregated information from the group managers and integrates them into an information model. Based on such information model, the base station controls and balances the entire wireless sensor network. It constructs an aggregated map of the remaining energy levels for different regions in a network.

The aggregated diagnostic information is then used for the re-configuration of sensor networks. For example; the residual battery energy of a cell manager is not sufficient enough to continuously support its management role. To avoid the sudden death of the cell manager because of energy depletion, a new cell manager is expected to replace the cell manager. In addition, if there is no available candidate node that has sufficient energy to shift the cell manager role. Group manager will initiate the cell merging process to merge the event cell with the neighbouring cells and recover the network from a critical failure. Diagnostic information is used by managing nodes to recover from different types of failures i.e. common node failure, cell manager and group manager failure and relocating mobile sensor nodes to fill coverage holes.

4.4 Management Process

We consider a layer-based system to support our proposed self-organizing management framework. It provides various integrated functions for sensor nodes that handle network management. It is a lightweight management framework that supports WSN management. Also, it is self-adjustable and reconfigurable according to the management role changes.

4.4.1 Policy-based Management

We adopt a policy based management to support the design of self-organizable distributed management services in WSNs. These network policies describe the management behavior with certain execution conditions for sensor nodes in different management hierarchy. Based on such available information, sensor nodes can make local decision and respond to the occurred event directly without consulting the base station or central manager. This reduces the network traffic overhead and conserves energy to prolong network life time. Furthermore, network management policies specify the mapping between management roles and their corresponding functions or tasks in the management hierarchy.

4.4.2 Role Assignment

Role assignment plays an important role in group formation and management responsibilities of our proposed framework. These roles may be based on varying sensor nodes properties (e.g. available sensors, location, network neighbors) and may be used to support application based on homogenous or heterogeneous functionality. Our goal is to break initial symmetry and assign specific roles to individual sensor nodes based on their properties and enable sensor nodes to co-ordinate with each other to reassign their management responsibilities in the network. Based on the assigned roles, sensor nodes may adapt their behavior accordingly and establish cooperation with other nodes. Candidate nodes are elected optimally according to their current hardware status, such as battery energy level. This enables the network management system to assign tasks based on nodes optimal capabilities rather than relying on specific nodes with extra resources. The basic idea of this design is to encourage nodes to be more self-manageable and extend the network life time for as long as possible. Network dynamics and real-time node changes (such as energy depletion) always trigger the management role reconfiguration of sensor node in a group. For example, a cell manager may degrade to a common node when its energy drops to a certain value.

4.4.3 Management Functional Units

WSNs are embedded in applications to monitor the environment and act upon it. Thus, it is important for the management application to be compatible with the kind of application being monitored. In order to have better development of WSN management services and functions, it is necessary to characterize the WSN and establish a novel management dimension. Thus, looking at the characteristics of various WSN applications, we have proposed five major functional units for our proposed management framework. These function units are used for individual management tasks and special needs of sensor applications. Thus, sensor nodes selectively choose function units according to their management role assignment in the network. Execution of management functions depend on various conditions described by network policies. The conditions for executing a function are retrieved from the real-time nodes and network state. Management functionality of a node is also reconfigurable to reflect the role changes.

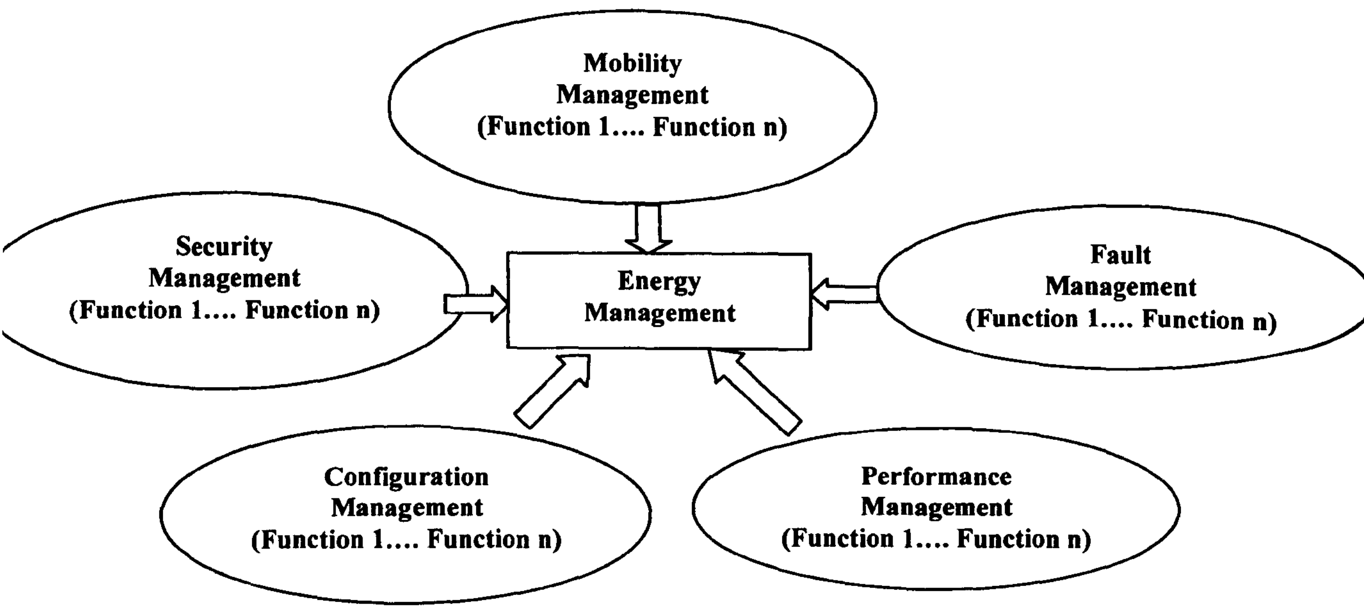

Figure 4.3: Management unit

Figure 4.3 shows the major functional areas of our proposed framework, and represents the relationship among management services and management functionalities.

1) Configuration management

Energy management unit in collaboration with configuration management unit plays an important role in managing the energy consumption of a sensor node, and control the real-time node management function execution, i.e. faults or mobility management. A configuration management service includes the self-organization and self-configuration of sensor nodes. It collects information about the network status and based upon that information it reconfigure the network. Wireless sensor networks are prone to network dynamics such as node dying, being disconnected, node power on or off, and new nodes joining the network and so the nodes need to be able to self-reconfigure themselves without knowing anything about network topology in advance.

Sensor nodes are usually operated on a limited battery and keeping the sensor nodes active all the time will limit the duration that battery last. Sensor nodes must enter a sleep mode when possible to preserve energy, consequently, extend the network life time. The hardware components of a sensor node have different power modes. Existing approaches like dynamic power management [Sinha 2001], or an agent-based power management [Tynan 2005], control valuable energy consumption of a sensor node by configuring node hardware setting (e.g. CPU performance), or switching nodes into 'sleeping' mode when energy level is below certain thresholds [Yu 2008]. Sensor nodes status change involves reconfiguration and must be dealt with energy efficiently.

2) Fault management

Most of such networks are deployed in an uncontrolled hostile environments and unmonitored operation area, where the physical presence of human administrators is impractical. For this reason, faults are frequent and unexpected in WSNs. Sensor nodes failure may cause network partitioning, connectivity loss and coverage holes. Node and network faults critically affect sensor networks management. Node faults may disconnect management data structure. Network faults may cause management data being lost or unavailable while other hardware and software faults may produce corrupted or incorrect management data. Therefore, appropriate measures and action must be taken to recover sensor network from failures. We address this problem by introducing a fault management unit in our proposed management framework.

3) Mobility management

One of the main objectives of WSNs is to achieve the desirable coverage. Moreover, if there is coverage hole in the network, the data transmission path through the coverage hole will be broken and needed to rebuild to avoid data loss. In order to rebuild the coverage hole, mobile nodes can be moved to improve coverage in certain areas of the network. This involves in finding redundant mobile nodes in the network and replaces the faulty nodes as soon as possible. This process is called sensor relocation. Our mobility management unit supports sensor relocation to deal with coverage holes in the network.

As we discussed earlier that configuration management deals with network dynamics such as node dying due to energy exhaustion, being faulty, and new nodes joining the network to improve coverage. Fault management and mobility management co-operate with configuration management to deal with network dynamics and to recover connectivity in wireless sensor networks.

4) Performance management

Performance management is needed to monitor the performance of the network and optimize it in terms of resource consumption and quality of service (QoS) requirements. One of the major performance issues of the WSN is the event reliability which is defined as the number of unique data packets received by the sink node. For the optimum performance the management system sets the data aggregation rate of the sensors and also keeps some nodes in sleep state and other in the normal state. The configuration (in terms of sensor capabilities, number of sensors density, node distribution, self-organization, and data dissemination) plays an important role in determining the performance of the network. Performance management must consider the performance of the network and provide services that are best measured in terms of meeting the accuracy and delay requirements of the observer, as well as consumed energy [Ilyas 2004].

5) Security management

The limited resources of a sensor node and its different characteristics from those of a traditional computer make it difficult to use traditional security techniques for WSNs. A WSN is subject to different safety related threats, i.e. eavesdropping on the communication channel; an adversary can easily intercept and alter messages. Information or resources can be destroyed; Information can be modified; stolen, removed, lost, or disclosed and service can be interrupted.

The list of potential applications of WSN that require protection mechanisms includes early target tracking and monitoring on a battlefield; law enforcement applications; automotive telemetric applications; measuring temperature and pressure in oil pipelines and forest fire detection. All these applications have unlimited benefits and potential. However, if the sensor information is not protected properly, possible compromises in user information, the environment, and even physical actuators could result. Security management is extremely important for resource constrained WSNs, and must provide self-protection, reliability, disposability, privacy, authenticity, and integrity [Ilyas 2004].

In this thesis we will concentrate on configuration, fault and mobility management. Performance and security management are beyond the scope of this thesis and will not be discussed.

4.5 Summary

Wireless sensor networks have become an emerging new research area in the distributed computing environment. However, we still need significant efforts to address a set of technical challenges. One of the major challenges is to design efficient network management architectures for continuously maintaining the network efficiency with minimal human intervention. In this chapter, we proposed a hierarchical cellular based system to support a self-organized WSN management architecture. The proposed layer-based hierarchical architecture supports sensor nodes to perform management tasks individually or in a cooperative fashion. This approach reduces in-network communication and traffic for conserving the valuable network energy.

The proposed management framework will be discussed based on three core functional areas i.e. configuration management, fault management and mobility management. The configuration part of the framework supports WSN to autonomously adjust its management structure and efficiency according to network changes. Configuration management unit deals with network dynamics to recover connectivity in the network. Fault management enables the network to detect faulty nodes and maintain connectivity. Mobility management participates in finding redundant sensor nodes to address the network connectivity, coverage, and network life time problems in WSNs.

5. Configuration Management of Wireless Sensor Networks

Configuration management is a first functional area of high importance in our proposed WSN management model. The configuration management must provide basic features such as self-organization, self configuration and self optimization. The network configuration management services collects information about the network states and based upon that information it reconfigures the network [Ilyas 2004]. Sensors must operate in many different types of environment in the absence of human administration. Configuration management indicates the action, not the requirement, to assert autonomy in a system after the initial deployment and while awaiting maintenance. As such, a sensor can re-configure its state given partial or complete knowledge of the network [Koliousis 2007]. Wireless sensor networks are prone to network dynamics such as node dying, being disconnected, node power on or off, and new nodes joining the network and so the nodes need to be able to self-reconfigure themselves without knowing anything about network topology in advance [Lee 2006a].

In this chapter we propose a new re-configuration algorithm to energy efficiently re-organize the network topology due to network dynamics. According to our literature survey we found that clustering is a very promising technique that could be used to tackle the WSN energy constraint by involving sensor nodes in communication within a particular cluster and exploiting the sensors maintenance possibilities to self-configure the network due to network dynamics. Most of the existing clustering approaches use flooding to create the clusters. However, if the clusters are reconfigured by flooding as the initial cluster creation process, then such a re-clustering process is very costly for the sensor network in term of energy. We have therefore proposed a new re-configuration algorithm to energy efficiently re-configure and maintain the network by adopting a localize criteria and in a distribution fashion. It avoids message flooding for cluster reconfiguration and maintenance.

5.1 Related Work

The problem of self-configuration has been a hot topic of research in wireless sensor networks. The study in [Venkataraman 2007] developed a size-restricted cluster formation and cluster maintenance technique for mobile ad hoc networks. The algorithm used a size restriction S, for cluster formation and cluster maintenance. In addition, while forming the clusters they also use a diameter restriction K. After cluster formation, a cluster maintenance mechanism is employed to deal with network dynamics i.e. new node joining a cluster, node leaving a cluster, election of cluster head, and cluster merging and splitting. In this algorithm, nodes in a cluster are classified into four types: boundary node, pre-boundary node, internal node and the clusterhead. Cluster maintenance is complicated and difficult in this algorithm.

In [Subramanian 2000] authors proposed a self configuration architecture that leads to hierarchical network with auto-configuration and a number of other useful properties. It is a generic architecture for a specific subclass of sensor applications which they defined as self-configurable system where a large number of sensors coordinate amongst themselves to carry out a large sensing task. Their self organizing algorithm lists four phases of operation. These are the discovery phase, organizational phase, maintenance phase, and self reorganization phase. Re-organization occurs as a result of node failure, link failure, group partition, or node rediscovery. Their network architecture is based on heterogeneity of sensor nodes while our proposed algorithm is based on homogeneity of sensor nodes.

In [Uchida 2008] authors proposed a new cluster-based architecture for the maintenance of wireless sensor networks, with two atomic operations node-move-in and node-move-out which are performed by appearance and disappearance of a node. [Wen 2006] developed a dynamic decentralized algorithm for re-clustering the sensors of an ad-hoc sensor network. Each sensor uses a random waiting timer and local criteria to determine whether to form a new cluster or to join a current cluster. The clusterhead reselection process is triggered when the energy reserves of the clusterhead falls below a threshold.

5.2 Configuration Management Algorithm

In this section, we present a re-configuration algorithm as a part of our configuration management to support sensor networks to energy-efficiently re-organize the network topology due to network dynamics such as node dying, node being disconnected, node power on or off, and new nodes joining the network. We utilized our hierarchical cellular architecture for sensor networks configuration management. As we have discussed previously, the group manager is capable of self-managing its group performance without the consultation from the base station. With the assistance of its cell managers, the group manager is capable of tracking the residual status of sensor nodes in its group, and spontaneously responds to events (e.g. node failure, node movement) occurred in the network.

Our proposed scheme is based on homogenous sensor nodes to support the balanced distribution of managing nodes and to extend the network life time. In order to re-organize the cell, the proposed algorithm sets limits on the cell size. The cell size is a user-defined parameter, which can be adjusted to meet the required cell manager density. Thus, the cluster size is a key parameter to achieve balanced load among clusters and to keep hierarchical structure efficient. Our proposed algorithm is based on message filtering to lessen the redundant message broadcast in the cluster maintenance process for energy conservation. We applied localized criterions for cluster re-configuration and maintenance in a distributed fashion.

Configuration management includes: the mechanism to generate the topology of the network, followed by reconfiguration of the cell.

5.2.1 Topology Generation

Cell managers frequently send update message to their group managers that present the residual energy level of cell members, and the number of available nodes in the cell. After retrieving and aggregating update messages from cell managers, the group manager has an overview of its group status and constructs a topology map. Thus, it is capable of taking the proper actions (e.g. altering the cell formation) responding to the events or changes in the groups. To resume and maintain the network performance, the boundary of virtual cells is capable of merging together to produce a large cell if the status of network connectivity and sensor coverage rate have dropped to a critical level.

Following the same process, group managers from different groups form another virtual grid structure towards the base station in the management hierarchy. At the top of the management hierarchy, the base station has the overview of the sensor network by accumulating the received topology information from the group managers. Thus, it has sufficient information to direct the group re-formation or group merging actions if the working nodes of certain groups have dropped to a critical level.

5.2.2 Cell Re-configuration

After partitioning the network into a virtual grid, cell maintenance techniques are employed to manage network dynamics.

Cell re-formation includes:

1) Node joining a cell
2) Node status change
3) Cell manager status change
4) Cell merging

1) Node joining a cell

This can be the result of new nodes drifting into the network or mobile nodes moved in to fill the coverage hole. Figure 5.1 depicts the join algorithm, where node 8 is the new node that wanted to join Cell 1. The new node 8 will first determine the cell id using its co-ordinates and then broadcast a Join_Msg. The join request message consists of the node id, location of the node and the cell id it wishes to join. The cell id in Join_Msg avoid flooding of the message i.e. the node will discard the message if it does not belong to its cell. When cell manager (node 2) receives the Join_Msg from node 8, it first checks the node co-ordinates to make sure that the new node joining request is a valid request. The cell manager then replies back with the Join_Acc message. Also, the cell manager then informs its cell members about the arrival of the new node by broadcasting a New_Node_Join message. The New_Node_Join message contains the new node id.

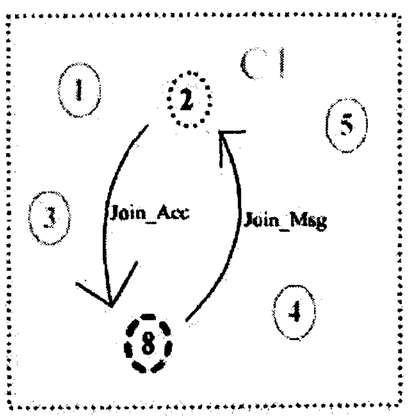

Figure 5.1: Node joining a cell

2) Node status change

Node status change means; when a node cannot carry on its normal operation due to low residual energy and needs to change its state to a low computational mode. It is very important that a node should monitor its residual energy and take appropriate measure before it completely shuts down. Figure 5.2 depicts the node status change algorithm, where node 8 wants to change its status and broadcast a StatusChange_Msg to its cell members. The Status change messages consist of node id and cell id. It waits for an acknowledge message (StatusChange_Ack) from its cell manager. The cell manager then informs its cell members about node 8 status change.

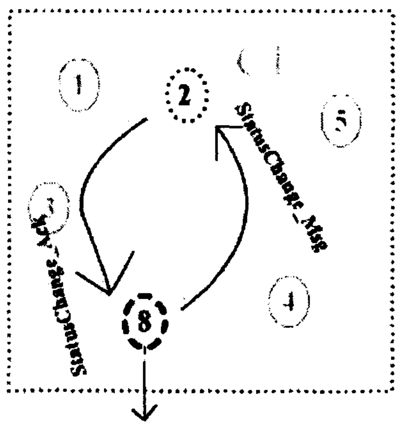

Figure 5.2: Node status change

3) Cell manager status change

When a cell manager wants to change its status due to low residual energy, a new cell manager is required to take the responsibility. The most intuitive way to elect a new leader is to re-cluster the network. However, re-clustering is not only a resource burden on the network nodes but it is often very disruptive to the ongoing network operation. Therefore, we employed a second in command node (secondary manager) to replace its cell manager when required. A cell manager broadcasts a StatusChange_Msg to its cell members, which is an indication for secondary cell manager to standup as a new cell manager.

