Self Organizing Sensor Network: Deployment and Relocation

A Dissertation submitted in partial fulfillment of
the requirement for the award of

MASTER OF ENGINEERING
IN
ELECTRONICS AND COMMUNICATION ENGINEERING
OF
DELHI UNIVERSITY

By

Mrs Saji. M. Antony
21/E&C/06
UNIVERSITY ROLL NO:12673
Under the guidance of
Mrs S. Indu
(Senior Lecturer)

Department of Electronics and Communication Engineering
Delhi College of Engineering
Bawana Road, Delhi –110 042.

2006 - 2009

CERTIFICATE

This is to certify that Mrs Saji. M. Antony, student of Delhi College of Engineering, Delhi worked under the guidance of Mrs. S. Indu (Senior Lecturer, Department of Electronics and Communication Engineering) on major project entitled "Self Organizing Sensor Network : Deployment and Relocation". This work is being submitted in partial fulfillment of the requirement for the award of the degree of "MASTER OF ENGINEERING" to the Department of Electronics and Communication Engineering, Delhi College of Engineering, Delhi – 42.

The work presented in this project is a record of bonafide work carried out by me under the guidance and supervision of Delhi College of Engineering, Delhi. The matter embodied in this dissertation has not been submitted elsewhere, either in part or full, for the award of any other degree.

Saji. M. Antony
Class Roll No : 21/E&C/06

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Mrs. S. Indu
Senior Lecturer,
Deptt. of Electronics & Comm Engineering,

Prof. Asok Bhattacharya,
Head of Department,
Deptt. of Electronics & Comm Engineering,
Delhi College of Engineering,
Bawana Road, Delhi.

Acknowledgments

I would like to express my deepest gratitude to all the people who have supported and encouraged me during the course of this project without which, this work could not have been accomplished. First I am very grateful to my project guide **Mrs. S. Indu** (Senior Lecturer, Deptt. of Electronics and Communication Engineering) for giving me the opportunity of carrying out this project under her guidance. Her insight into the field of Sensor Networks made it easier to understand the nuances and helped me achieve my goals.

I would also like to express my sincere gratitude to the **Prof. Asok Bhattacharya,** Head of Department and the faculty of Electronics and Communication Engineering Department, Delhi College of Engineering.

I also express my sincere gratitude to Prof. Anuradha Basu and my colleagues at the Department of Electronics and Communication Engineering, Bharati Vidyapeeth`s College of Engineering, New Delhi for their guidance and support.

Last but not the least; I am thankful to my parents, husband and friends, for their encouragement and guidance.

Saji. M. Antony

June, 2009.

ABSTRACT

Sensor networks have been identified as one of the most important technologies of the 21st century. Advances in micro-electro-mechanical systems (MEMS), electronics and wireless communications have enabled the development of low-cost, low-power, multifunctional sensor nodes that are small in size and have capabilities of sensing, data processing, communication, storage etc. A sensor network is composed of a large number of such nodes densely deployed in a region to monitor a particular phenomenon. Sensor network is a fresh research area with applications in military, environment monitoring, disaster management etc. Sensors are used to measure and/or monitor parameters that may vary with place and time which prompts the need for the Dynamic Sensor Network (DSN). Self-Organized Sensor Network (SOSN) is a kind of DSN, which can achieve the necessary organizational structures without requiring human intervention in order to sense the time varying spatial signal efficiently. In this project we address one of the fundamental problems of SOSN, namely Coverage, which reflects how well a sensor network is monitored or tracked by sensors. We divide this sensor coverage problem as sensor deployment and sensor relocation. We propose a partial distributed algorithm for sensor deployment, which assures movement of redundant sensors to cover multiple events.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Prologue

A wireless sensor network (WSN) is a wireless network consisting of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations. The development of wireless sensor networks was originally motivated by military applications such as battlefield surveillance. However, wireless sensor networks are now used in many civilian application areas, including environment and habitat monitoring, healthcare applications, home automation, and traffic control.

Sensor networks are inherently different from standard communication networks because their aim is to monitor a phenomena over space and time, and not just send data from one node to another i.e. the underlying communication infrastructure is used to achieve a larger objective of spatio-temporal sampling and the subsequent detection and classification of events of interest.