***4)* Cell merging**

This is an important part of configuration management and plays a vital role in prolonging network life time. In this scenario, we assume the cell manager in the event cell is under a critical condition. The residual battery energy of this cell manager is not sufficient enough to continuously support its management role. To avoid the sudden death of the cell manager because of energy depletion, a new cell manager is expected to replace the cell manager. In addition, there is no available candidate node that has sufficient energy to shift the cell manager role. Therefore, sensor nodes in the event cell are expected to join the neighbouring cells for maintaining the network connectivity in that specific area. The event cell manager informs the cell managers within the neighbour cells to broadcast the Join_In message to merge the remaining nodes in the event cell.

The cell merging has three stages as follows:

- The cell managers of neighbouring cells broadcast a 'Join_in Message' to sensor nodes in the destination cell.
- The 'Join_in Message' of neighbouring cells is delivered to all of the sensor nodes in the event Cell to notify the available cell managers.
- Sensor nodes in the destination cell select the appropriate neighbouring cell to join in by checking the minimum_hop_count and the residual energy of source cell managers. The nodes then reply acknowledge message to the selected cell manager once they have accepted a cell manager.

Figure 5.3 shows a small portion of virtual cells within a group. It demonstrates how the neighbouring cell managers broadcast messages to merge sensor nodes in the event cell (e.g. cell 4).

As shown in Figure 5.3, node 'a' might receive three 'Join_In' messages from its neighbouring cell managers (as Cell 1, Cell2, and Cell3).

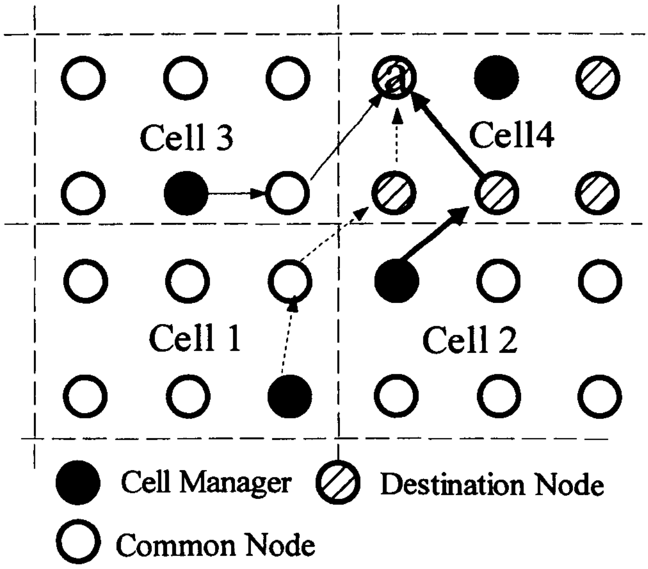

Figure 5.3: Cell managers broadcast Join_in Message to destination node

Using Figure 5.3 the cell merging algorithm is as follows:

- Each node in the event cell (e.g. cell 4) has been aware that there is no available node to take over the cell manager role. They are waiting for the 'Join_in' messages from their neighbouring cells.

- Cell managers of neighbouring cells start to broadcast 'Join_in' messages and wait for the acknowledge messages from the nodes.

- After received 'Join_in' messages, a node first checks whether it belongs to the event cell as declared in the 'Join_in' message. If not, it modifies the hop count of packet before it rebroadcasts.

- A node in the event cell (e.g. cell 4) records the information of cell manager such as: cell_id, node_id, residual energy of the source cell manager, and also the number of communication hops when it receives the 'Join_in' message. Thus, the node modifies the source and the hop count of the packet before it rebroadcasts.

- When the node receives a 'Join_in' message from the same source cell manager again, they drop the packet for reducing the redundant messages transmitted in the network.

- If the node in the event cell receives the 'Join_in' messages from different cell managers (i.e. from cell 1, cell 2, and cell 3). It selects the right cell manager to join by comparing the hop count towards the source cell managers and the residual energy of the cell managers. It will select the cell manager with fewest hops and sufficient residual energy.

- A node waits for a random delay period whenever it decides to rebroadcast.

- If the drop count exceeds threshold C, then the rebroadcast is cancelled.

Figure 5.4 Demonstrate the flow chart of cell merging algorithm

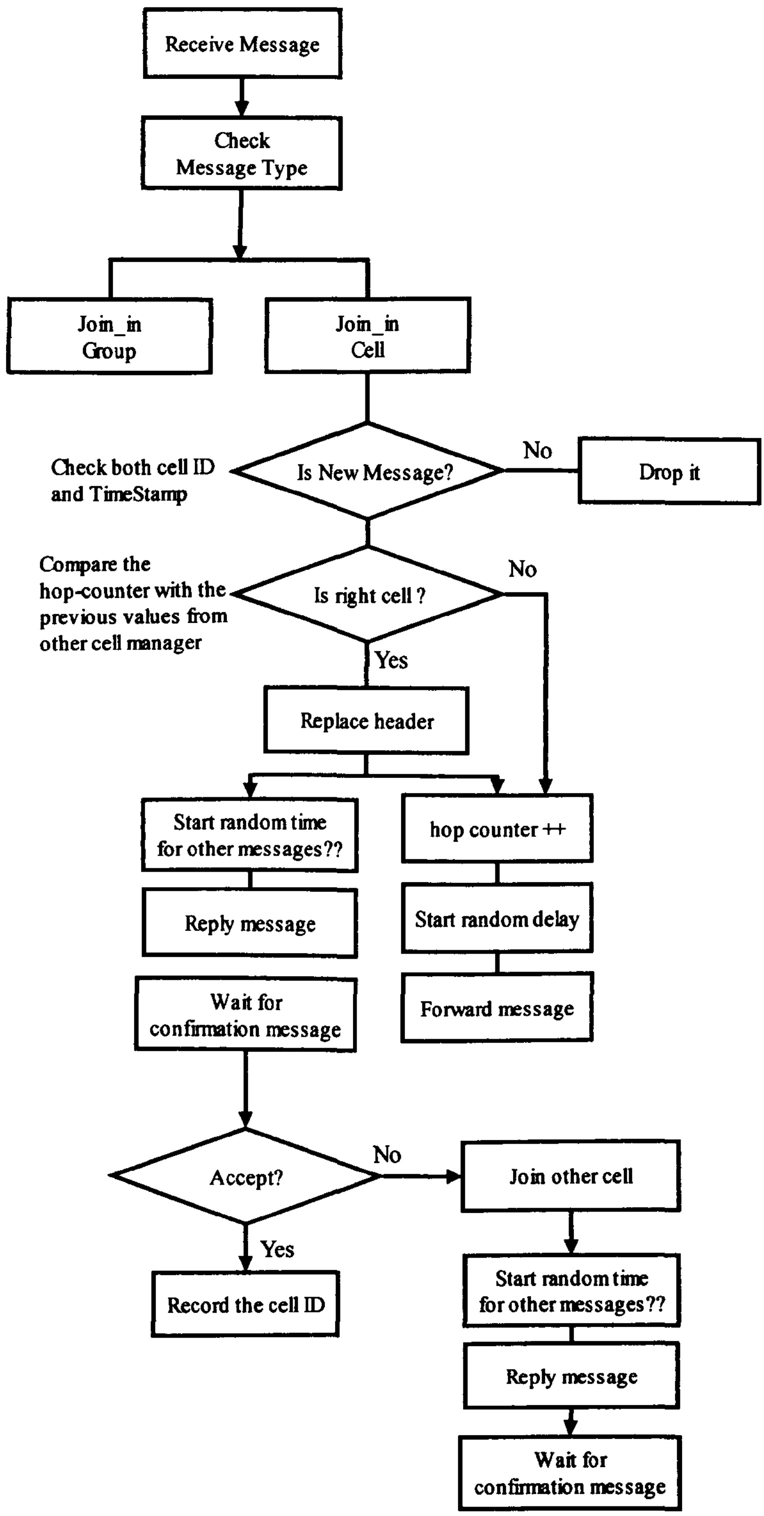

Figure 5.4: Cell merging message handling algorithm

5.3 Message Broadcast and Collision Issue

The proposed cell reformation algorithm relies on the message exchange among sensor nodes in the network. This might subsequently cause the communication flooding by broadcasting or re-broadcasting messages (including the reply messages) from different sensor nodes. In this section, we examine these problems as follows:

Addr	GROUP	MS_TYPE	....	DATA	....

Figure 5.5: Message format

We employed a message filtering mechanism to further reduce the redundancy of message exchange (as discussed in chapter 3). We apply the MS_TYPE field contains three types of values, referring separately to Join_in message, reply message, and warning message. Each value links with a specific event handler, which processes the received message individually. We particularly focus on the discussion of the Join_in and reply message.

The 'GROUP' field, containing the value of group id, and is applied to distinguish and avoid receiving messages from other groups. The 'DATA' field is actually the structured message packet (i.e. both Join_in message and reply message) as shown in Figure 5.6.

```
class Join_In Message
{ public:
    IPAddr_t  src;          // source address
    NodeId_t dest_cellid;        // the destination cell id;
    NodeId_t  groupid;      // the group id
    NodeId_t  mnid;         // the managing node id
    Count_t   hop_cn;       // record the communication hop
    Time_t    timestamp;    // the message sending out time
}
class Reply Message
{ public:
    IPAddr_t  src;          // source address
    IPAddr_t  des;          // destination address
    NodeId_t  groupid;      // the group id
    NodeId_t  mnid;         // the managing node id
    Count_t   hop_cn;       // record the communication hop
    Time_t    timestamp;    // the message sending out time
    Energy_t  curEnergy;    // the current node battery energy
}
```

Figure 5.6: Data field attributes

The 'groupid' and 'mnid'(cell manager id) fields in 'DATA' field describe the group and cell manager node that the source node belongs to. The 'hop_cn' counts the communication hop between the source node and current node; or between the source node and destination node if it is the reply message. The 'curEnergy' field holds the information of node residual energy. It is applied to notify the nodes in the event cell about the current energy status of neighbouring cell managers. In addition, it is used by the group manager and cell manager nodes to determine whether the source node is in a critical status and has higher possibility to cause the network connectivity failure.

Three stages of message filtering have been applied to our proposed scheme to lessen the redundant broadcast in the network for energy conservation i.e. the message type, and timestamp.

The message type stage first check the 'GROUP' field to quickly determine whether the received message belongs to the same group of current node. If not, the message will be dropped to avoid unnecessary message re-broadcasting. It then checks the

'MS_TYPE', distinguishing data packet between the Join_in and reply messages. After retrieving the value of 'destination_cell_id', the node decides whether it belongs to the event cells (e.g. destination cell) to process the message. Otherwise, it will rebroadcast the message after modifying the hop count value of the packet.

A sensor node might receive multiple copies of the same message forwarded by different intermediate nodes. To avoid redundant rebroadcast, we apply the value of 'timestamp' field in the second stage to determine whether the receiving message has been handled previously. If the receiving message is a new one, it will be processed and forwarded to the neighbouring nodes. On the contrary, that message will be dropped to lessen the network traffic and conserve the node energy. Figure 5.7 demonstrate the message redundancy control.

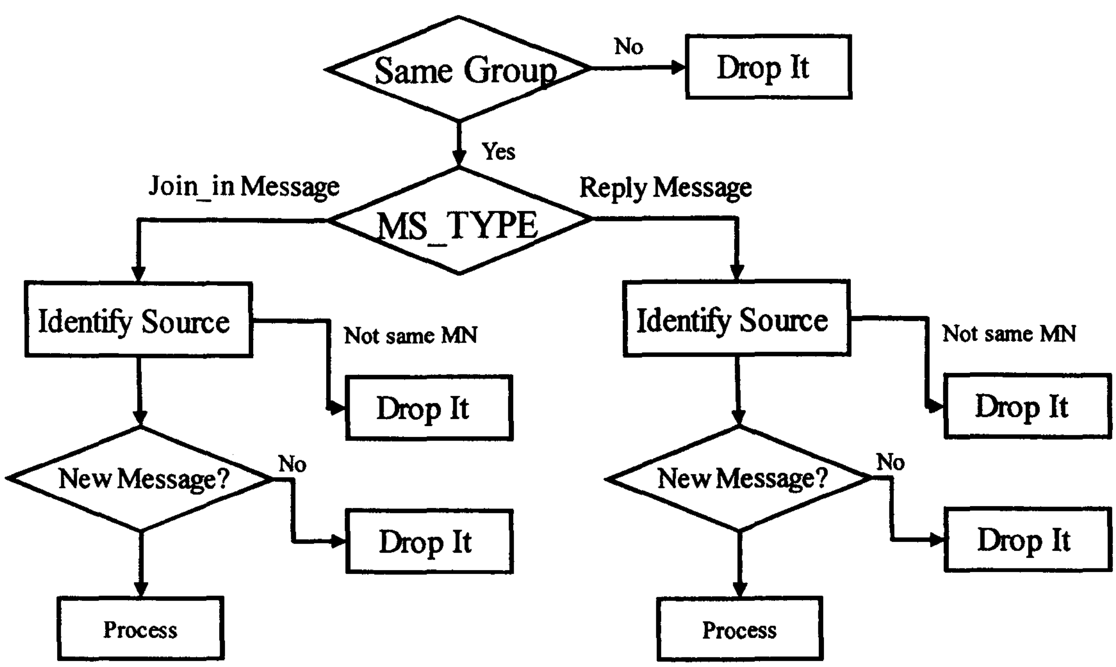

Figure 5.7: Message redundancy control

However, message broadcast collision may result from the simultaneous broadcast and replying towards the same destination nodes. To solve this problem, we apply a random delay time in the sensor node before it rebroadcasts or replies to the messages. This lowers the possibilities of message collision because of the simultaneous message exchange.

5.4 Experimental Validation

In this section we evaluate the performance of our proposed algorithm. We used the same energy model as discussed in chapter 3. Number of sensors varied from 40 to 80, which are randomly deployed over 150 X 150 square meter area. The experiment assumed that channel allowed collision and that packets could be dropped in the medium. Sensors are given IDs in random fashion. All nodes are considered same and no preference is given to any sensor. We compared our work with autonomic self-organization algorithm [Chen 2007a] during cluster merging and re-clustering.

1) Cell merging

Autonomic algorithm is a 3 tier hierarchical management system. Higher level nodes (headers) may cause low level nodes to be clustered in an inhospitable environment by the actuator using wireless communication. High level nodes broadcast a 'cover request' message periodically. The lower level nodes select a clusterhead based on minimum hop count value. This involves flooding the network with 'cover request' messages to form a network. If a header dies or depletes its energy then all its cluster members have to join a new header based on minimum hop count value. For example in figure 5.3, cluster 4 header is no longer available to perform its normal operation (i.e. failed) and therefore, all its member need to join a new header. Cluster 1, 2, and 3 header will send a 'cover request' messages to all the members of cluster 4. Based on minimum hop count value, cluster 4 members will select a new header for themselves. For example, they joined cluster 1 header due to minimum hop count. But cluster one header is also low in energy and soon need to go off. This initiate another re-configuration phase as all the newly added nodes from cluster head 4 and cluster 1 member required a new header to carry on their normal operations. Therefore, using only hop count parameter for cluster head selection is not an energy efficient approach.

We used both hop count and residual energy in conjunction for the selection of new header. Along with checking the hop count value, the node will also check if the header has enough residual energy. If the header has low minimum hop count value but does not have enough residual energy then it will not be selected as a new header. A node selects a header, if its hop count value is low and also has sufficient residual energy. A threshold value for residual energy can be specified at deployment stage. This helps in prolonging network life time and avoiding extra flooding for re-configuration.

We simulated the cluster merging phase for both algorithms and it can be observed from figure 5.8 and figure 5.9 that our proposed algorithm consumed less energy than the autonomic one. The autonomic algorithm performed merging twice due to the selection of new header with low residual energy (using hop count). However, our algorithm performed merging once and considered both hop count and residual energy for the selection of new cell manager.

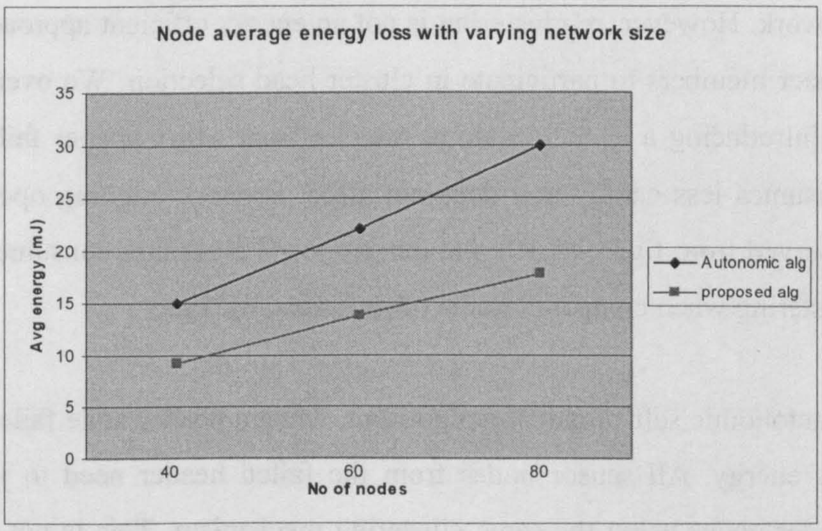

Figure 5.8: Node average energy loss with varying network size

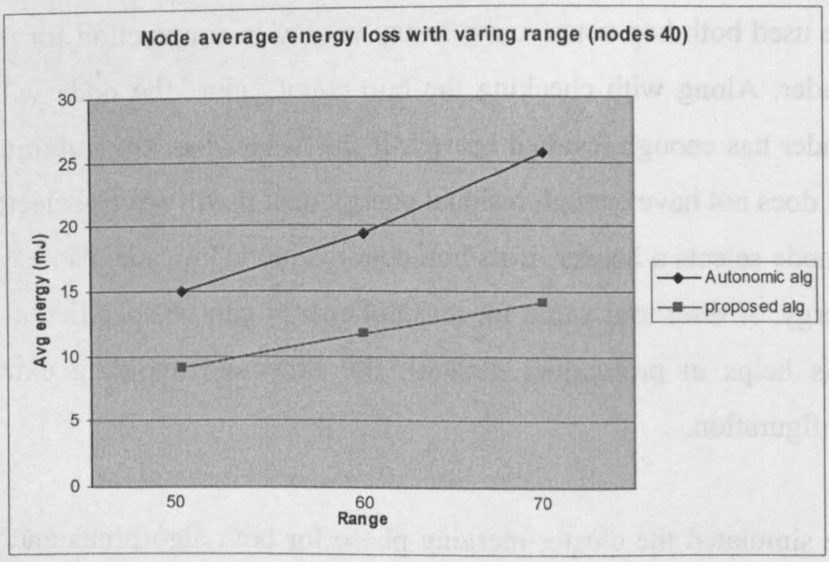

Figure 5.9: Node average energy loss in cluster merging with varying range

2) Re-clustering

The most common way to recover from a clusterhead failure is to re-cluster the network. However, re-clustering is not an energy efficient approach as it involves all cluster members to participate in cluster head selection. We overcome this problem by introducing a backup node to recover from cell manager failure. This approach consumes less energy and does not affect network ongoing operation. . It can be observed from figure (5.10) that our proposed algorithm consumes less energy in re-clustering when compared to the other autonomic one.