Presently, sensor networks are mostly used to sense phenomena and therefore act as information sources only. Some conditions may require that a related action be performed on the occurrence of certain events; such as mobilizing swarm or robots in case a target is detected or activating water sprinklers in case of a fire. In such cases it is required that the actuation needed in response to events should be automatically generated without any human intervention. Moreover, it is desirable that the network design should be as distributed as possible for it to be fast acting and fault tolerant at the same time. The work presented in this report is one such effort towards assisting the design of an autonomous sensor-actuator network.

1.2 Scope and Objective of the Project

The scope of the current project was to assist designing a self organizing Sensor-Network that uses mobile sensor nodes. Event coverage can be maximized by moving redundant sensors without leaving any region unsensed.

The main objectives of the project were to develop a deployment algorithm for initial coverage and a frame work for sensor relocation to cover multiple events.

1.3 Sensor Networks

Recent advances in computer technology have made it possible to develop computers the size of a coin that are not only capable of performing complex tasks but also come shipped with an on-board radio for communication and support a host of sensors for monitoring various phenomena. Sensor nodes are capable of sensing many types of information from the environment, including temperature; light; humidity; radiation; the presence or nature of biological organisms; geological features; seismic vibrations; specific types of computer data; and more[8]. Very small in size, these nodes have embedded processing ability and can have multiple on-board sensors operating in the acoustic, seismic, infrared (IR) and magnetic modes, as well as imagers and micro-radars. Also onboard are storage, wireless links to neighboring nodes, and location and position knowledge through the global positioning system (GPS) or local positioning algorithms [12]. The sensor nodes are therefore capable of gathering, processing, storing and communicating information to other nodes and to the outside world.

A wireless sensor network is composed of a large number of such sensor nodes, deployed densely either inside a phenomenon or very close to it. It represents a significant improvement over the traditional sensors.

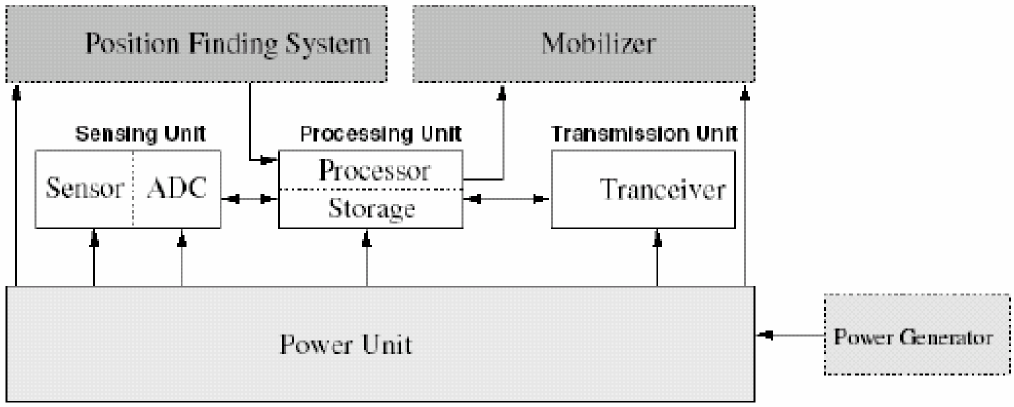


Figure 1.1: The Components of a Sensor Node

Compared to the use of a few expensive (but highly accurate) sensors, the strategy of deploying a large number of inexpensive sensors has significant advantages, at smaller or comparable total system cost: much higher spatial resolution; higher robustness against failures through distributed operation; uniform coverage; small obtrusiveness; ease of deployment; reduced energy consumption; and, consequently, increased system lifetime. The position of sensor nodes need not be engineered or predetermined. This allows random deployment in inaccessible terrains or disaster relief operations. These could involve: in the air, under water, on bodies, in vehicles, and inside buildings. On the other hand, this also means that sensor network protocols and algorithms must possess self-organizing capabilities. Self-organization means that the system can achieve the necessary organizational structures without requiring human intervention, i.e., the sensor network should be able to carry out functional operations through cooperation among individual nodes rather than set up and operated by human operators.

Another unique feature of sensor networks is the cooperative effort of sensor nodes in gathering the data. Sensor nodes are fitted with an onboard processor. They are usually battery based, with limited energy resources and capabilities; and it is difficult or unpractical to recharge each node. Instead of sending the raw data to the nodes responsible for the fusion, they use their processing abilities to locally carry out simple computations and transmit only the required and partially processed data.

1.3.1 Features

Limited Energy Resources

As with any embedded system, sensor nodes also face the challenge of reducing energy consumption, as much as possible, to maximize its operational lifetime. A traditional hand held, consumer electronic class, embedded system has the provision of recharging whenever its batteries drain out, but this luxury is not available to a sensor node. This is due to the fact that most of the time a sensor node will be deployed in a remote area where a battery change is either impossible or costs more than the node itself. Another factor that highlights the importance of judicious use of energy resources is the fact that battery technology itself hasn’t advanced as much as the computers in the last few decades. Under these conditions it is imperative that software written for sensor networks should be energy aware.

Limited computational resources

An important requirement for a sensor node is low form factor [19]. This is essential because in a typical scenario hundreds or thousands of such sensors need to be sprinkled over the target region. Also, applications such as health monitoring of animals or humans, require the sensor nodes to be unobtrusive in the normal functioning of the individual. Micro-electro-mechanical systems (MEMS) have made it possible to bring the size of sensors and actuators down to the millimeter scale. Similarly, advances in VLSI technology have made button sized computers a reality. Such devices, however, are designed with low cost in mind and as a result the computational resources, such as CPU speed and memory are limited [18]. The amount of computational resources on a node determines the amount of computation that can be performed in a given unit of time. This clearly influences the design of algorithms and data structures that run on these nodes. This calls for collaborative data processing among neighboring nodes, putting pressure on the communications channel, which also is at premium. Thus, any software designed for a sensor node should maintain a neat balance between the amount of local and distributed processing, taking into amount the task deadlines and available computation and communication resources.

Autonomous Operation

Deploying a sensor network in isolated environments requires that the network be capable of self-configuring itself as its goals change, nodes add or drop, or radio bandwidth changes. Self-configuration is not only necessary, but rather implicitly tied with the very definition of sensor networks. Almost any sensor network, deployed today, will have the capability to automatically boot up and select its goals. This also involves nodes allocating the tasks among themselves and subsequently relaying the results to a base station. Any query generated at a base station, automatically routes to a region of interest, even as nodes or links fail. More capable networks have the capability to self heal, i.e. healthy nodes take up the position / allotted tasks of the nodes that fail.

To summarize, some of the basic features of sensor networks are:

- Self-organizing capabilities
- Short-range broadcast communication and multi hop routing
- Dense deployment and cooperative effort of sensor nodes
- Frequently changing topology due to fading and node failures
- Limitations in energy, transmit power, memory, and computing power.

These characteristics [12] make sensor networks different from other wireless networks.

1.3.2 Opportunities

Growing Research and Commercial Interest

Research and commercial interest in the area of wireless sensor networks are currently growing exponentially, which is manifested in many ways:

• The number of Web pages (Google: 26,000 hits for sensor networks; 8000 for wireless sensor networks in August 2003)

• The increasing number of dedicated annual workshops, such as IPSN (information /processing in sensor networks); SenSys; EWSN (European workshop on wireless sensor networks); SNPA (sensor network protocols and sensor networks and applications)

• Conference sessions on sensor networks in the communications and mobile computing communities (ISIT, ICC, Globecom, INFOCOM, VTC, MobiCom, MobiHoc)

• Research projects funded by NSF (apart from ongoing programs, a new specific effort now focuses on sensors and sensor networks) and DARPA through its SensIT (sensor information technology), NEST (networked embedded software technology), MSET (multisensor exploitation), UGS (unattended ground sensors), NETEX (networking in extreme environments), ISP (integrated sensing and processing), and communicator programs Special issues and sections in renowned journals are common, e.g., in the IEEE Proceedings and signal processing, communications, and networking magazines. Commercial interest is reflected in investments by established companies as well as start-ups that offer general and specific hardware and software solutions.