In autonomic self-organizing algorithm, when a header node failed or step down due low energy. All sensor nodes from the failed header need to join other available header nodes using the same clustering mechanism. This is not an energy efficient approach to recover from a cluster head failure.

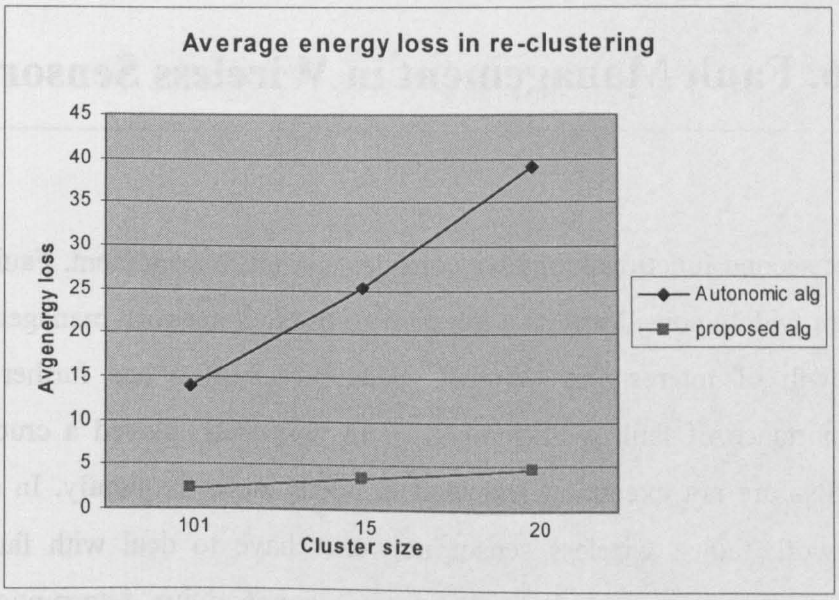

Figure 5.10: Average energy loss in re-clustering

5.5 Summary

In this chapter, we present an algorithm that supports sensor networks to energy-efficiently re-organize the network topology due to the network dynamics i.e. node dying, being disconnected, node power on or off, and new nodes joining the network. We adopted virtual cellular based hierarchical architecture for supporting self-organized group formation in wireless sensor networks. We also discussed a cell merging algorithm to maintain connectivity in the network. The results obtained from the simulation have shown that our re-configuration algorithm is more energy efficient than the autonomic algorithm.

6. Fault Management in Wireless Sensor Networks

The second functional area we consider is fault management. Fault management has been widely considered as a key part of today's network management. Recent rapid growth of interests in Wireless Sensor Networks has further strengthened the importance of fault management, or in particular, played a crucial role. Faults in WSNs are not exception and tend to occur more frequently. In addition to typical network faults, wireless sensor networks have to deal with faults arising out of unreliable hardware, limited energy, connectivity interruption, environmental variation and so on. Thus, in order to guarantee the network quality of service and performance, it is essential for WSNs to be able to detect failures and to perform something akin to heal and recover the network from events that might cause faults or misbehaviour. A set of functions and applications designed specifically for this purpose is called a fault management platform [Paradis 2007, Yu 2007].

One way of dealing with faults is to design a system that is fault-tolerant to begin with. Fault tolerance is the ability to maintain sensor networks functionalities without any interruption due to sensor nodes failure. However, this requires network designer to be fully aware, at design time, of the different types of faults and the extent to which they may occur once the network is deployed. The power supply is the most critical restriction as it is usually difficult to be rechargeable. For this reason faults occurs frequently and will not be isolated events. Attacks by adversaries could happen because these networks will be often embedded in critical applications. Worse, attacks could be facilitated because these networks will be deployed in open spaces or enemy territories, where adversaries cannot only manipulate the environment but gain physical access to the node. Also, communication in sensor networks takes place by radio frequencies means that adversaries can easily inject themselves in the network and disrupt infrastructure functions. Moreover, sensor nodes are commonly used to monitor external environment, due to which sensor nodes are susceptible to natural phenomenons like rain, fire and fall of trees [Asim 2008b].

Sensor network faults cannot be approached similarly as in traditional wired or wireless networks due to the following reasons [Paradis 2007]:

1) Traditional wired network protocol are not concerned with the energy consumptions as they are constantly powered and wireless ad hoc networks are also rechargeable regularly.

2) Traditional network protocols aim to achieve point-to-point reliability, where as wireless sensor networks are more concerned with reliable event detection.

3) Faults occur more frequently in wireless sensor networks than in traditional networks, where client machine, servers and routers are assumed to operate normally.

In this chapter we used our cellular architecture and describe a new mechanism to detect failing nodes and recover the connectivity in wireless sensor networks. We propose a localized cellular based method for detecting faults due to energy exhaustion of sensor nodes. This novel approach saves energy and improves network lifetime by detecting faulty sensor nodes locally and therefore reducing the number of transmissions required to convey the relevant information to the sink. In existing clustering scheme, the most intuitive way to elect a new leader is to re-cluster the network. However, re-clustering is not only a resource burden on the network nodes but it is often very disruptive to the ongoing network operation. In our scheme, the faulty sensor nodes are detected and recovered in their respective cells without causing any disruption to the ongoing network operation.The hierarchical management framework and node management role is also expected to be self-adjustable dynamically to the changes occurred in the network. For example, replacing the failed cell manager; shifting over some workload from the sensor nodes whose residual resource status is in a critical level.

6.1 Related Work

Existing fault management approaches for WSNs vary in forms of architectures, protocols, detection algorithm or detection decision fusion algorithm etc [Yu 2007]. A survey on fault tolerance in wireless sensor networks can be found in [Paradis 2007]. This section starts by reviewing the fault detection approaches, then we present fault diagnosis and failure recovery mechanisms.

6.1.1 Fault Detection

Since sensor network conditions undergo constant changes, network monitoring alone may not be sufficient to identify network faults. Therefore, fault detection techniques need to be in place to detect potential faults [Paradis 2007]. Generally, fault detection in WSNs has two types: explicit detection and implicit detection [Yu 2007]. The first one is performed directly by the sensing devices and their sensing applications. The implicit detection refers that anomalistic phenomena might disable a sensor node from communication or behave properly, and has to be identified by the network itself. Implicit detection is normally achieved in two ways: active and passive model. The active detection model is carried out by the central controller of sensor network. Sensor nodes continuously send keep-alive messages to the central controller to confirm their existence. If the central controller does not receive the update message from a sensor node after a pre-specified period of time, it may believe that the sensor is dead. Passive detection model (event-driven model) triggers the alarm only when failure has been detected. However this model will not work properly if a sensor is disabled from communication due to intrusion, tampering or being out of range. Fault detection mainly depends on the type of application and the type of failures. Some exiting fault detection schemes are discussed below. We classify the existing failure detection approaches into two primary types: centralized and distributed approach.

1) Centralized approaches

In centralized fault management systems, usually a geographical or logical centralized sensor node identifies failed or misbehaving nodes in the whole network. This centralized node can be a base station, a central controller or a manager. This central node usually has unlimited resources and performs wide range of fault management tasks [Yu 2007]. Some common centralized fault management approaches are as follows:

Sympathy [Ramanathan 2005a] is a debugging system and is used to identify and localize the cause of the failures in sensor network application. Sympathy algorithm does not provide automatic bug detection. It depends on historical data and metrics analysis in order to isolate the cause of the failure. Sympathy may require nodes to exchange neighborhood list, which is expensive in terms of energy. Also, Sympathy flooding approach means imprecise knowledge of global network states and may cause incorrect analysis.

The author in [Staddon 2002] enabled the base station to construct an overview of network by integrating each piece of network topology information (i.e. node neighbor list) embedded in node usual routing message. This approach uses a simple divide-and-conquer rule to identify faulty nodes. It assumes that base station is able to directly transmit messages to any node in the network and rely on other nodes to route measurements to the base station. Also, this approach assumes that each node has a unique identification number. This first step enabled the base station to know the network topology and for this purpose it executes route-discovery protocols. Once the base station knows the node topology it then detects the faulty node by using a simple divide-and-conquer strategy based on adaptive route update messages.

Centralized approach is suitable for certain application. However, it is composed of various limitations. It is not scalable and cannot be used for large networks. Also, due to centralized mechanism all the traffic is directed to and from the central point. This creates communication overhead and quick energy depletions. Moreover, central point is a single point of data traffic concentration and potential failure. Lastly, if a network is partitioned, then nodes that are unable to reach the central server are left without any management functionality.

2) Distributed Approaches

This is an efficient way of deploying fault management. Each manager controls a sub network and may communicate directly with other managers to perform management functions. Distributed management provides better reliability and energy efficiency and has lower communication cost than centralized management systems [Lee 2006a].

The algorithm proposed for faulty sensor identification in [Ding 2005] is purely localized. Nodes in the network coordinate with their neighboring nodes to detect faulty nodes before contacting the central point. In the scheme, the reading of a sensor is compared with its neighboring' median reading, if the resulting difference is large or large but negative then the sensor is very likely to be faulty. This algorithm can easily be scaled for large network. However, the probability of sensor faults need to be small as this approach works for large networks. Also, if half of the sensor neighbors are faulty and the number of neighbors is even, algorithm cannot detect the fault as expected. But the algorithm developed in [Chen 2006] tried to overcome the limitations of this approach by identifying good sensor nodes in the network and uses their results to diagnose the faulty nodes. These results are then propagated in the network to diagnose all other sensor nodes. This approach performs well with even number of sensors nodes and do not require sensors physical locations. This approach is not fully dynamic and is required to be pre-configured. Also, each node should have a unique ID and the center node should know the existence and ID of each node. Another scheme proposed in [Marti 2000], where sensor nodes police each other in order to detect faults and misbehavior. Nodes listen-in on the neighbor it is currently routing to and can determine whether the message it sent was forwarded. If the message it sent was not forwarded then it conclude its neighbor as a faulty node and chooses a new neighbor to route to.

The algorithm proposed in [Koushanfar 2002b] is a straightforward and simple mechanism where fault detection is based on the binary output of the sensors. In this approach, each node observes the binary output of its sensor and then compares it with the pre-defined fault model. Fault models can use probability or statistics to detect faulty sensors.

The author in [Venkataraman 2008] proposed a failure detection and recovery mechanism due to energy exhaustion. It focused on node notifying its neighboring nodes before it completely shut down due to energy exhaustion. The paper describes four types of failure recovery mechanisms depending on the type of node in the cluster. The nodes in the cluster are classified into four types, boundary node, pre-boundary node, internal node and the clusterhead. Boundary nodes do not require any recovery but pre-boundary node, internal node and the clusterhead have to take appropriate actions to connect the cluster. Usually, if node energy becomes below a threshold value, it will send a fail_report_msg to its parent and children. This will initiate the failure recovery procedure so that failing node parent and children remain connected to the cluster.

As we have seen, the distributed approach will be the design trends for fault management in WSNs. Sensor nodes gradually take more management responsibility and decision-making in order to achieve the vision of self-managed WSNs. Node self-detection scheme [Harte 2005] and neighbour coordination [Hsin 2006] have provided us a good example of management distribution, but their focuses are on a small region (a group of nodes) or individual node. Research work as MANNA [Ruiz 2004a], WinMS [Lee 2006b] etc proposed management architecture to look after the overall network from a central manager scheme. MANNA [Ruiz 2004a] is a policy-based approach using external managers to detect faults in the network. MANNA assigns different management roles to various sensor nodes depending on the network characteristics (Homogenous vs. heterogeneous). These distinguish nodes exchange request and response messages with each other for management purpose. To detect node failures, agents execute the failure management service by sensing GET operations for retrieving node states. Without hearing from a node, manager declares it as a faulty node. MANNA has a drawback of providing false debugging diagnosis. There are several reasons a node can be disconnected from the network. It can be disconnected from its cluster and not able to receive any GET message. GET message can be lost during environmental noise. Random distribution and limited transmission range can also cause disconnection. Also, this scheme performs centralized diagnosis and requires an external manager.

WinMS [Lee 2006b] provides a centralized fault management approach. It uses the central manager with global view of the network to continually analyses network states and executes corrective and preventive management actions according to management policies predefined by human managers. The central manager detects and localized fault by analyzing anomalies in sensor network models. The central manager analyses the collected topology map and the energy map information to detect faults and link qualities. It has the ability to self configure in case of failure, without prior knowledge of network topology. Also, it analyzes the network state to detect and predict potential failures and perform action accordingly.

6.1.2 Fault Diagnosis

In this stage, detected faults are properly identified by the network system and distinguished from the other irrelevant or spurious alarms. Fault diagnosis include fault isolation (where is the fault located), fault identification (what is the type of detected fault), and root cause analysis (what has caused the fault). However, there is still no comprehensive descriptive model to identify or distinguish various faults in WSNs, which supports the network system on accurate fault diagnosis or action-taken in the fault recovery stage [Yu 2007]. Existing approaches are based on hardware faults and consider hardware components malfunctioning only. Some assume that system software's are already fault tolerant as in [Chen 2006, Koushanfar 2002a]. The author in [Koushanfar 2002b] described two fault models. The first one corresponds to sensors that produce binary outputs. The second fault model is based on sensors with continuous (analog) or multilevel digital outputs. [Clouqueur 2004] proposed work only consider faulty nodes are due to harsh environment. Thus, there is a need to address a generic fault model that is not based on individual node level, but also consider the network and management aspects.

6.1.3 Failure Recovery

In this stage, the sensor network is reconfigured in such a way that failures or faulty nodes do not bring any further impact on the network performance. Most existing approaches isolate faulty (or misbehaving) nodes directly from the network communication layer. For examples, in [Marti 2000], after the failure of a neighboring node, a new neighboring node is selected for routing. WinMS [Lee 2006b], used a proactive fault management maintenance approach i.e. the central manager detect areas with weak network health by comparing the current node or network state with historical network information model (e.g. energy map and topology map). It takes a proactive action by instructing nodes in that area to send data less frequently for node energy consumption. In [Gupta 2003a], when a gateway node dies, the cluster is dissolved and all its nodes are reallocated to other healthy gateways. This consumes more time as all the cluster members are involved in the recovery process. The author in [Koushanfar 2002b] suggested a heterogeneous backup scheme for healing the hardware malfunctioning of a sensor node. They believe a single type of hardware resource can backup different types of resources. Although this solution is not directly relevant to fault recovery in respect of the network system level management [Yu 2007]. In consideration of complexity of fault management design and constrains of a sensor node, we are seeking a localized hierarchical solution to update and reconfigure the management functionality of a sensor node.

In this section, we highlighted different issues and problems existed in already proposed fault management approaches for WSNs. It is clear from the literature survey that different approaches for fault management in WSNs suffer from the following problems:

- Most existing fault management solutions mainly focus on failure detection, and there is still no comprehensive solution available for fault management in WSNs from the management architecture perspective.

- Different mechanisms proposed for fault recovery i.e. [Koushanfar 2002b], are not directly relevant to fault recovery in respect of the network system level management i.e. network connectivity and network coverage area etc.

- Failure recovery approaches are mainly application specific, and mainly focus on small region or individual sensor nodes thereby are not fully scalable.
- Some management frameworks require the external human manager to monitor the network management functionalities.

- Another important factor that needs to be considered is vulnerability to message loss. For example, in MANNA [Ruiz 2004a], if a clusterhead does not hear from its cluster member than it announced it as a faulty node. However, a message can be lost due to various reasons. It can be lost during transmission and cause a correct node to be declared as faulty.

We therefore contend that there is still a need of a new fault management scheme to address all the problems in existing fault management approaches for wireless sensor networks. We must take into account a wide variety of sensor applications with diverse needs, different sources of faults, and with various network configurations. In addition, it is also important to consider other factors i.e. mobility, scalability and timeliness.

6.2 Fault Model

To facilitate the self managing capability of our proposed fault management scheme, we proposed a fault knowledge model to support sensor nodes responding to network faults. This knowledge model describes different types of faults for our proposed fault management scheme.

We classified the node fault into two types: permanent, and potential. The permanent fault completely disconnects the sensor node from other nodes, and brings eternal impact on the network performance. For example, hardware faults within a component of a sensor node. A permanent fault once activated remains effective until it is detected and handled. The impact of this failure is usually measured when assessing the network performance. On the other hand, a potential fault usually results from the depletion of node hardware resource, i.e. battery energy. Such fault might cause the node sudden death, and eventually threaten the network life time. When the battery depleted, a node is useless and cannot share in sensing or data dissemination. Potential failure can be detected and treated before it causes the sudden death of a node e.g. sensor node with low residual energy can be sent to sleep mode before it completely shuts down and disrupts network operation. Faults can be further classified into: node level fault and network level fault. We proposed a fault model in a tree structure to describe faults monitored in sensor network. As shown in Figure 6.1, "node level" represents the potential and permanent failure of a node while "network level" describes the network faults caused by either potential or permanent failure of one or a set of sensor nodes.