• Compared to the use of a few expensive (but highly accurate) sensors, the strategy of deploying a large number of inexpensive sensors has significant advantages, at smaller or comparable total system cost: much higher spatial resolution; higher robustness against failures through distributed operation;

uniform coverage; small obtrusiveness; ease of deployment; reduced energy consumption; and, consequently, increased system lifetime. The main point is to position sensors close to the source of a potential problem phenomenon, where the acquired data are likely to have the greatest benefit or impact. Pure sensing in a fine-grained manner may revolutionize the way in which complex physical systems are understood. The addition of actuators, however, opens a completely new dimension by permitting management and manipulation of the environment at a scale that offers enormous opportunities for almost every scientific discipline. Indeed, Business 2.0 (http://www.business2.com/) lists sensor robots as one of "six technologies that will change the world," and Technology Review at MIT and Globalfuture identify WSNs as one of the "10 emerging technologies that will change the world" (http://www.globalfuture. com/mit-trends2003.htm). The combination of sensor network technology with MEMS and nanotechnology will greatly reduce the size of the nodes and enhance the capabilities of the network. The remainder of this chapter lists and briefly describes a number of applications for wireless sensor networks, grouped into different categories. However, because the number of areas of application is growing rapidly, every attempt at compiling an exhaustive list is bound to fail.

1.3.3 Performance Metrics

To discuss the issues in more detail, it is necessary to examine a list of metrics that determine the performance of a sensor network:

• **Energy efficiency/system lifetime :** The sensors are battery operated, rendering energy a very scarce resource that must be wisely managed in order to extend the lifetime of the network.

• **Latency:** Many sensor applications require delay-guaranteed service. Protocols must ensure that sensed data will be delivered to the user within a certain delay. Prominent examples in this class of networks are certainly the sensor-actuator networks.

• **Accuracy:** Obtaining accurate information is the primary objective; accuracy can be improved through joint detection and estimation. Rate distortion theory is a possible tool to assess accuracy.

• **Fault tolerance:** Robustness to sensor and link failures must be achieved through redundancy and collaborative processing and communication.

• **Scalability:** Because a sensor network may contain thousands of nodes, scalability is a critical factor that guarantees that the network performance does not significantly degrade as the network size (or node density) increases.

• **Transport capacity/throughput:** Because most sensor data must be delivered to a single base station or fusion center, a critical area in the sensor network exists, whose sensor nodes must relay the data generated by virtually all nodes in the network. Thus, the traffic load at those critical nodes is heavy, even when the average traffic rate is low. Apparently, this area has a paramount influence on system lifetime, packet end-to-end delay, and scalability. Because of the interdependence of energy consumption, delay, and throughput, all these issues and metrics are tightly coupled. Thus, the design of a WSN necessarily consists of the resolution of numerous trade-offs, which also reflects in the network protocol stack, in which a cross-layer approach is needed instead of the traditional layer-by-layer protocol design.

1.3.4 Energy Consumption

To model energy consumption, four basic different states of a node can be identified: transmission, reception, listening, and sleeping. They consist of the following tasks:

• **Acquisition:** Sensing, A/D conversion, preprocessing, and perhaps storing.

• **Transmission:** Processing for address determination, packetization, encoding, framing, and maybe queuing; supply for the baseband and RF circuitry. (The nonlinearity of the power amplifier must be taken into account because the power consumption is most likely not proportional to the transmit power.)

• **Reception:** Low-noise amplifier, down converter oscillator, filtering, detection, decoding, error detection, and address check; reception even if a node is not the intended receiver.

• **Listening:** Similar to reception except that the signal processing chain stops at the detection.

1.3.5 Node Distribution and Mobility

Regular grids (square, triangle, hexagon) and uniformly random distributions are widely used analytically tractable models. The latter can be problematic because nodes can be arbitrarily close, leading to unrealistic received power levels if the path attenuation is assumed to be proportional to distance. Regular grids overlaid with Gaussian variations in the positions may be more accurate. Generic mobility models for WSNs are difficult to define because they are highly application specific, so this issue must be studied on a case-by-case basis.

1.3.6 Traffic

Often, simulation work is based on constant bit rate traffic for convenience, but this is most probably not the typical traffic class. Models for bursty many-to-one traffic are needed, but they certainly depend strongly on the application.

1.3.7 Connectivity

Network connectivity is an important issue because it is crucial for most applications that the network is not partitioned into disjoint parts. If the nodes' positions are modeled as a Poisson point process in two dimensions (which, for all practical purposes, corresponds to a uniformly random distribution), the problem of connectivity has been studied using the tool of continuum percolation theory. For large networks, the phenomenon of a sharp phase transition can be observed: the probability that the network percolates jumps abruptly from almost 0 to almost 1 as soon as the density of the network is bigger than some critical value. Most such results are based on the geometric disk abstraction. It is conjectured, though, that other connectivity functions lead to better connectivity, i.e., the disk is apparently the hardest shape to connect. A practical consequence of this conjecture is that fading results in improved connectivity. Recent work also discusses the impact of interference. The simplifying assumptions necessary to achieve these results leave many open problems.

1.3.8 Quality of Service

Quality of service refers to the capability of a network to deliver data reliably and timely. A high quantity of service, i.e., throughput or transport capacity, is generally not sufficient to satisfy an application's delay requirements. Consequently, the speed of propagation of information may be as crucial as the throughput. Accordingly, in addition to network capacity, an important issue in many WSNs is that of quality of Service (QoS) guarantees. Previous QoS-related work in wireless networks mostly focused on delay QoS, in a broader sense, consists of the triple (R, Pe, D), where R denotes throughput; Pe denotes reliability as measured by, for example, bit error probability or packet loss probability; and D denotes delay.

1.3.9 Security

Depending on the application, security can be critical. The network should enable intrusion detection and tolerance as well as robust operation in the case of failure because, often, the sensor nodes are not protected against physical mishandling or attacks. Eavesdropping, jamming, and listen-and-retransmit attacks can hamper or prevent the operation; therefore, access control, message integrity, and confidentiality must be guaranteed.

1.3.10 Implementation

Companies such as Crossbow, Ember, Sensoria, and Millenial are building small sensor nodes with wireless capabilities. However, a per-node cost of $100 to $200 (not including sophisticated sensors) is prohibitive for large networks. Nodes must become an order of magnitude cheaper in order to render applications with a large number of nodes affordable. With the current pace of progress in VLSI and MEMS technology, this is bound to happen in the next few years. The fusion of MEMS and electronics onto a single chip, however, still poses difficulties. Miniaturization will make steady progress, except for two crucial components: the antenna and the battery, where it will be very challenging to find innovative solutions. Furthermore, the impact of the hardware on optimum protocol design is largely an open topic. The characteristics of the power amplifier, for example, greatly influence the energy efficiency of routing algorithms.

1.3.11 Other Issues

- **Distributed signal processing:** Most tasks require the combined effort of multiple network nodes, which requires protocols that provide coordination, efficient local exchange of information, and, possibly, hierarchical operation.
- **Synchronization and localization:** The notion of time is critical. Coordinated sensing and actuating in the physical world require a sense of global time that must be paired with relative or absolute knowledge of nodes' locations.

• **Wireless reprogramming:** A deployed WSN may need to be reprogrammed or updated. So far, no networking protocols are available to carry out such a task reliably in a multi-hop network. The main difficulty is the acknowledgment of packets in such a joint multi-hop/multicast communication.

1.4 Applications

All these features ensure a wide range of applications for sensor networks. Though the number of areas of application is growing rapidly, some of the key application areas are listed below [12]:

1.4.1 General Engineering

• **Automotive telematics:** Cars, which comprise a network of dozens of sensors and actuators, are networked into a system of systems to improve the safety and efficiency of traffic.

• **Fingertip accelerometer virtual keyboards:** These devices may replace the conventional input devices for PCs and musical instruments.

• **Sensing and maintenance in industrial plants:** Complex industrial robots are equipped with up to 200 sensors that are usually connected by cables to a main computer. Because cables are expensive and subject to wear and tear caused by the robot's movement, companies are replacing them by wireless connections. By mounting small coils on the sensor nodes, the principle of induction is exploited to solve the power supply problem.

• **Aircraft drag reduction:** Engineers can achieve this by combining flow sensors and blowing/sucking actuators mounted on the wings of an airplane.