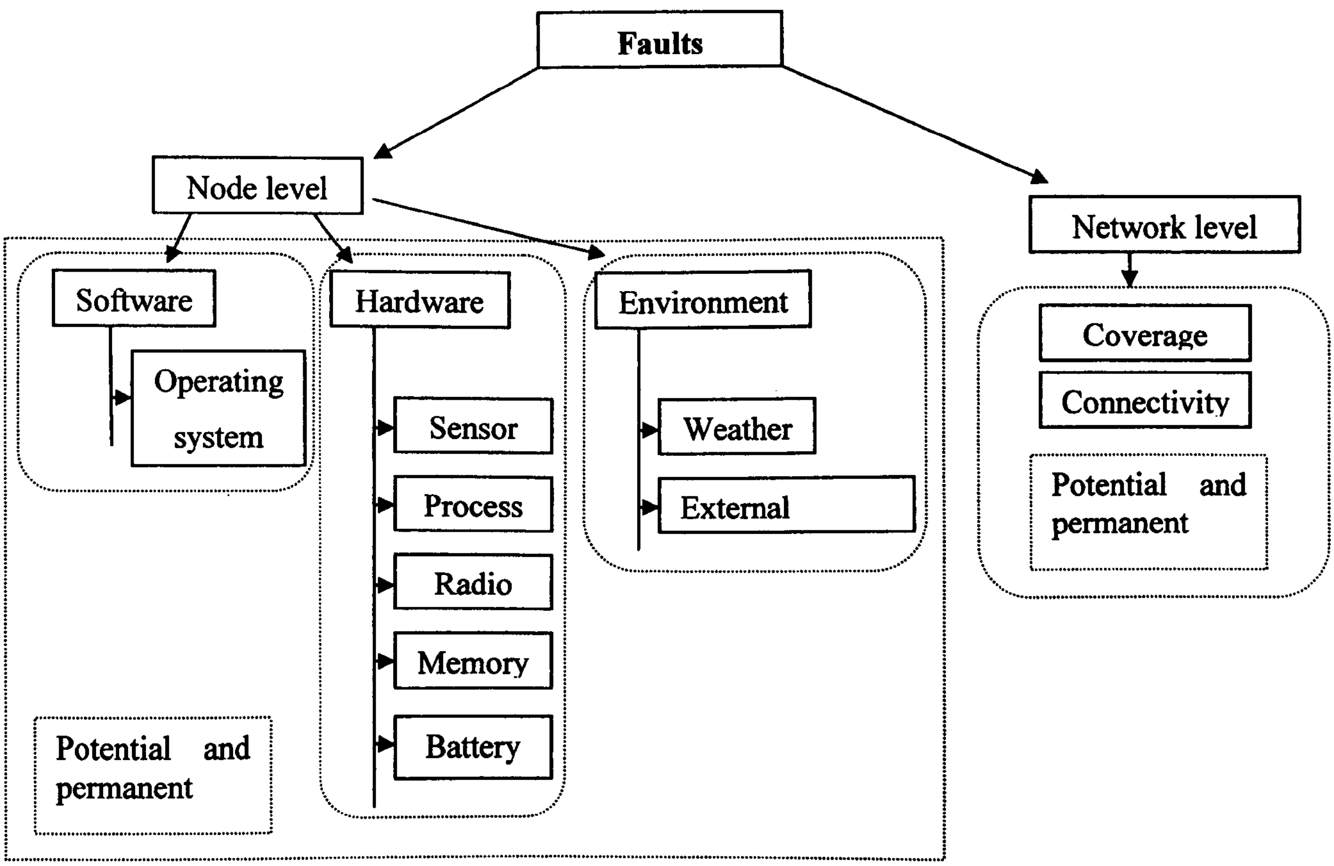

Figure 6.1: Fault model

Individual node level fault usually results from: application software misbehavior, hardware failure and external impact of harsh environmental conditions (direct contact with water causing short circuit, node crash by a falling tree etc). In this work, we assume that software components are fault-free or maintained by the sensor application. Fault-tolerance of sensor data have been discussed by various existing research approaches [Ssu 2006]. In this work, we particularly focus on hardware resource depletion as the major cause of sudden death, and its effects at both node and network level. The network level faults are as a result of either the potential or permanent failure, and are usually related to the network connectivity, and sensor coverage rate. In our scheme, the network faults are assessed and analyzed by the management component i.e. group manager, cell manager. It holds the knowledge of its entire region in the network. Based on such information, the fault management system is capable of responding to various network failures with little human administration intervene. For example, when a group manager detects a cell with weak network health, it takes a proactive action by instructing nodes in that cell to send data less frequent for node energy consumption or alternatively, initiate the cell merging procedure.

6.3 A Self-managing Fault Management Mechanism for Wireless Sensor Networks

We used our hierarchical cellular architecture (described in chapter 3), and proposed a new self-managing fault management scheme [Asim 2008b, Asim 2009] for WSNs. The hierarchical model distributes the fault management tasks according to the node management responsibilities. The proposed fault management process can be divided into two phases

- Fault detection and diagnosis
- Fault recovery

6.3.1 Fault Detection and Diagnosis

Detection of faulty sensor nodes can be achieved by two mechanisms i.e. self-detection (or passive-detection) and active-detection as shown in Figure 6.2. In self-detection, sensor nodes are required to periodically monitor their residual energy, and identify the potential failure. In this approach, we consider the battery depletion as a main cause of node sudden death. As in Figure 6.2, we defined two functions of self-detection as: Check residual level and Trigger alarm. The first function checks the node residual battery energy. It then calls the 'Diagnose residual level' sub-function to compare the battery reading against the pre-defined threshold. If the residual battery dropped to a critical level the 'Trigger alarm' function is triggered to warn the managing nodes via message transmission. Thus, managing nodes respond to such fault after assessing the network performance. In our scheme, when a common node is failing due to energy depletion, it sends a message to its cell manager that it is going to sleep mode due to energy below the threshold value. This requires no recovery steps. Self-detection is considered as a local computational process of sensor nodes, and requires less in-network communication to conserve the node energy. In addition, it also reduces the response delay of the management system towards the potential failure of sensor nodes.

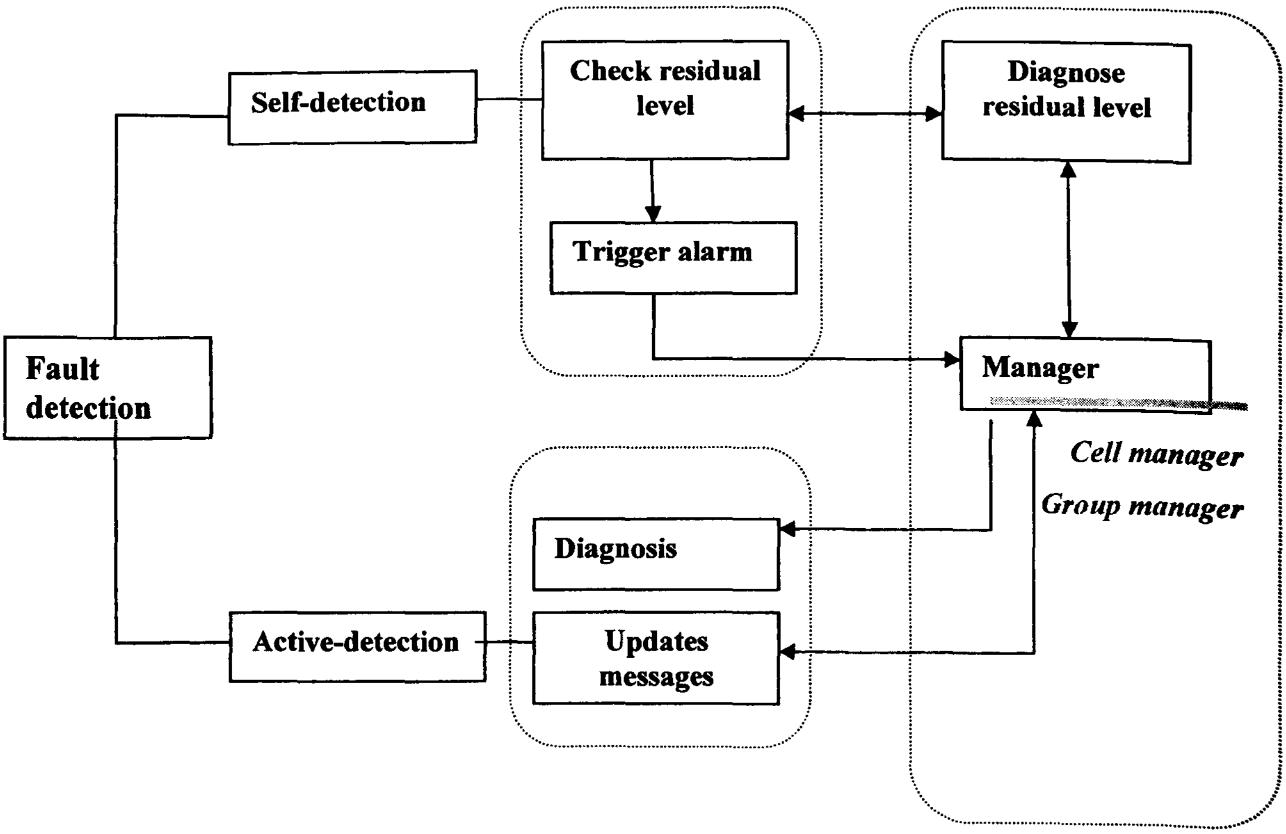

Figure 6.2: Fault detection and diagnosis process

To efficiently detect the node sudden death, our fault management system employed an active detection mode. In this approach, the message of updating the node residual battery is applied to track the existence of sensor nodes. In active detection, cell manager asks its cell members on regular basis to send their updates. Such as; the cell manager sends "get" messages to the associated common nodes on regular basis and in return nodes send their updates. This is called in-cell update cycle. The update_msg consists of node ID, energy and location information. As shown in Figure 6.2, exchange of update messages takes place between cell manager and its cell members through 'Update messages' sub-function. If the cell manager does not receive an update from any node, it then calls the 'Diagnosis' sub-function to send an instant message to the node acquiring about its status. If cell manager does not receive the acknowledgement in a given time, it then declares the node faulty and passes this information to the remaining nodes in the cell. This is performed during the active diagnosis. Cell managers only concentrate on its cell members and only inform the group manager for further assistance if the network performance of its small region has been in a critical level.

A cell manager also employs the self-detection approach and regularly monitors its residual energy status. As discussed in chapter 3, all sensor nodes start with the same residual energy. After going through various transmissions, the node energy decreases. If the node energy becomes less than or equal to 20% of battery life, the node is ranked as low energy node and becomes liable to put to sleep. If the node energy is greater or equal to 50% of the battery life, it is ranked as high and becomes the promising candidate for the cell manager. Thus, if a cell manager residual energy becomes less than or equal to 20% of battery life, it then triggers the alarm and notifies its cell members and the group manager of its low energy status and appoints a new cell manager to replace it.

Every cell manager sends health status information to its group manager. This is called out-cell update cycle and are less frequent than in-cell update cycle. If a group manager does not hear from a particular cell manager during out-cell update cycle, it then sends a quick reminder to the cell manager and enquires about its status. If the group manager does not hear from the same cell manager again during second update cycle, it then declares the cell manager faulty and informs its cell members. This approach is used to detect the sudden death of a cell manager.

Group manager also monitors its health status regularly and respond when its residual energy drops below the threshold value. It notifies its cell members and neighboring group managers of its low energy status and an indication to appoint a new group manager. Sudden death of a group manager can be detected by the base station. If the bases station does not receive any traffic from a particular group manager, it then consults the group manager and asks for its current status. If the base station does not receive any acknowledgement, it then considers the group manager faulty (sudden death) and propagates this information to its cell managers. The base station primarily focuses on the existence of the group managers from their sudden death. Meanwhile, the group managers and cell managers take most parts in passive and active detection in the network.

6.3.2 Fault Recovery

After nodes failure detection (as a result of self-detection or active detection), sleeping nodes can be awaked to cover the required cell density or mobile nodes can be moved to fill the coverage hole. A cell manager also appoints a secondary cell manager within its cell to acts as a backup cell manager. Cell manager and secondary cell manager are known to their cell members. If the cell manager energy drops below the threshold value (i.e. less than or equal to 20% of battery life), it then sends a message to its cell members including secondary cell manager. It also informs its group manager of its residual energy status and about the candidate secondary cell manager. This is an indication for secondary cell manager to standup as a new cell manager and the existing cell manager becomes common node and goes to a low computational mode. Common nodes will automatically start treating the secondary cell manager as their new cell manager and the new cell manager upon receiving updates from its cell members; choose a new secondary cell manager. The failure recovery mechanisms are performed locally by each cell. In Figure 6.3, let us assume that cell 1 cell manager is failing due to energy depletion and node 3 is chosen as secondary cell manager. Cell manager will send a message to node 1, 2, 3 and 4 and this will initiate the recovery mechanism by invoking node 3 to stand up as a new cell manager.

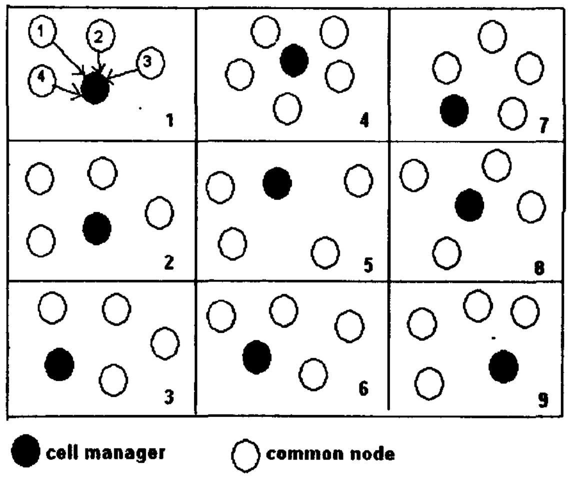

Figure 6.3: Virtual grid of nodes

In a scenario, where the residual battery energy of a particular cell manager is not sufficient enough to support its management role, and the secondary cell manager also does not have sufficient energy to replace its cell manager. Thus, common nodes exchange energy messages within the cell to appoint a new cell manager with residual energy greater or equal to 50% of battery life. In addition, if there is no candidate node within the cell that has sufficient energy to replace the cell manager. The event cell manager sends a request to its group manager to merge the remaining nodes with the neighboring cells. Cell merging process has been discussed in chapter 5.

When a group manager detects the sudden death of a cell manager, it then informs the cell members of that faulty cell manager (including the secondary cell manager). This is an indication for the secondary cell manager to start acting as a new cell manager. A group manager also maintains a backup node within the group to replace it when required. If the group manager residual energy drops below the threshold value (i.e. greater or equal to 50% of battery life), it may downgrade itself to a common node or enter into a sleep mode, and notify its backup node to replace it. The information of this change is propagated to neighboring group managers and cell managers within the group. As a result of group manager sudden death, the backup node will receive a message from the base station to start acting as the new group manager. If the backup node does not have enough energy to replace the group manager, cell managers within a group co-ordinate to appoint a new group manager for themselves based on residual energy.

Each cell maintains its health status in terms of energy. It can be High, Medium or Low. These health statuses are then sent out to their associate group managers periodically during out-cell update cycle. Upon receiving these health statuses, group manager predict and avoid future faults. For example; if a cell has health status high then group manager always recommends that cell for any operation or routing but if the health status is medium then group manager will occasionally recommend it for any operation. Health status Low means that the cell has insufficient energy and should be avoided for any operation. Therefore, a group manager can easily avoid using cells with low health status or alternatively, instruct the low health status cell to join the neighboring cell. Consider Figure 6.3, let cell 4 manager is a group manager and it receives health status updates from cell 1, 2 and 3. Cell 2 sends a health status low to its group manager, which alert group manager about the energy status of cell 2.

6.4 Message Broadcast Issue

The proposed fault management scheme relies on the message exchange among sensor nodes in the network. This might subsequently cause the communication flooding by broadcasting or re-broadcasting messages from different sensor nodes. To address this issue, we employed a message filtering mechanism to further reduce the redundancy of message exchange (as discussed in chapter 3). The message format contains fields as shown in table 6.1.

Group_id	The group id
Cell_id	The cell manager id
Timestamp	The message sending out time
Curr_energy	The current node battery enery

Table 6.1: Message attributes

The Group_id field is used to determine whether the received message belongs to the same group of current node. If not, the message will be dropped to avoid unnecessary message re-broadcast. Cell_id field helps a node to decide whether the message belong to its cell. If not, the message will be ignored and not forwarded. A sensor node might receive multiple copies of the same message forwarded by different intermediate nodes. To avoid redundant rebroadcast, we apply the value of 'timestamp' field in the second stage to determine whether the receiving message has been handled previously. If the receiving message is a new one, it will be processed and forwarded to the neighbouring nodes. On the contrary, that message will be dropped to lessen the network traffic and conserve the node energy.

6.5 Performance Evaluation

In this section we evaluate the performance of our proposed algorithm and analyze its cost by measuring node energy expenditure. In this experiment, we apply fault detection and recovery as main tasks of our fault management approach. Number of sensor is varied from 40 to 80, which are randomly deployed over 120 X 120 square meter area. Each sensor is assumed to have an initial energy of 2000 mJ. Every result shown is an average of 30 experiments. We first compared our work with that of Venkataraman algorithm [Venkataraman 2008], which is based on failure detection and recovery due to energy exhaustion.

1) Failure detection

In Venkataraman algorithm, neighboring information is already available to the cluster members through exchange of hello messages. The failure detection procedure starts after the cluster formation. When a node fails, the failing node parents and children take appropriate action to connect the cluster and bridge the gap formed by the failing node. The failing node itself reports its likeliness to fail so that appropriate measures can be taken to rectify the failures. The fail_report-msg is only passed to immediate hop members and then later on passed to the clusterhead.

In our proposed algorithm, if node energy drops below a threshold value, it then sends a failure report message directly to its one hop cell manager and goes to a low computational mode. In our proposed algorithm, there are two types of nodes: common node and a cell manager. Only one failure report message is sent out to the cell manager. Thus, avoid sending any extra message. This reduces the energy consumption and will not disrupt network operation.

2) Failure recovery

In Venkataraman algorithm, nodes in the cluster are classified into four types: boundary node, pre-boundary node, internal node and the clusterhead. Boundary nodes do not require any recovery but pre-boundary node, internal node and the clusterhead have to take appropriate actions to connect the cluster. Usually, if node energy becomes below a threshold value, it will send a fail_report_msg to its parent and children. This will initiate the failure recovery procedure so that failing node parent and children remain connected to the cluster. A join_request_mesg is sent by the healthy child of the failing node to its neighbors. All the neighbors within the transmission range respond with a join_reply_mesg/join_reject_mesg messages. The healthy child of the failing node then selects a suitable parent by checking whether the neighbor is not one among the children of the failing node and wether the neighbor is also not a failing node. In our proposed mechanism, common nodes does not require any recovery but goes to low computational mode after informing their cell managers.

In Venkataraman algorithm, clusterhead failure causes its children to exchange energy messages. The children who are failing are not considered for the new clusterhead election. The healthy child with the maximum residual energy is selected as the new clusterhead and sends a final_CH_mesg to its members. After the new clusterhead is selected, the other children of the failing clusterhead are attached to the new clusterhead and the new clusterhead becomes the parent for these children. This clusterhead failure recovery procedure consumes more energy as it exchange energy messages to elect the new clusterhead. Also, if the child of the failing clusterhead node is failing as well, then it also requires appropriate steps to get connected to the cluster. These can disrupt network operation and is time consuming.

In our proposed algorithm, we employ a back up secondary manager which will replace the cell manager in case of failure. Every time a cell manager is failing it sends a message to all its members including the backup secondary cell manager. Upon receiving this message from its cell manager, secondary manager automatically starts acting as a new cell manager and no further messages are required to send to other cell members to inform them about the new cell manager as they are already aware of secondary cell manager.