- **Smart office spaces:** Areas are equipped with light, temperature, and movement sensors, microphones for voice activation, and pressure sensors in chairs. Air flow and temperature can be regulated locally for one room rather than centrally.
- **Tracking of goods in retail stores:** Tagging facilitates the store and warehouse management.
- **Tracking of containers and boxes:** Shipping companies are assisted in keeping track of their goods, at least until they move out of range of other goods.
- **Social studies:** Equipping human beings with sensor nodes permits interesting studies of human interaction and social behavior.

1.4.2 Agriculture and Environmental Monitoring

- **Precision agriculture:** Crop and livestock management and precise control of fertilizer concentrations are possible.
- **Planetary exploration:** Exploration and surveillance in inhospitable environments such as remote geographic regions or toxic locations can take place.
- **Geophysical monitoring:** Seismic activity can be detected at a much finer scale using a network of sensors equipped with accelerometers.
- **Monitoring of freshwater quality:** The field of hydrochemistry has a compelling need for sensor networks because of the complex spatiotemporal variability in hydrologic, chemical, and ecological parameters and the difficulty of labor-intensive sampling, particularly in remote locations or

under adverse conditions. In addition, buoys along the coast could alert surfers, swimmers, and fishermen to dangerous levels of bacteria.

- **Zebranet:** The Zebranet project at Princeton aims at tracking the movement of zebras in Africa.
- **Disaster detection:** Forest fire and floods can be detected early and causes can be localized precisely by densely deployed sensor networks.
- **Remote exploration:** Exploration and surveillance in inhospitable environments such as remote geographic regions or toxic locations.

1.4.3 Civil Engineering

- **Monitoring of structures:** Sensors will be placed in bridges to detect and warn of structural weakness and in water reservoirs to spot hazardous materials. The reaction of tall buildings to wind and earthquakes can be studied and material fatigue can be monitored closely.
- **Urban planning:** Urban planners will track groundwater patterns and how much carbon dioxide cities are expelling, enabling them to make better land-use decisions.
- **Disaster recovery:** Buildings razed by an earthquake may be infiltrated with sensor robots to locate signs of life.
- **Habitat monitoring:** Sensors are deployed to to measure humidity, pressure, temperature, infrared radiation, total solar radiation, and photo synthetically active radiation (http:// www. greatduckisland.net/).
- **Contaminant transport:** The assessment of exposure levels requires high spatial and temporal sampling rates, which can be provided by WSNs.

1.4.4 Military Applications

- **Asset monitoring and management:** Commanders can monitor the status and locations of troops, weapons, and supplies to improve military command, control, communications, and computing.
- **Urban warfare:** Sensors are deployed in buildings that have been cleared to prevent reoccupation; movements of friend and foe are displayed in PDA-like devices carried by soldiers. Snipers can be localized by the collaborative effort of multiple acoustic sensors.
- **Protection:** Sensitive objects such as atomic plants, bridges, retaining walls, oil and gas pipelines, communication towers, ammunition depots, and military headquarters can be protected by intelligent sensor fields able to discriminate between different classes of intruders. Biological and chemical attacks can be detected early or even prevented by a sensor network acting as a warning system.
- **Self-healing minefields:** The self-healing minefield system is designed to achieve an increased resistance to dismounted and mounted breaching by adding a novel dimension to the minefield. The self-healing minefield is an intelligent, dynamic obstacle that senses relative positions and responds to an enemy's breaching attempt by physical reorganization.
- **Surveillance and battle-space monitoring:** Vibration and magnetic sensors can report vehicle and personnel movement, permitting close surveillance of opposing forces.

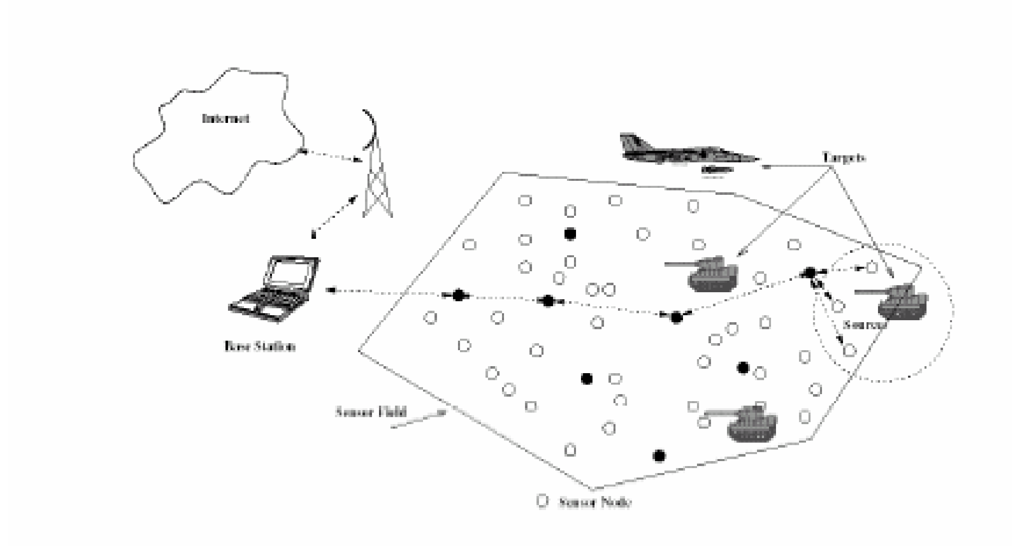


Figure 1.2: Sensor network used in military application [12].

A set of sensor nodes (black circles) are selected to work as data aggregators; through them data are sent to the external base station. If an Internet connection is available, a quality copy of the readings can be sent through the Internet to the central command.

1.4.5 Health Monitoring and Surgery

- **Medical sensing:** Physiological data such as body temperature, blood pressure, and pulse are sensed and automatically transmitted to a computer or physician, where it can be used for health status monitoring and medical exploration. Tiny sensors in the blood stream, possibly powered by a weak external electromagnetic field, can continuously analyze the blood and prevent coagulation and thrombosis.
- **Micro-surgery:** A swarm of MEMS-based robots may collaborate to perform microscopic and minimally invasive surgery. The opportunities for wireless sensor networks are ubiquitous. However, a number of formidable

challenges must be solved before these exciting applications may become reality.

1.5 Motivation

The motivation for the present work comes from the need to build an autonomous system, using mobile nodes, for tracking time varying spatial events/targets. A key feature of our approach will be the focus on complete sensing field coverage with the given nodes.

For detecting the events/targets, adaptive sampling method in both space and time will be carried out in an energy efficient manner.

Another area where we shall like to optimize is the node movements during sensor relocation while covering events with multiple sensors, since mobility costs contribute significantly to the total power consumption.

1.6 Chapter Layout

The first chapter is introduction in which we start our discussion by looking, in detail, wireless sensor networks and challenges and opportunities in this field and provides the motivation behind the present work.

Literature review is presented in chapter 2.

Chapter 3 surveys what self organization is, and how can we apply the principles of biological self organization to design sensor networks.

Chapter 4 concentrates on 'Deployment phase' of sensor networks. It briefly explains distributed deployment algorithm.

Chapter 5 is about ‘Event detection and Relocation phase’of sensor networks and explains the working of algorithms for sensor relocation that we have used in this project. In this chapter, we also propose a frame work for sensor relocation to cover the multiple events.

In chapter 6 we analyze the results and observations.

We conclude the report in chapter 7 and also look at what lies ahead.

Chapter 2

State of the Art

The current chapter discusses the state of art with respect to the problem addressed in this work. The discussion is divided under the categories of coverage, mobile nodes, multi sensor detection and energy efficiency. Many research efforts are being directed towards overcoming the common issues that are peculiar to sensor networks. Such efforts are focused equally in improving the hardware as well as designing energy efficient, intelligent software.

Sensor networks are self sustaining systems of nodes that co-ordinate amongst themselves autonomously but, their development is hindered by the constraints of the devices used. Firstly, they are power constrained which makes device failure inevitable and energy efficient communication essential. They also have limited computing power, preventing sophisticated network protocols from being run, and limited bandwidth which constraints the amount of communication. Human intervention to keep the network up and running, in such conditions, is at the least a tedious job and mostly infeasible. It is for this reason that there is a continued effort to make sensor networks as autonomous as possible.