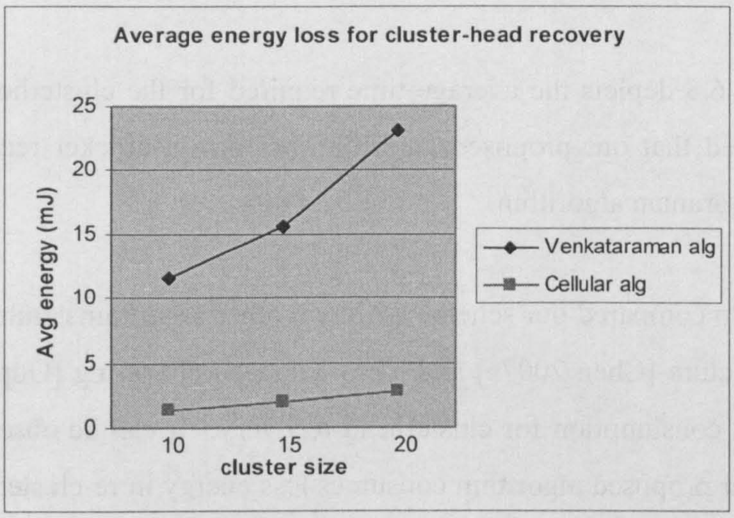

Figure 6.4: Average energy loss for clusterhead recovery

It can be observed from figure 6.4 that our proposed algorithm consumes less energy for clusterhead failure recovery when compared to Venkataraman algorithm. In Venkataraman algorithm, message exchange for the election of new cluster manager is both time and energy consuming. In our proposed algorithm, cell manager sends one message only to its member to recover from a failure.

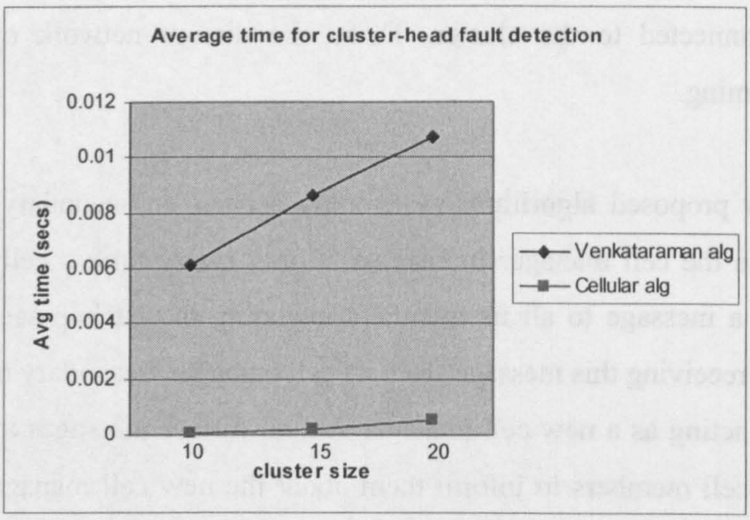

Figure 6.5: Average time for clusterhead recovery

Figure 6.5 depicts the average time required for the clusterhead recovery. It can be observed that our proposed algorithm perform a quicker recovery as compared to Venkataraman algorithm.

We also compared our scheme with two other algorithms: autonomic self-organizing architecture [Chen 2007a] and load- balanced clustering [Gupta 2003b], in terms of energy consumption for clusterhead recovery. It can be observed from figure (6.6) that our proposed algorithm consumes less energy in re-clustering when compared to the other two.

In autonomic self-organizing algorithm, when a high level node (header) failed to operate or need to step down due to low residual energy. All sensor nodes from the failed header need to join other available header nodes using the same mechanism. This again is not an energy efficient way to re-organize the cluster and also time consuming as compared to our cellular approach. In load-balanced clustering, when a gateway fails, the cluster dissolved and all its nodes are re-allocated to other healthy gateways. This consumes more time and energy as all cluster members are involved in the re-clustering process. In our proposed algorithm, only few nodes are involved in re-clustering.

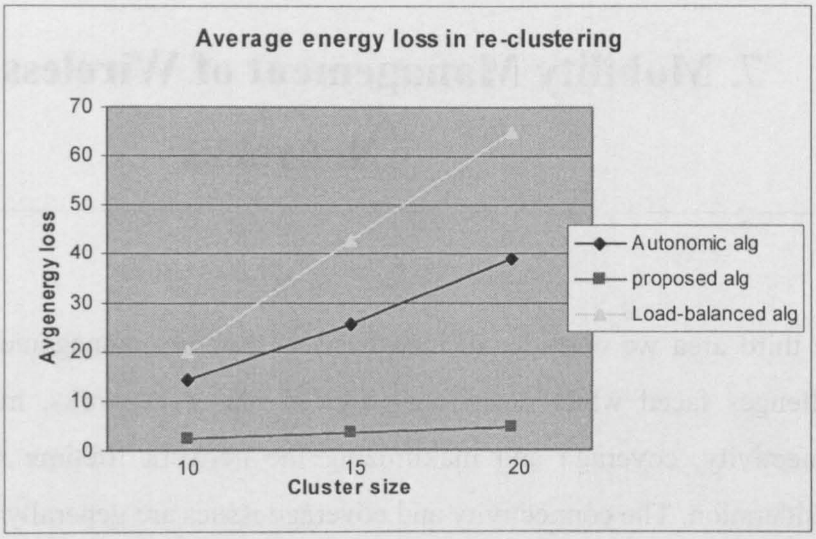

Figure 6.6: Average energy loss in re-clustering

6.6 Summary

In this chapter we proposed a localized cellular based scheme for fault detection and recovery in wireless sensor network. We divided the network into a virtual grid, where each cell consists of a group of nodes. This supports scalability of the network and increase network life time. Most of existing solution used some type of central entity to perform fault management tasks but in our proposed solution, the aim is to perform fault detection locally and in distributed fashion. The result obtained from the simulation clearly shows that our proposed scheme performs failure detection and recovery much faster than other existing schemes, and consumed significantly lower energy.

7. Mobility Management of Wireless Sensor Networks

The third area we consider in this thesis is mobility management. Among various challenges faced while designing wireless sensor networks, maintaining network connectivity, coverage and maximizing the network lifetime standout as critical consideration. The connectivity and coverage issues are generally met by deploying a sufficient number of sensor nodes, or using specialized nodes with long-range capabilities to maintain a connected graph. The network life time can be increased through energy conservation methods by using energy efficient protocols and algorithms.

Due to various factors, such as the inaccessibility of the terrain, scale of the network etc., optimal deterministic deployment of the sensor network is often infeasible. A common scenario envisioned for deployment is that of randomly scattering of sensor devices over the field of interest [Wang 2006b]. Thus, it makes the task of guaranteeing coverage much harder. As an alternative, mobile sensor nodes can be used to heal coverage holes in the network so that the randomness in sensor deployment can be compensated. Mobile platform are already available in many deployment scenarios, such as soldiers in battlefield surveillance application, animals in habitat monitoring applications, and buses in traffic monitoring applications. In other scenarios mobile devices can be incorporated into the design of the WSN architecture [Ekici 2006].

Failures in sensor networks are common and can be cured by using the redundant nodes in the network i.e. moving mobile redundant nodes to overcome the failure of sensor nodes or activate any sleeping redundant node in the group. Sensor nodes failure may cause connectivity loss and in some cases network partitioning. However, such situation can be corrected by injecting a few mobile nodes in the network which are then moved to desired locations and repair broken network or by using redundant mobile nodes in the network to heal the network failures.

Utilization of redundant mobile nodes plays an important role in prolonging network life time. However, reallocating mobile sensor nodes has many challenges and special requirements. First, movement in sensor networks involved communication and can be very expensive in terms of energy. Mobility in WSN would also require network reconfiguration. When a node moves in the network its relation to the environment and neighboring nodes will change and thus, cause the network to reconfigure. As a result mobility will add additional overhead to the network, in terms of communication messages and reconfiguration. Therefore, an energy efficient strategy is required to adopt mobile nodes in the network. Second, the reallocation of redundant mobile sensor nodes should have minimum effect on network sensing topology. Third, reallocation should be localized to achieve quick response time. For example; failure of sensor nodes monitoring a patient should be replaced immediately.

Our proposed distributed cellular architecture can be used for finding redundant mobile sensor nodes in a timely and energy efficient manner. Initially, all sensors are in the *active* state. If an area exceeds the required degree of coverage, redundant nodes will find themselves unnecessary and switch to the *sleep* state. In our framework, the whole network is divided into a virtual grid where each cell consists of a group of nodes. As discussed in [Wang 2005], the problem of finding redundant nodes has some similarities with the publish/subscribe problem, where the publisher advertise some information and the subscriber request the information. This terminology can be mapped to our problem where cells with redundant nodes are publisher and the cells that need more sensors are the subscribers. In the publish/subscribe system, the matching of a request to an advertisement is called match making. Generally, there are three types of matchmaking:

1) **Broadcast advertisement**

   Where publisher cells with redundant sensors flood the advertisement message and cells need sensor nodes can get information quickly. But this is not energy efficient approach as too many messages are involved due to network-wide broadcast.

2) **Broadcast request**

   In this approach, subscriber cells flood the request for more sensors and publisher replies after receiving the request. For the broadcast request approach, the delay is relatively long since it is on-demand. Also, flooding the network with request message is not an energy efficient approach.

3) This is similar to letting publishing cells advertise the information to some intermediate nodes and subscriber cells obtain this information when required [Carzaniga 2001, Eugster 2003, Ge 2003, Wang 2005]. Our solution is based on third type of matchmaking as it does not involving too much message exchange and can provide good response time.

During our research we identified that existing sensor relocation schemes consume too many messages in finding the nearest redundant mobile sensor nodes. We therefore propose a new sensor relocation scheme to locate redundant sensor nodes locally with minimum message overhead. We adopt a two level filtering mechanism to reduce the message exchange overhead. Information about the redundant sensor nodes is only available at some intermediate nodes. This approach helps in achieving a good response time and low message complexity.

7.1 Related Work

Mobility and its effects on the sensor network operation have been extensively studied and emerged as an important requirement for wireless sensor networks. In [Wang 2004] authors presented a proxy-based sensor relocation algorithm for the sensor networks composed of both static nodes and mobile ones. Static sensor nodes

construct a Voronoi diagram and bids closest mobile nodes to fill the sensing hole in their Voronoi polygons. Mobile nodes from nearby locations move to fill the coverage hole. This results in the emergence of new holes. Thus, more and more sensors are involved in relocation. This approach relies on flooding for replacement and uses a direct relocation method that can produce inconsistent relocation delay.
In [Wang 2005] authors presented a grid-quorum-based relocation protocol for mobile sensor networks. In this protocol, the network field is geographically partitioned into grids. In each grid, a node as grid head runs the quorum-based location service to find the redundant sensor nodes in the network. Then the discovered replacement is relocated along a carefully selected path in a cascaded (shifted) way.

Mobility of sensor nodes to fill in a coverage hole has been studied by some researchers [Li 2006, Li 2007, Wang 2005, Wong 2004]. The author in [Coskun 2008] discussed some related work and highlighted some important features:

- Distributed solutions, those run on all sensor nodes and do not take much help from clusterheads or sinks; results in early exhaustion of sensor nodes.
- Distributed algorithms will also possibly result in overlapping by relocating many nodes to the same hole.
- The relocation activity starts after the node failure; hence creating delay to heal the coverage hole.
- Too much message exchange involve in finding redundant nodes; thus causing too much energy consumption and over head.

7.2 Assumptions

We assume that a number of mobile sensor nodes are already deployed. Mobile sensor nodes have the same coverage range as static sensors. We assume that each mobile are provisioned with sufficient energy so that after relocation, they can sense and communicate for at least the same duration as static sensors. Due to energy and cost consideration, mobile sensor should move over a limited distance. The network is assumed to be heterogeneous as mobile nodes would need to contain additional power to drive the mobile actuation. As a result we propose the network will contain both static and mobile nodes to achieve the application and performance specification.

7.3 Proposed Mobility Management Framework

We used our hierarchical cellular architecture and management framework (proposed in chapter 3 & 4 respectively) for finding redundant sensor nodes in the network and reallocate them in timely, balanced and energy efficient manner. Our algorithm consists of two main phases: identifying redundant nodes and sensor relocation.

The cell manager is responsible for collecting information of its cell members, and determines the existence of redundant sensors based on their location. For redundant sensors located on the boundary of the cells, cell managers coordinate to make decisions. The cell manager can also monitor its cell members and initiate a relocation process in case of new event or sensor failure. Redundant nodes may be sent to a sleep mode to conserve energy.

1) Identification of redundant nodes

A wireless sensor may consist of hundreds to thousands of sensor nodes, and are usually deployed randomly through a vehicle, helicopter or any other mean. This may result in some area having more sensor nodes than others. The cell size can be other criteria to identify redundant nodes i.e. restricting the cell to have a total number of S nodes. *S is* a user-defined parameter, which can be adjusted to meet the required cell manager density. If a cell size is above the threshold value *S*, then some nodes can be sent to a sleep mode to adjust the cell size. The size of the cell (i.e., *S)* is made available to the sensors during network initialization.

The average number of static sensors needs to cover a cell is represented by *p* and is maintained by the cell manager. However, some cells may contain fewer sensors than *p* due to the randomness in deployment or node failures. If a cell i contains static nodes $(Ni) < p$, mobile nodes need to move into the cell to fill in the vacancies.

The cell managers within the same group represent a virtual grid structure towards their group manager. Instead of flooding subscribe/publish messages across the network and polling information from hundreds of thousands nodes, the cell manager contacts its group manager in the virtual grid structure to track the redundant mobile nodes. This design minimises the number of communication messages, and thus conserve node energy. Our proposed framework is based on finding redundant sensor nodes in a localized fashion. We believe that adopting localization to a certain degree reduces network traffic whenever possible. Addionally, such an approach also has a quick response to events that occurred in the network.

Each group manager maintains information about the publisher cells within its group and shares this information with closest neighboring group managers only. This supports the short distance movement of mobile sensor nodes. If the mobile sensor node travels a long distance to replace a faulty node or fill the coverage, it may run out of power and create a new coverage hole. When a cell has redundant sensor nodes, the cell manager propagates this information to its group manager. When any cell wants more sensors, the cell manager only needs to contact its group manager. Group manager will first look for redundant sensor nodes with in a group, and if there are no redundant nodes within its group, it then searches which nearest group has redundant sensor nodes.

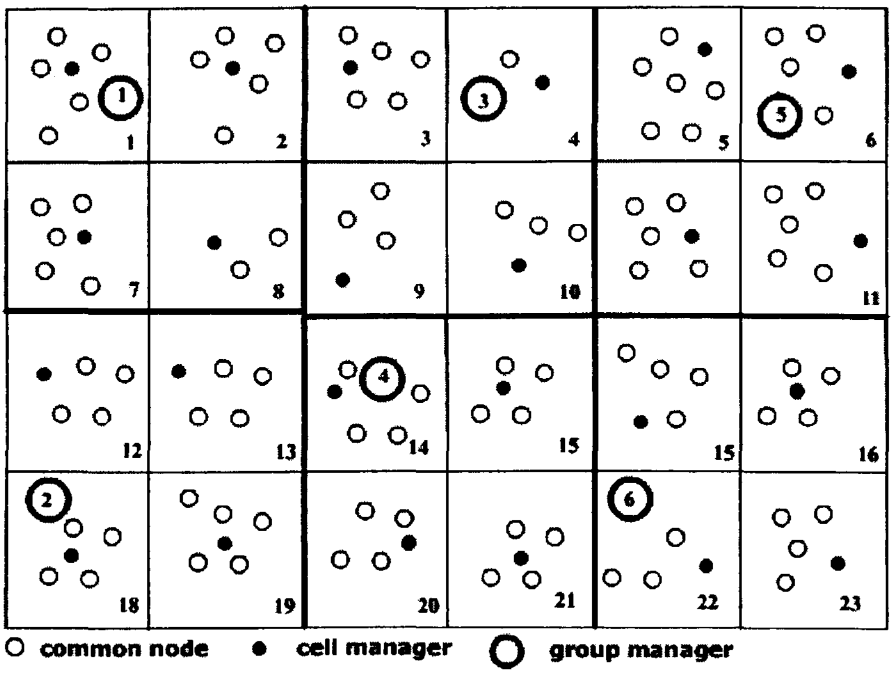

Figure 7.1: Finding redundant nodes

For example, as shown in figure 7.1, suppose cells (1), (2) and (7) have redundant sensors, while cell (8) needs more sensors. The cell managers of cells (1), (2), and (7) propagate its redundant sensor information to their group manager in the form of publish_messages. The cell manager of cell (8) puts forward its demand for more sensors to its group manager. This is a subscribe_message. Selection of the cell (providing redundant nodes) will be based on number of redundant nodes it contains and its distance to the subscriber cell. The distance between the subscriber cell and all possible publishing cells will be determined by the group manager. The publishing cell with the shortest distance to the subscribing cell will get the priority. The group manager will notify the selected publisher cell to move its redundant sensor nodes to the subscriber cell. This will be advised through the move_message.

The group manager will also send an ack_message to the subscriber cell, if it is able to find some redundant nodes. Registration phase will be invoked once the new nodes join the subscriber cell.

As shown in figure 7.1, group manager 1 can share redundant sensor information only with group manager (2), (3), and (4). This localized restriction reduces in-network communication, and conserves valuable energy and network bandwidth. Consider a scenario, when redundant nodes cannot be offered within the same group. Still using the example of figure 7.1, suppose cell (8) need more sensors and cells (1), (2), and (7) cannot offer sensor nodes. Group manager 1 then checks the publishing information it received from other group managers (2), (3), and (4) and propagates the subscriber request to the nearest group manager. Group managers can share publishing information either on demand or through regular messages, depending on the type of application. Again, publishing cells are selected based on available redundant nodes and shortest distance to the subscribing cell.

2) Sensors Relocation

After locating the redundant sensor nodes, the next step is how to move the sensor to the new destination. Moving sensor nodes directly to the destination is a possible solution but, may take longer time than the application requirement. Moreover, moving a sensor node for a long distance consumes too much energy. If the sensor node dies shortly after it reaches the destination, this movement is wasted and another sensor has to be found and relocated [Wang 2005]. Cascaded movement can be used to address this problem. It finds some cascading (intermediate) nodes which help the redundant node in relocation to reduce delay and balance the power. Consider figure 7.1, suppose cell (4) needs more sensors and cell (14) has been selected to provide some redundant sensors. Instead of letting the sensors move directly from cell (14) to cell (4), some sensor node are chosen in cell (9) as cascading nodes. As a result, cascade nodes will move to cell (4) and redundant nodes from cell (14) will move to cell (9) to fill the cascade movement.

Cascaded movement is not feasible for a very long distance movement as it will consume more energy than direct movement. Therefore, a balance is required to minimize the energy consumption and to achieve a good response time. We suggested two types of movements in our framework:

a) **Direct movement**

When a movement is between two direct neighboring cells, this should be addressed by direct movement. Consider figure 7.1, nodes from cell (2) can directly move to cell (8) to heal the coverage. This would help in consuming less energy and provide quick response time.

b) **Cascaded movement**

Consider figure 7.1, nodes from cell (1) and cell (3) are connected though cell (2), then sensor movement between cell (1) and (3) will take place though cascaded movement. Intermediate node from cell (2) will move to cell 1 and redundant nodes from cell 3 will move to the cascading cell (2).

If no redundant nodes are available within the group and with neighboring groups then the subscriber cell can merge with one if its neighboring cell to carry on its network operation. This stage does not involve any movement as we believe moving sensor node from a long distance cell is not feasible in terms of energy and response time.

The parameters for the above two movement are based on the type of application and cell size. It can be adjusted according to network deployment and availability of redundant nodes.

The newly moved redundant sensor nodes in the subscribe cell will then be registered using the reconfiguration algorithm (discussed in chapter 5). Therefore, we can say that mobility management works closely with configuration management services to efficiently reconfigure the network.

7.4 Data Definitions

A number of control messages are defined by our algorithm. They play an essential role in coordinating nodes and helping accomplish protocol goal. There are four main types of control messages i.e. subscribe message (subscribe_message), publish message (publish_message), acknowledge message (ack_message), and join message (join_msg).

1) Subscribe message

When a cell needs more sensors it then send a subscribe message to its group manager. Subscribe message consist of sender node id, cell id, group id, and number of sensor nodes it needs. These messages are sent by the cell managers with in a group.

2) Publish message

A cell manager sends a publish message to its group manager to register itself as a publishing cell with redundant sensor nodes. Publish message consists of sender node id, cell id, group id and number of redundant nodes it can offer.

3) Acknowledge message

When a group manager can find redundant sensor nodes for a subscriber cell, it then inform the subscribing cell by sending an acknowledge message.

4) Join message

This message is used to register the newly injected mobile nodes with in the network. The join request message consists of the node id, location of the node, group id and the cell id it wishes to join. The cell id in join_msg avoid flooding of the message i.e. the node will discard the message if it does not belong to its cell.

7.5 Performance Optimization

To further reduce the message exchange, we used the filtering scheme for propagating request messages (discussed in chapter 3).

Addr	GROUP	MS_TYPE	....	DATA	....

Figure 7.2: Message format

As stated earlier, the MS_TYPE field in the data packet distinguishes messages from different sensor nodes in the management hierarchy. It contains four types of values, referring separately to subscribe message, publish message, acknowledge message and move message. We particularly focus on the discussion of subscribe and publish messages. The 'GROUP' field, containing the value of group id, and is applied to distinguish and avoid receiving messages from other groups. The 'DATA' field is actually the structured message packet (i.e. both subscribe message and publish message) and contain fields like group_id, cell_id, node_id, node_location and no of nodes as shown in figure 7.3.

```
Class Subcribe Message
{ public:
    IPAddr_t src;            // source address
    NodeId_t dest_cellid;        // the destination cell id;
    NodeId_t groupid;        // the group id
    NodeId_t mnid;           // the managing node id
    Count_t  hop_cn;         // record the communication hop
    Time_t   timestamp;      // the message sending out time
}
Class Publish Message
{ public:
    IPAddr_t src;            // source address
    IPAddr_t des;            // destination address
    NodeId_t groupid;        // the group id
    NodeId_t mnid;           // the managing node id
    Count_t  hop_cn;         // record the communication hop
    Time_t   timestamp;      // the message sending out time
    Energy_t curEnergy;      // the current node battery energy
}
```

Figure 7.3: Data field attributes

The three stages of message filtering lessen the redundant broadcast in the network for energy conservation i.e. the message type, and timestamp.

The message type stage first adopts the 'GROUP' field to quickly determine whether the received message belongs to the same group of current node. If not, the message will be dropped to avoid unnecessary message re-broadcasting. It then checks the 'MS_TYPE', distinguishing data packet between subscribe and publish messages.

After retrieving the value of 'cell_id', the node decides whether it belongs to the event cells (e.g. destination cell) to process the message.

A sensor node might receive multiple copies of the same message forwarded by different intermediate nodes. To avoid redundant rebroadcast, we apply the value of 'timestamp' field in the second stage to determine whether the receiving message has been handled previously. If the receiving message is a new one, it will be processed and forwarded to the neighbouring nodes. On the contrary, that message will be dropped to lessen the network traffic and conserve the node energy. In [Wong 2004] a self-organizing technique for enhancing the coverage of wireless sensor networks has been proposed. One of the weak points is the possibility that more than one sensor node may move towards the same location. This issue has been addressed in our algorithm by introducing another layer of managing nodes, comprise of group managers. Group manager will select the appropriate publishing cell for the subscriber and will send a message to both subscriber and publishing cell managers to progress the move. This will perform a controlled move and avoid any additional movement.

7.6 Experimental Validation

In this section we evaluate the performance of our proposed algorithm, and a total of 100 sensor nodes were deployed over 100 x 100 square meter area. Each sensor is assumed to have an initial energy of 2000 mJ. The experiment assumed that channel allowed collision and that packets could be dropped in the medium. To evaluate the performance of our proposed algorithm, we compare the results with the grid-quorum-based relocation protocol [Wang 2005]. We have selected the grid-quorum based protocol for comparison because it is also based on grid based architecture to relocate sensor nodes. Two other sensor relocation algorithms [Wang 2003b] and [Wang 2006a] were proposed by the same author of grid-quorum algorithm. In [Wang 2005], the author compared grid-quorum with one of their own VOR scheme [Wang 2006a, Wang 2006b] and proved that grid-quorum is more efficient in terms of relocation time and energy consumption. Our simulation results shows significant improvement over the grid-quorum based protocol in terms of energy efficiency and response time. In grid-quorum-based protocol, the network is geographically partitioned into grids, and each grid is represented by its grid head. A grid row is called demand quorum, while a grid column is called supply quorum. Each grid head publishes the information about the redundant sensor nodes to all the grid heads in the supply quorum. When a grid needs more sensors, it broadcasts a request within its demand quorum to discover the closest redundant node. Because every demand quorum intersects with all the supply quorums, a redundant node can always be found if any exists.

Experiments were performed to elucidate the characteristics of the proposed mechanism in two stages:

1) Publishing phase

In grid-quorum based approach, a grid with redundant sensors advertises itself through supply quorum. This involves advertising publication information through a number of grids or cells, and involving a great number of messages to advertise the publication information. Grid-quorum-based protocol shows significant improvement over the "broadcast advertisement" [Eugster 2003] approach in terms of response time and message complexity. We further reduce the message complexity by introducing a hierarchical architecture, where cells combines to form various groups. Instead of sending the publication information to any column of cells, we encourage cells to send the information only to their group managers. This significantly reduces message complexity for not involving too many cells. Also, hierarchical architecture supports the filtration of duplicated messages, which further reduces the message overhead. As shown in figure (7.4), our proposed algorithm achieves better energy consumption as compared to grid-quorum-based approach. The reason is that our proposed algorithm reduces the message exchange (involving in finding the nearest redundant sensor nodes) by sending the publication messages to fewer nodes and utilized the filtering scheme. Also, the proposed algorithm outperforms the grid-quorum-based algorithm in the relocation time as shown in figure (7.5).

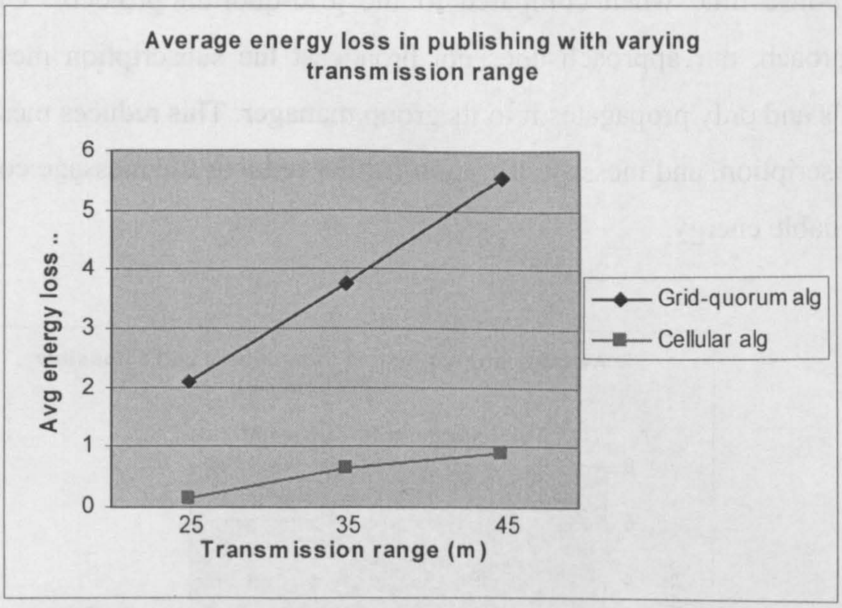

Figure 7.4: Average energy loss in publishing

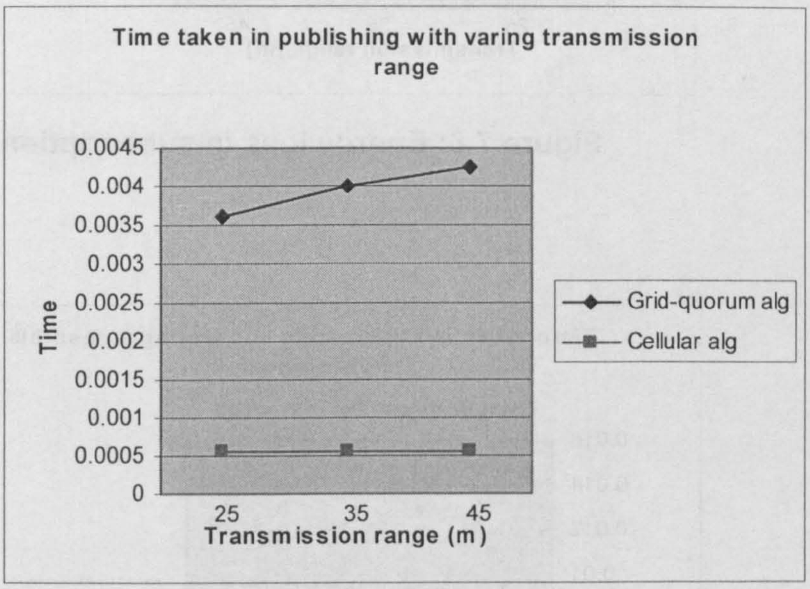

Figure 7.5: Time taken in publishing

2) Subscription phase

When a grid head or cell manager detects a sensing hole, it broadcast a subscription request for redundant sensor nodes. The subscriber cell is then notified of the nearest cell with redundant sensors. It can be observed from figure (7.6) and figure (7.7) that our proposed algorithm consumes less energy in subscription and achieve a good response time when compared to the grid-quorum protocol. Unlike grid-quorum approach, our approach does not broadcast the subscription message to too many cells and only propagates it to its group manager. This reduces message exchanged in subscription, and message filtration further reduces the message complexity and save valuable energy.

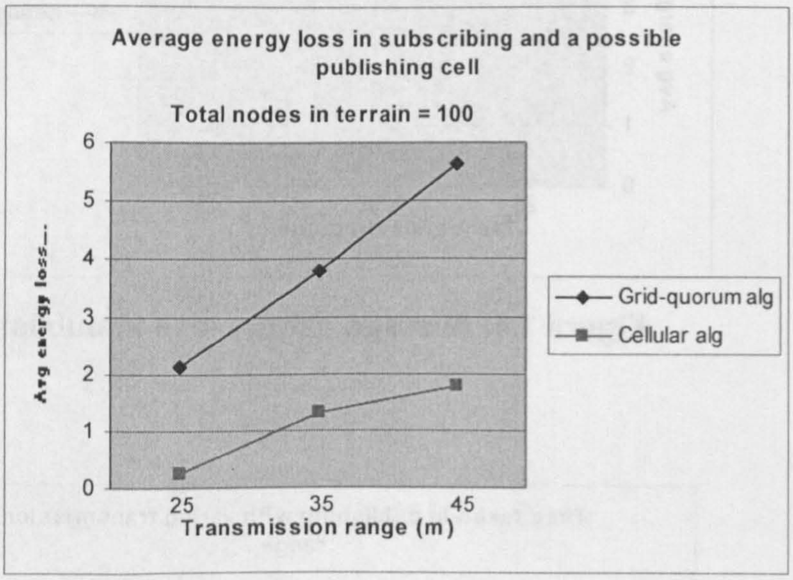

Figure 7.6: Energy loss in subscription

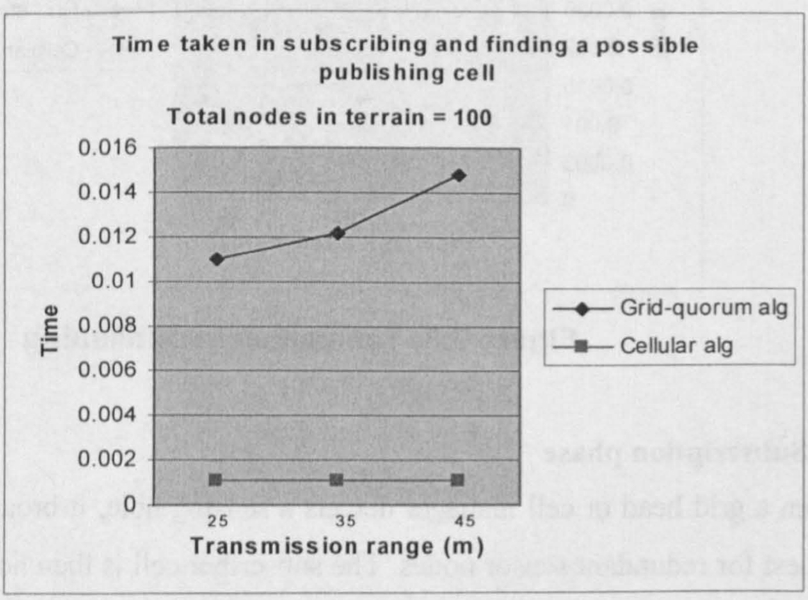

Figure 7.7: Time taken in subscription

7.7 Summary

In the chapter, we discussed the problem of sensor relocation that can be used to deal with sensor coverage holes or sensor failures. We proposed a two phase sensor relocation algorithm: redundant sensor nodes are first identified and then relocated to the nearest target location. We used our cellular hierarchical architecture to locate redundant mobile sensor nodes with minimum message complexity, and proposed to use both direct and cascaded movement to relocate sensor nodes quickly. Information about the redundant sensor nodes is only available at some intermediate nodes. This helps to reduce message complexity through message filtration and avoid message flooding. Simulation results verify that the proposed solution outperforms others in terms of relocation time and total energy consumption.

8. Conclusion and Future Work

This thesis has presented a framework for the self-organization and self-management of resource constraints wireless sensor networks. A number of novel mechanisms have been developed for the new framework. The aim of the framework is to be flexible enough to accommodate different types of WSN applications and extend the network life by efficiently utilizing nodes energy and support the scalability of the system in a densely deployed sensor networks.

The chapter is organized as follows. We present a summary of the thesis in section 8.1. Our main contributions and a summary of the cellular framework are presented in section 8.2. The comparison of our proposed work with existing approaches is discussed in section 8.3. Future work is investigated and proposed in section 8.4.

8.1 Thesis Summary

A wireless sensor network is a network that consists of a base station and large number of sensor nodes distributed or positioned in the environment of interest. Each sensor node is expected to detect events of interests and estimate parameters that characterize these events. The resulting information at a node needs to be transferred to the base station either directly or in "multi-hop" fashion involving automatic routing through several nodes in the network. Sensors networks provide an easy solution to those applications that are based in the inhospitable and low maintenance areas where conventional approaches prove to be impossible and very costly. Examples include environmental monitoring- which involves monitoring air soil and water, condition based maintenance, habitat monitoring, seismic detection, military surveillance, inventory tracking, smart spaces etc. The design, implementation, deployment and maintenance of such large scale wireless sensor networks differ from and is more challenging than traditional systems due to factors such as dynamic topology, energy and memory constraints, infrastructure less architecture, and the harsh environment in which wireless sensor networks are deployed. Thus, network management becomes extremely important in order to keep the whole network and application work properly and continuously. WSNs are different from traditional networks and present a new set of properties. Typically the structure of a traditional network will remain the same in all applications while a WSN's structure will change according to its application. Therefore, traditional management schemes are impractical for wireless sensor networks. The task of developing and deploying management system in environment that contain hundreds to thousands of energy constrained sensor nodes is not trivial. This task becomes more complicated due to the physical restriction of the unattended sensor nodes. Despite the importance of wireless sensor network management, there is no generalized solution available for WSN management.

Our work in this thesis focuses on designing a self-organizing hierarchical cellular architecture for WSN and then maps the cellular architecture into a self-management framework for wireless sensor networks to monitor the network with minimum overhead, collect the management data energy efficiently, and adapt and reconfigure autonomously to cope with changes of node conditions, resources and network environment. In this thesis we have presented our work on developing and evaluating a self-organization and self-management framework for WSNs. In order to achieve this we included the following materials:

Chapter 1 discussed the wider context and outlines the problem of self-organization and self-management in large scale WSNs. It includes the definition of WSNs, their main applications and current WSN projects. It describes the communication architecture and components of sensor nodes. We also briefly describe the importance of clustering and management in WSNs. Chapter 1 also highlights the parameters necessary for developing a self-organizing and self-managing WSN.

Chapter 2 presented a survey of existing clustering schemes and highlighted some problems: (1) existing clustering schemes have high cost of clustering and re-clustering (2) existing clustering schemes are not flexible enough to accommodate the different application of WSNs (3) not flexible enough to deal with the highly dynamic nature of WSNs (4) do not offer optimal distribution of clusterheads (5) do not support mobility of sensor nodes. All these deficiencies of existing clustering schemes put emphasis on developing a flexible clustering scheme for WSNs to address these problems. In chapter 2, we also surveyed existing management schemes for WSNs and highlighted their drawbacks: (1) No generalized management solution available to deal with different types of WSN applications (2) existing solutions do not consider the mobility of sensor nodes in the network (3) they are based on heterogeneous sensor nodes, which usually requires a careful placement of sensor nodes to contribute to the performance of the sensing application. This is not feasible for the random deployment of sensor networks in a harsh environment (4) they incur high message complexity and do not scale with the growth of the network (5) they perform centralized diagnosis and puts extra overhead on managing nodes (6) failure detection and recovery techniques are not energy efficient.

Chapter 3 provides an overview of the self-organizing cellular architecture. First it describes the background of the architecture design to highlight necessary requirements. We have also identified issues and challenges that are important when designing an effective self-organizing scheme for wireless sensor networks. The new cellular scheme is evaluated and compared to other existing schemes using simulation techniques

Chapter 4 provides an overview of the self-management framework (based upon the self-organizing cellular scheme) to efficiently support the management of wireless sensor networks. This chapter describes the management hierarchy, management roles, and the management process. Chapter 4 also describes different management units for the management framework.

Chapter 5 presents a re-configuration algorithm to re-organize the network topology due to network dynamics. Different techniques have been discussed to deal with network dynamics i.e. node joining a cell, node status change, cell manager status change and cell merging. The re-configuration algorithm is evaluated and compared to other existing schemes using simulation techniques

Chapter 6 presents a fault management scheme to identify failing sensor nodes and recover the connectivity in wireless sensor networks. We begin by describing our pre-design investigation. This section provides a detailed analysis for identifying some core information needed to help us towards the development of an efficient and improved fault management scheme. It also presents an evaluation of the proposed fault management scheme using simulation techniques.

Chapter 7 deals with the mobility of sensor nodes and discuses a sensor relocation scheme to locate redundant mobile sensor nodes and move them to heal coverage holes in the network. First we identified issues and challenges that are important when designing an effective and energy efficient sensor relocation scheme for wireless sensor networks. We evaluate the performance of the sensor relocation scheme through simulation and compared it to existing work. Finally, suggestions for future work and conclusions are presented in this chapter.

8.2 Research Contributions

This thesis presents a novel self-organizing and self-management cellular framework [Asim 2008a, Asim 2008b, Asim 2009, Asim 2010a] for WSN that enables the sensor nodes to efficiently coordinate amongst themselves to achieve a large sensing task. The framework monitors the sensor network with minimum overhead, collect the management data energy efficiently, and can adapt and reconfigure autonomously to cope with changes of node conditions, resources and network environment. The framework is based upon the following mechanisms developed as parts of our contributions:

- We first developed a novel hierarchical cellular architecture [Asim 2008a] to efficiently utilize sensor nodes energy and extend the network life time. We propose an n-tier hierarchical framework for wireless sensor networks. However, the number of hierarchical levels is based on application type and no of nodes. Existing hierarchical scheme for WSNs are based on fixed parameters and therefore can be used for specific applications. However, our generic cellular hierarchical framework allows us to define a number of parameters i.e. number of hierarchical levels, cluster size. These parameters can be defined based on application requirements. For example, hierarchical levels has a significant impact on data delay and data aggregation (i.e. increasing the number of hierarchical levels may result in increase in data delay). The cellular architecture supports the optimal distribution of managing nodes across the network and provides the maximum coverage of the sensor nodes. Unlike existing clustering schemes, the formation of the cellular architecture consumes less energy as it is based upon the actual or virtual coordinates of the node. The cellular architecture used a layered data aggregation process to avoid clusterhead overhead and offer more energy saving. Load balancing amongst managing nodes in our architecture is guaranteed by the constraints on the maximum number of nodes in a cluster. This design encourages the sensor nodes to be more self-organized and extend the network life time for as long as possible.

- As discussed earlier, the maintenance and control of WSNs is essential to ensure efficient use of network resources for appropriate information gathering and processing. Despite the importance of wireless sensor network management, there is no generalized solution available for wireless sensor network management. To address this challenge, we mapped our hierarchical cellular architecture into a self-management framework to support the network management system design for wireless sensor networks. This self-managing framework can be used as a generic management solution for wireless sensor networks, which consist of different functional units for different management services i.e. fault management, mobility management and configuration management. These functional units are integrated with each other to provide an energy efficient network management system for dealing with the resource constrained wireless sensor networks. The cellular framework enables sensor nodes to perform management tasks individually or in combined fashion, and reduces in-network communication and traffic for conserving the network energy. Instead of heavily relying on few central management entities (e.g. clusterhead nodes) or small portion of nodes, the management framework consists of homogenous sensor nodes that encourage sensor nodes to evenly and efficiently share the management burdens for battery-energy conservation. The self-managing framework describes different management roles, management policies and different management tasks for managing nodes.

- We have proposed a re-configuration algorithm [Asim 2010a] as a part of our configuration management unit, to support sensor networks to energy-efficiently re-organize the network topology due to network dynamics such as node dying, node power on/off, new nodes joining the network and cell merging. Most of existing maintenance schemes used some kind of flooding to reconfigure the network. However, our re-configuration algorithm does not use flooding to maintain and reconfigure the network but employs a localized criterion for cell re-configuration and maintenance in a distributed fashion. The re-configuration algorithm self-organizes the network efficiently to accommodate new sensor nodes drifting into network or mobile nodes move in to fill the coverage holes. Node status change is a common activity in our proposed management framework i.e. when a node changes its status to sleep mode to conserve energy. The re-configuration algorithm responds well to node status changes and is managed locally. Cell merging as an important phase of our re-configuration algorithm and plays a vital role to maintain connectivity in the network.

- To detect faulty nodes and recover the connectivity in wireless sensor networks, we have developed a new fault management scheme [Asim 2008b, Asim 2009] (based upon the cellular architecture). We aimed to maintain the cell structure in the event of failures caused by energy-drained nodes. The energy drained nodes are detected and recovered in their respective cells without affecting overall structure of the network. The faulty sensor nodes are detected and recovered in their respective cells without causing any disruption to the ongoing network operation. The grid based fault management scheme permits the implementation of fault detection and recovery in a distributed manner and allows the failure report to be forwarded across cells. The fault management scheme performs fault detection and recovery quickly and energy efficiently.

- Redundant mobile sensor nodes can be moved to repair coverage holes caused by node failures or random deployment of sensor nodes. We have developed a new two-phase sensor relocation solution: redundant sensors are first identified and then relocated to the target location. During our research we identified that existing sensor relocation schemes consume too many messages in finding the nearest redundant mobile sensor nodes. However, our scheme quickly locates the closest redundant sensors with low message complexity. Our scheme adopts a two level filtering mechanism to reduce the message exchange overhead. Information about the redundant sensor nodes is only available at some intermediate nodes. This helps in relocating sensor nodes in timely, balanced and energy efficiently manner.

8.3 Comparison with Existing Approaches

As discussed in chapter 3, the main objective of our proposed self-organizing hierarchical architecture is to extend wireless sensor network life time by efficiently utilizing sensor nodes energy and supports the scalability of the system. The most energy consuming activity of sensor networks is radio communication between sensor nodes. To save energy consumption, generally two schemes are used: data aggregation and switching of redundant sensor nodes into sleep mode. Grouping or clustering of sensor nodes is an energy efficient approach for data aggregation and to control the mode of sensor nodes. The most notable clustering schemes for wireless sensor networks are [Chatterjee 2002, Chen 2007a, Gupta 2003b, Heinzelman 2000, Heinzelman 2002, Subramanian 2000, Venkataraman 2005, Younis 2004]. However, most of existing clustering schemes consume too much energy in group formation and re-formation. Heterogeneous clustering scheme requires clusterheads to be carefully placed in the network to contribute towards the performance of the application. This is not suitable for applications that need random deployment of sensor node in a harsh environment, where human intervention is not possible. Also, most of existing clustering schemes do not support nodes mobility. In contrast, our cell and group formation algorithm consumes less energy as it is based upon the actual or virtual coordinates of the nodes. Our proposed scheme is based on homogenous sensor nodes to support the balanced distribution of managing nodes and to extend the network life time. The hierarchical design of our proposed solution minimizes the communication messages, eliminates the redundancy of transmitted data, and thus conserves energy. It can easily keep track of mobile sensor nodes i.e. node joining/leaving a cell. We simulated our proposed algorithm and compared it to existing work. The proposed scheme shows better result with regards to the life time of the network.

In addition, we have mapped the cellular architecture into a management framework to support network management system design for wireless sensor networks. Existing management solutions for WSN can be categories into centralized [Lee 2006b, Ramanathan 2005a, Song 2005, Tolle 2005] distributed [Boulis 2003, Perillo 2003, Ramanathan 2005b, Ruiz 2003] or hierarchical [Deb 2001, Deb 2004, Ying 2005]. Centralized management solutions incurs high message overhead in terms of bandwidth and energy. Also, they are not scalable with the growth of network. Distributed management has lower communication costs than centralized, but it is complex and difficult to manage. Hierarchical management solutions though more efficient for WSNs, but consume much energy to form the management hierarchy. Our proposed management framework is based on hierarchical levels. The appliance of hierarchical structure is to specify different management roles and efficiently distribute management tasks across the network. Instead of heavily relying on any central management entity or small portion of nodes, our management framework encourage sensor nodes to evenly and efficiently share the management burdens for battery-energy conservation. The proposed management framework has been discussed on three core functional area i.e. configuration management, fault management and mobility management. Our proposed re-configuration algorithm has shown better performance against existing schemes. The fault management scheme has been compared with existing schemes in terms of failure detection and recovery. The result obtained from the experiments clearly shows that our proposed scheme performs failure and detection and recovery much faster than other existing work, and consumed significantly lower energy. The mobility management unit has been discussed as a sensor relocation scheme. Our comparison through simulation shows that our sensor relocation scheme outperforms existing solutions in terms of relocation time and total energy.

8.4 Future Work

For future research, we plan to extend this work in several directions. In this thesis three functional areas have been considered while the remaining functional areas will be considered as future work. The first is to develop a quality-of-service management protocol suitable for the new cellular self-managed framework. The second is to extend our proposed management framework to provide security related services. The third is to consider communication related faults in our proposed fault management scheme. The fourth is to develop a routing protocol for our new cellular hierarchical architecture.

8.4.1 A New Quality-of-Service Management Protocol

It is envisioned that WSNs will become pervasive in our daily lives, for example, in our homes, offices, and cars. Just as internet transformed how we interact with one another, WSNs promise to revolutionize the way we understand and manage the physical world. Ultimately, WSNs will be connected to the internet to achieve global information sharing. This technical trend is driving WSNs to provide quality-of-service support to satisfy the service requirements of various applications. Depending on the type of application, Qos in WSNs can be characterized by reliability, timeliness, availability, and security, among others [Chen 2004, Xia 2008]. Despite intensive research in wireless sensor networks, limited work has been found on QoS management. Using our self-management framework, QoS management services can be used in conjunction with configuration management to manage the energy consumption of sensor nodes in the network. Also, there is a tradeoff between the life time of the network and the quality of service i.e. investing more energy can increase quality but may drastically reduce network lifetime. Therefore, a balance between network energy consumption and quality of service is necessary. Consequently, a new QoS management protocol is needed to provide QoS support to satisfy the service requirements of various WSNs applications.

8.4.2 A New Security Management Protocol

Security is important for various wireless sensor network applications such as intrusion detection or actuation and control, where an adversary could influence the network to avoid detection or perform incorrect actions to destabilize the system. In chapter 4, we proposed our management framework with various functional units. Security management is the part of our management framework model but has not been discussed in this thesis. In the future work we will extend our proposed management framework to provide security management services in order to protect networks sensitive data and sensors reading.

8.4.3 Communication Related Faults

In chapter 6, we proposed our fault management scheme that particularly focus on faults related to hardware resource depletion. However, there are other types of faults in WSNs i.e. communication related faults. Communication in WSNs is more prone to failures than in traditional networks because sensor nodes normally operate with high density in harsh environment. The communication failure in sensor network can be defined as route or link failure during data transmission. The link failure means the communication channels between the sender and receiver nodes inside the same radio range cannot be set up. The routing path failure means the communication link between the source and destination nodes in the same networks cannot be established. Therefore, how to support link stability and maintain route lifetime become key issues in communication fault tolerance.

8.4.4 A New Routing Protocol

The study [Pottie 2000] shows that energy consumption is dominated by communication for wireless sensor networks. Wireless sensor networks are thus self-organized networks where the node discovers each other and act as routers, maintaining information about their neighbors and themselves. Each node in the network may be the final destination for a packet, or may act as a forwarding node to their destination. Our proposed hierarchical cellular system architecture includes several components: common sensing nodes, monitoring nodes (cell managers and group managers), routing nodes, and the base station. In this way, our proposed architecture can be divided into three layers: a transmitting layer, based on routing nodes that transmit data sensed by common sensing nodes; sensing layer, which is used to sensed the required information from the environment and a control layer, which is used to monitor the network condition and perform functions like data aggregation. Thus, an energy efficient routing protocol is required to support in-network data processing which can reduce data packets greatly and only transmit processed and necessary data instead of all raw data to the base station or to any other managing node (cell manager/group manager). Also, our proposed cellular architecture is based on location information. Geographic routing protocols take advantage of the location information of sensor nodes to provide higher efficiency and scalability. Thus, the future work will explore this scenario.

Research in WSNs, especially in the self-organization and self-management of WSN, is still immature. There are still many research challenges to be addressed in order to implement WSNs realistically in our daily life. We believe that our novel self-organizing and management framework and investigatory research findings will help toward the future development of WSNs.

References

[Abbasi 2006] A. A. Abbasi, M. I. Buhari, and M. A. Badhusha, "Clustering Heuristics in Wireless Networks: A Survey," in *Proceedings 20th European Conference on Modelling and Simulation*, 2006.

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

[Agre 2000] J. Agre and L. Clare, "An Integrated Architecture for Cooperative Sensing Networks," in *IEEE Computer*, 2000, pp. 106-108.

[Akyildiz 2002] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," in *IEEE Communication Magazine*, 2002, pp. 102-114.

[Albowitz 2001] J. Albowitz, A. Chen, and L. Zhang, "Recursive Position Estimation in Sensor Networks," in *Proceedings of ICNP'01*, 2001, pp. 35-41.

[Amis 1999] A. Amis, R. Prakash, T. Vuong, and D. T. Huynh, "Max-Min D-Cluster Formation in Wireless Ad Hoc Networks," in *Proceedings of IEEE INFOCOM*. vol. 1, 1999, pp. 32-41.

[Anastasi 2004] G. Anastasi, M. Conti, A. Falchi, E. Gregori, and A. Passarella, "Performance measurements of motes sensor networks," in *Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, 2004.

[Asim 2008a] M. Asim, H. Mokhtar, and M. Merabti, "A Cellular Self-Organization Architecture for Wireless Sensor Networks," in *PG NET 2008* Liverpool, 2008a.

[Asim 2008b] M. Asim, H. Mokhtar, and M. Merabti, "A Fault Management Architecture for Wireless Sensor Networks," in *International Wireless Communications and Mobile Computing Conference, IWCMC '08* Crete Islan, Greece, 2008b.

[Asim 2009] M. Asim, H. Mokhtar, and M. Merabti, "A cellular approach to fault detection and recovery in wireless sensor networks," in *The Third International Conference on Sensor Technologies and Applications, SENSORCOMM 2009* Greece, 2009.

[Asim 2010a] M. Asim, H. Mokhtar, and M. Merabti, "A Self-Configurable Architecture for Wireless Sensor Networks," in *DeSE 2010 - Developments in E-Systems Engineering*, 2010a.

[Asim 2010b] M. Asim, H. Mokhtar, and M. Merabti, "Sensor Networks Management: A Survey," in *PGNet 2010* Liverpool, UK, 2010b.

[Baker 1981] D. J. Baker and A. Epheremides, "The Architectural Organization of a Moblie Radio Network via a Distributed Algorithm," in *IEEE Transactions on Communications*, 1981, pp. 1694-1701.

[Banerjee 2001] S. Banerjee and S. Khuller, "A Clustering Scheme for Hierarchical Control in Multi-hop Wireless Networks," in *in Proceedings of IEEE INFOCOM*, 2001, pp. 1028-1037.

[Bapat 2006] S. Bapat, "On reliable and scalable management of wireless sensor networks." vol. PhD Thesis: The Ohio State University, 2006.

[Bharathidasan 2002] A. Bharathidasan and V. Anand, "Sensor networks: An overview," Dept. of Computer Science, University of California at Davis, Technical Report CA 95616, 2002.

[Boulis 2003] A. Boulis and M. B. Srivastava, "Node-level Energy Management for Sensor Networks in the Presence of Multiple Applications," in *Proceedings of IEEE Intl. Conf. on Pervasive Computing and Communications (PerCom)* USA, 2003, pp. 41–49.

[Bulusu 2001] N. Bulusu, D. Estrin, L. Girod, and J. Heidemann, "Scalable coordination for Wireless Sensor Networks: self- configuring localization systems," in *Proceedings of the Sixth International Symposium on Communication Theory and Applications* UK, 2001, pp. 797-1800.

[Bunke 2006] H. Bunke, P. J. Dickinson, M.Kraetzl, and W. D. Wallis, *A Graph-Theoretic Approach to Enterprise Network Dynamics*: Birkhauser Boston, 2006.

[Carle 2004] J. Carle and D. Simplot-Ryl, "Energy-Efficient Area Monitoring for Sensor Networks," in *IEEE Computer Magazine*. vol. 37, 2004, pp. 40-46.

[Carzaniga 2001] A. Carzaniga, D. Rosenblum, and A. Wolf, "Design and Evaluation of a Wide-area Event Notification Service," in *ACM Transactions on Computer Systems*. vol. 19, 2001.

[Case 1990] J. Case, M. Fedor, M. Schoffstall, and J. Davin, "A Simple Network Management Protocol," in *Internet Standard RFC 1157*, 1990.

[Cecil 2006] A. Cecil, "A Summary of Network Traffic Monitoring and Analysis Techniques," Washington University in St. Louis, 2006.

[Cerpa 2001] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao, "Habitat monitoring: Application driver for wireless communications technology," in *ACM SIGCOMM Workshop on Data Communications* Latin America and the Caribbean, 2001, pp. 20-+.

[Cerpa 2004] A. Cerpa and D. Estrin, "ASCENT: Adaptive Self-Configuring sensor Networks Topologies," in *IEEE Transactions on Mobile Computing (TMC)*, 2004, pp. 272--285.

[Chatterjee 2002] M. Chatterjee, S. K. Das, and D. Turgut, "WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks," in *Clustering Computing*. vol. 5, 2002, pp. 193–204.

[Chen 2004] D. Chen and P. K. Varshney, "QoS support in wireless sensor networks: A survey," in *Proceedings of the 2004 International Conference on Wireless Networks (ICWN 2004)* Las Vegas, Nevada, USA, 2004.

[Chen 2006] J. Chen, S. Kher, and A. K. Somani, "Distributed Fault Detection of Wireless Sensor Networks," in *Proceedings of DIWANS 06*, 2006.

[Chen 2007a] J. L. Chen, H. F. Lu, and C. A. Lee, "Autonomic self-organization architecture for wireless sensor communications," *International Journal of Network Management*, vol. 17, pp. 197-208, 2007a.

[Chen 2007b] Y. M. Chen and S. H. Chang, "Purposeful Deployment via Self-Organizing Flocking Coalition in Sensor Networks," *International Journal Computer Science and Applications*, vol. 4, pp. 84-94, 2007b.

[Clouqueur 2004] T. Clouqueur, K.Saluja, and P. Ramanathan, "Fault Tolerance in Collaborative Sensor Networks for Target Detection," in *IEEE Transactions on Computers*, 2004, pp. 320-333.

[Coskun 2008] V. Coskun, "Relocating Sensor Nodes to Maximize Cumulative Connected Coverage in Wireless Sensor Networks," in *Sensors 2008*, 2008, pp. 2792-2817.

[Culler 2004] D. Culler, D. Estrin, and M. Srivastava, "Overview of Sensor Networks," in *IEEE Computer Magazine*, 2004, pp. 41-49.

[Deb 2001] B. Deb, S. Bhatnagar, and B. Nath, "A Topology Discovery Algorithm for Sensor Networks with Applications to Network Management," Rutgers University, Technical Report DCS-TR-441, May 2001.

[Deb 2004] B. Deb, S. Bhatnagar, and B. Nath, "STREAM: Sensor topology retrieval at multiple resolutions," in *Telecommunication Systems*, 2004, pp. 285-320.

[Dechene 2006] D. J. Dechene, A. E. Jardali, M. Luccini, and A. Sauer, "A Survey of Clustering Algorithms for Wireless Sensor Networks," Project Report 2006.

[Ding 2005] M. Ding, D. Chen, K. Xing, and X. Cheng, "Localized fault-tolerant event boundary detection in sensor networks," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '05)*. vol. 2 USA, 2005, pp. 902-913.

[Ekici 2006] E. Ekici, Y. Gu, and D. Bozdag, "Mobility-based communication in wireless sensor networks," in *IEEE Communications Magazine* vol. 44, 2006, pp. 56-62.

[Ephremides 1987] A. Ephremides, J. E. Wieselthier, and D. J. Baker, "A Design concept for Reliable Mobile Radio Networks with Frequency Hopping Signaling," in *Proceedings of the IEEE*, 1987, pp. 56-73.

[Erdogan 2003] A. Erdogan, E. Cayirci, and V. Coskun, "Sectoral Sweepers for Sensor Node Management and Location Estimation in Adhoc Sensor Networks," in *Proceedings of IEEE MILCOM Conf* Boston, Massachusetts, 2003, pp. 555-560.

[Estrin 2001] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2001)* Salt Lake City, Utah, 2001, pp. 2033-2036.

[Estrin 1999] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: Scalable coordination in sensor networks," in *Proceedings of ACM Mobicom*, 1999, pp. 263--270.

[Eugster 2003] P. Eugster, P. Felber, R. Guerraoui, and A. Kermarrec, "The Many Faces of Publish/Subscribe," in *ACM Computing Surveys*. vol. 35, 2003, pp. 114-131.

[Frye 2007] L. Frye and L. Cheng, "Network management of a wireless sensor network," Lehigh University, Technical Report LU-CSE-07-003, 2007.

[Gao 2005] T. Gao, D. Greenspan, and M. Welsh, "Vital Signs Monitoring and Patient Tracking over a Wireless Network," in *IEEE-EMBS 27th Annual Int. Conference of the Eng. in Medicine and Biology* Los Alamitos: IEEE Press, 2005, pp. 102–105.

[Gautam 2009] A. K. Gautam, "Accurate Localization Technique using Virtual Coordinate System in Wireless Sensor Networks," *International Journal of Recent Trends in Engineering*, vol. 2, 2009.

[Ge 2003] Z. Ge, P. Ji, J. Kurose, and D. Towsley, "Matchmaker: Signaling for Dynamic Publish/Subscribe Applications," in *11th IEEE International Conference on Network Protocols (ICNP)*, 2003.

[Gupta 2003a] G. Gupta and M. Younis, "Fault-Tolerant Clustering of Wireless Sensor Networks," in *Proceedings of the IEEE WCNC 2003* New Orleans, Louisiana, 2003a.

[Gupta 2003b] G. Gupta and M. Younis, "Load-Balanced Clustering in Wireless Sensor Networks," in *Proceedings of International Conference on Communication (ICC 2003)* Anchorage, AK, 2003b.

[Hansen 2006] E. Hansen, J. Neander, M. Nolin, and M. Björkman, "Energy-Efficient Cluster Formation for Large Sensor Networks using a Minimum Separation Distance," in *The Fifth Annual Mediterranean Ad Hoc Networking Workshop*, 2006.

[Hariri 2005] S. Hariri and M. Parashar, "Autonomic Computing: An overview," in *Springer-Verlag Berlin Heidelberg*, 2005, pp. 247–259.

[Harte 2005] S. Harte and A. Rahman., "Fault tolerance in sensor networks using self-diagnosing sensor nodes," in *In IEEE International Workshop on Intelligent Enviromnent*, 2005.

[Hartung 2006] C. Hartung, S. Holbrook, R. Han, and C. Seielstad, "Firewxnet: A multitiered portable wireless system for monitoring weather conditions in wildland fire environments," in *Proceedings of International Conference on Mobile systems, Applications and Services (MobiSys'06)*, 2006, pp. 28-41.

[He 2006] T. He, S. Krishnamurthy, L. Luo, T. Yan, L. Gu, R. Stoleru, G. Zhou, Q. Cao, P. Vicaire, J. A. Stankovic, T. Abdelzaher, J. Hui, and B. Krogh, "Vigilnet: An Integrated Sensor Network System for Energy-Efficient Surveillance," in *ACM Transactions on Sensor Networks*, 2006, pp. 1–38.

[Heinzelman 2000] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy- Efficient Communication Protocol for Wireless Sensor Networks," in *Proceedings of the 33th Hawaii International Conference on System Sciences*. vol. 2, 2000.

[Heinzelman 2002] W. R. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," in *IEEE Transactions on Wireless Communications*, 2002, pp. 660-670.

[Hsin 2006] C. Hsin and M. Liu, "Self-monitoring of Wireless Sensor Networks," in *Elsevier Computer Communications*. vol. 29, 2006, pp. 462-476.

[Ilyas 2004] M. Ilyas and I. Mahgoub, *Handbook of Sensor Networks Compact Wireless and Wired Sensing Systems*: CRC Press, USA, 2004.

[Jovanov 2005] E. Jovanov, A. Milenkovic, C. Otto, and P. C. d. Groen, "A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation," *Journal of Neuro Engineering and Rehabilitation,* p. 6, March 2005.

[Junas 2009] S. Junas, "Simulation & Analysis of a Power Aware Routing Protocol in Wireless Sensor Networks." vol. Master Thesis: Thapar University, Patiala, 2009.

[Kochhal 2003] M. Kochhal, L. Schwiebert, and S. Gupta, "Role-based hierarchical self organization for wireless ad hoc sensor networks," in *Proceeding of the 2nd ACM International Conference on Wireless Sensor Networks and Applications*, 2003, pp. 98-107.

[Koliousis 2007] A. Koliousis and J. Sventek, "Management of networked sensor systems," Department of Computing Science, University of Glasgow, Technical report TR-2007-231, 2007.

[Koushanfar 2002a] F. Koushanfar, M. Potkonjak, and A. Sangiovanni-Vincentelli, "Fault Tolerance in Wireless Ad-hoc Sensor Networks," in *Proceedings of IEEE Sensors*, 2002a.

[Koushanfar 2002b] F. Koushanfar, M. Potkonjak, and A. SangiovanniVincentelli, "Fault tolerance techniques in wireless ad-hoc sensor networks," UC Berkeley technical reports 2002b.

[Lee 2006a] W. L. Lee, A. Datta, and R. Cardell-Oliver, "Network Management in Wireless Sensor Networks," in *Handbook of Mobile Ad Hoc and Pervasive Communications*: American Scientific Publishers, 2006a.

[Lee 2006b] W. L. Lee, A. Datta, and R. Cardell-Oliver, "WinMS: Wireless Sensor Network-Management System, An Adaptive Policy-Based Management for Wireless Sensor Networks," School of Computer Science and Software Engineering, The University of Western Australia, Technical Report UWA-CSSE-06-01, 2006b.

[Lewis 2002] F. Lewis, "Wireless Sensor Networks," in *Smart Environments: Technology, Protocols and Applications*: Wiley-InterScience, 2002.

[Li 2006] X. Li and N. Santoro, "ZONER: A ZONE-based Sensor Relocation Protocol for Mobile Sensor Networks," in *Proceedings of IEEE LCN/WLN*, 2006, pp. 923-930.

[Li 2007] X. Li, N. Santoro, and I. Stojmenovic, "Mesh-based Sensor Relocation for Coverage Maintenance in Mobile Sensor Networks," in *Proceedings of the*

4th Int. Conf. on Ubiquitous Intelligence and Computing (UIC) (LNCS 4611), 2007, pp. 696-708.

[Lin 2000] H. C. Lin and Y. H. Chu, "A Clustering Technique for Large Multihop Mobile Wireless Networks," in *Proceedings of the IEEE Vehicular Technology Conference* Tokyo, Japan, 2000, pp. 1545-1549.

[Lopez 2008] J. Lopez and J. Zhou, *Wireless Sensor Network Security*: IOS Press, 2008.

[Malasri 2009] K. Malasri and L. Wang, "Design and Implementation of a Secure Wireless Mote-based Medical Sensor Network," in *Proceedings of the 10th international conference on Ubiquitous computing*, 2008, pp. 172-181.

[Marti 2000] S. Marti, T. J. Giuli, K.Lai, and M. Baker, "Mitigating routing misbehaviour in mobile ad hoc networks," in *ACM Mobicom*, 2000, pp. 255-265.

[Meer 2003] S. v. d. Meer, "Middleware and Management, Concepts, Characteristics and Integration," Telecommunication Software and Systems Group Waterford Institute of Technology, 2003.

[Morreale 2009] P. A. Morreale and K. Terplan, *CRC Handbook of Modern Telecommunications, Second Edition* CRC Press, 2009.

[Ould-Ahmed-Val 2005] E. M. Ould-Ahmed-Val, G. F. Riley, and B. S. Heck, "Simulation of large-scale sensor networks using GTSNetS," in *13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, 2005.

[Paradis 2007] L. Paradis and Q. Han, "A Survey of Fault Management in Wireless Sensor Networks," *Journal of Network and Systems Management*, vol. 15, pp. 171-190, 2007.

[Perillo 2003] M. Perillo and W. B. Heinzelman, "Providing Application QoS through Intelligent Sensor Management," in *Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications (SNPA '03)*, 2003.

[Piotrowski 2006] K. Piotrowski, P. Langendoerfer, and S. Peter, "How Public Key Cryptography Influences Wireless Sensor Node Lifetime," in *Proceedings of the fourth ACM workshop on Security of ad hoc and sensor networks* USA, 2006, pp. 169 – 176.

[Pottie 2000] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," in *Communication of the ACM*, 2000, pp. 51-58.

[Rabaey 2000] J. Rabaey, J. Ammer, J. L. d. Silva, and D. Patel, "PicoRadio: Ad-hoc Wireless Networking of Ubiquitous Low Energy Sensor/Monitor Nodes," in *IEEE Computer Society Workshop on VLSI*, 2000, pp. 9-12.

[Ramanathan 2005a] N. Ramanathan, K. Chang, E. Kohler, and D. Estrin, "Sympathy for the Sensor Network Debugger," in *Proceedings of 3rd ACM Conference on Embedded Networked Sensor Systems (SenSys '05)*, San Diego, California, 2005a, pp. 255-267.

[Ramanathan 2005b] N. Ramanathan, M. Yarvis, J. Chhabra, N. Kushalnagar, L. Krishnamurthy, and D. Estrin, "A Stream-oriented Power Management Protocol for Low Duty Cycle Sensor Network Applications," in *Proceedings of the IEEE Workshop on Embedded Networked Sensors* Sydney, Australia, 2005b.

[Riley 2003] G. Riley, "The Georgia Tech Network Simulator," in *ACM SIGCOMM Workshop on Models, Methods and Tools for Reproducible Network Research* Karlsruhe, Germany, 2003.

[Romer 2004] K. Romer and F. Mattern, "The design space of wireless sensor networks," in *IEEE Wireless Communications*, 2004, pp. 54-61.

[Ruiz 2004a] L. B. Ruiz, I. G.Siqueira, L. B. Oliveira, H. C. Wong, J. M. S. Nogueira, and A. A. F. Loureiro, "Fault management in event-driven wireless sensor networks," in *MSWiM'04* Italy, 2004a.

[Ruiz 2003] L. B. Ruiz, J. M. S. Nogueira, and A. A. Loureiro, "MANNA: A management architecture for wireless sensor networks," in *IEEE Communications Magazine*, 2003, pp. 116-125.

[Ruiz 2004b] L. B. Ruiz, F. B. Silva, T. R. M. Braga, J. M. S. Nogueira, and A. A. F. Loureiro, "On Impact of Management in Wireless Sensor Networks," in *Proceedings of the 9th IEEE/IFIP Network Operations and Management Symposium (NOMS' 04)* Seoul, Korea, 2004b, pp. 19-23.

[Sinha 2001] A. Sinha and A. Chandrakasan, "Dynamic power management in wireless sensor networks," in *IEEE design and test of computers*. vol. 18, 2001, pp. 62-74.

[Sohraby 2007] K. Sohraby, D. Minioli, and T. Znati, *Wireless Sensor Networks: Technology, Protocols and Applications*: Wiley, 2007.

[Song 2005] H. Song, D. Kim, K. Lee, and J. Sung, "Upnp-Based Sensor Network Management Architecture," in *Second International Conference on Mobile Computing and Ubiquitous Networking* Osaka, Japan, 2005.

[Song 2008] H. Song, S. Zhu, and G. Cao, "SVATS: A Sensor-Network-Based Vehicle Anti-Theft System," in *Proceedings of IEEE INFOCOM*, 2008, pp. 171-175.

[Ssu 2006] K. F. Ssu, C. H. Chou, H. C. Jiau, and W. T. Hu, "Detection and Diagnosis of data inconsistency failures in wireless sensor networks," in *Computer Networks*, 2006, pp. 1247-1260.

[Staddon 2002] J. Staddon, D. Balfanz, and G. Durfee, "Efficient Tracing of Failed Nodes in Sensor Networks," in *First ACM International Workshop on Wireless Sensor Networks and Applications* USA, 2002.

[Subramanian 2000] L. Subramanian and R. H. Katz, "An architecture for building self-configurable systems," in *Proceeding of the IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHOC 2000)*, 2000, pp. 63-73.

[TmoteSky] TmoteSky, "Available at http://www.sentilla.com/moteiv-transition.html," in *Access date: 10/10/2010*.

[Tolle 2005] G. Tolle and D. Culler, "Design of an Application-Cooperative Management System for Wireless Sensor Networks," in *Proceedings of the 2nd European Workshop on Wireless Sensor Networks (EWSN 2005)* Istanbul, Turkey, 2005.

[TOM 2000] TOM: TeleManagement Forum Specification, Telecomm Operations Map (GB910), Approved Version 2.1, March 2000, 2000.

[Tynan 2005] R. Tynan, D. Marsh, D. OKane, and G. M. P. OHare, "Agents for Wireless Sensor Network Power Management," in *Proceedings of ICPP, IEEE* Norway, 2005, pp. 413-418.

[Uchida 2008] J. Uchida, I. Muzahidul, Y. Katayama, W. Chen, and K. Wada, "Construction and Maintenance of a Cluster-Based Architecture for Sensor Networks," in *Proceedings of the 39th Annual Hawaii International Conference on System Sciences*, 2008.

[Venkataraman 2007] G. Venkataraman, S. Emmanuel, and S.Thambipillai, "Size-restricted cluster formation and cluster maintenance technique for mobile ad hoc networks," in *International Journal of Network Management*, 2007, pp. 171 - 194

[Venkataraman 2008] G. Venkataraman, S. Emmanuel, and S.Thambipillai, "Energy-efficient cluster-based scheme for failure management in sensor networks," in *IET Communications*. vol. 2, 2008, pp. 528-537.

[Venkataraman 2005] G. Venkataraman, S. Emmanuel, and T. Srikanthan, "DASCA: A Degree and Size based Clustering Approach for Wireless Sensor

Networks," in *Proceedings of IEEE, International Symposium of Wireless Communication Systems (ISWCS 2005)* Siena, 2005, pp. 508-512.

[Wang 2003a] F. Wang, Q.Tian, Q. Gao, and Q.Pan, "A Study of Sensor Management Based on Sensor Networks," in *Processing of the 2003 IEEE International Conference on Robotics, Intelligent Systems and Signal Processing* Changsha, China, 2003a, pp. 1058-1062.

[Wang 2003b] G. Wang, G. Cao, and T. L. Porta, "A bidding Protocol for Devolping Mobile Sensors," in *11th IEEE Internation Conference on Network Protocols (ICNP)*, 2003b.

[Wang 2004] G. Wang, G. Cao, and T. L. Porta, "Proxy-Based Sensor Deployment for Mobile Sensor Networks," in *Proceedings of IEEE MASS*, 2004, pp. 493-502.

[Wang 2006a] G. Wang, G. Cao, and T. L. Porta, "Movement-Assisted Sensor Deployment," in *INFOCOM*, 2006a.

[Wang 2005] G. Wang, G. Cao, T. L. Porta, and W. Zhang, "Sensor Relocation in Mobile Sensor Networks," in *IEEE INFOCOM*. vol. 4, 2005, pp. 2305-2312.

[Wang 2006b] W. Wang, V. Srinivasan, and K. C. Chua, "Trade-offs Between Mobility and Density for Coverage in Wireless Sensor Networks," in *Proceedings of the 13th annual ACM international conference on Mobile computing and networking* Canada, 2006b.

[Wen 2006] C. Y. Wen and W. A. Sethares, "Adaptive decentralized re-clustering for wireless sensor networks," in *SMC* Taipai, Taiwan, 2006, pp. 709-2716.

[Wong 2004] T. Wong, T. Tsuchiya, and T. Kikuno, "A self-organizing technique for sensor placement in wireless micro-sensor networks," in *Proceedings of the International Conference on Advanced Information Networking and Applications* Fukuoka, Kyushu, Japan, 2004, pp. 78 - 83.

[Wood 2006] A. Wood, G. Virone, T. Doan, Q. Cao, L. Selavo, Y. Wu, L. Fang, Z. He, S. Lin, and J. Stankovic, "ALARM-NET: wireless sensor networks for assisted-living and residential monitoring," Department of Computer Science, University of Virginia, Technical Report CS-2006-11, 2006.

[Xia 2008] F. Xia, "QoS Challenges and Opportunities in Wireless Sensor/Actuator Networks," in *Sensors 2008*. vol. 8, 2008, pp. 1099-1110.

[Xu 2001] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking* Rome, Italy, 2001, pp. 70--84.