Recently, many efforts have been directed towards combining robotics with sensor networks for applications such as target tracking or surveillance. Mobile nodes can be brought together or spread away to increase the fidelity or area of coverage, respectively, making the sensor network truly dynamic.

2.1 Coverage

The effectiveness of a distributed self organizing mobile wireless sensor network depends to a large extent on coverage provided by sensor deployment. [2] considers the fundamental problem of a dynamic sensor network, which is the coverage. Localization of mobile sensors is a challenging issue and it is solved by developing a novel spatial addressing scheme for locating mobile sensors relative to the field. [2] uses hexagonal cell based mobile Ad Hoc network for localization, which maximizes the coverage with minimum overlapping.

Random placement of sensors may not satisfy the deployment requirement due to the hostile deployment environment. Two methods can be used to enhance the coverage: incremental sensor deployment and movement-assisted sensor deployment. In incremental sensor deployment [4], nodes are deployed one by one, using the location information of previously deployed nodes to deploy the current one. This algorithm is not scalable and is computationally expensive. Most existing movement-assisted sensor deployment protocols rely on the notion of virtual force to move existing sensors from an initial unbalanced state to a balanced state. Sensors are involved in a sequence of computation (for a new position) and movement.

In [16], a centralized virtual force based mobile sensor deployment algorithm (VFA) was proposed, which combines the idea of potential field and disk packing. In VFA, there is a powerful cluster head, which will communicate with all the other sensors, collect sensor position information, and calculate forces and desired position for each sensor. In [9], a robot works in coordination with a sensor network. The sensor network assists the robot in navigation and the robot deploys additional sensors to maximize the sensor coverage of the network. [6] applies the potential field technique in a centralized way. One powerful leader is used to calculate the field and generate the location for all other nodes. This algorithm assumes that the environment is static and known before deployment.

In [11], a novel distributed self-deployment protocol for mobile sensors was proposed. They used Voronoi diagrams [10] to find coverage holes in the sensor network, and proposed three algorithms, VEC (Vector-based), VOR (Voronoi based), and Minimax, to guide sensor movement toward the coverage hole. When applied to randomly deployed sensors, these algorithms can provide high coverage within a short time and limited moving distance. If the initial distribution of the sensors is extremely uneven, disconnection may occur, thus, the Voronoi polygon constructed may not be accurate enough, which results in more moves and larger moving distance.

The algorithm proposed in [7] randomly partitions space into sufficiently small neighborhoods at each iteration. Within each neighborhood a redistribution process directed by a cluster head is enacted. But in our approach we divide the space into fixed hexagonal cells at starting of the deployment algorithm which is different from [7] and the head node of the cell will distribute the nodes to the neighboring void cells. The main advantage of our algorithm when compared to [7] is that we are not forming the clusters at every iteration which has high message complexity.

In [11] every node determines its final location according to its initial location and the event distribution; hence the motion of each node can be predicted by any other node. However, the approaches in [3] assume that the initial distribution is uniform and depend on every sensor node predicting the motion of other nodes, which may be expensive. To predict the motion, each node also needs to know the initial location of all other nodes and all nodes need to know and agree on the locations of all events. Such information must be flooded across the whole network. Our approach makes no assumptions on the initial distribution of the sensor nodes and no information flooding is needed. Also Voronoi approach needs a very high computational resource which is not needed in our algorithm. [29] uses the idea of 'social potentials' where the potentials for a robot are constructed with respect to the other robots. The authors describe heuristics to design social potentials for achieving a variety of behaviors like clustering, patrolling, etc. This method does not aim at maximizing the coverage area .

2.2 Mobile Nodes

Recently, many efforts have been directed towards combining robotics with sensor networks for applications such as target tracking or surveillance. Mobile nodes can be brought together or spread away to increase the fidelity or area of coverage, respectively, making the sensor network truly dynamic. [1] discusses the use of a data mule for increasing the energy efficiency and coverage in under water, thermo cline detection. By moving the nodes from regions with low or no activity, to that with high activity, one can utilize a node's resources to the fullest. This also means that fewer nodes are needed to achieve the same effect as with a static network.

2.3. Multi-Sensor Detection

Interest in multi-sensor detection and estimation has surfaced with anticipated applications in multiple-target detection and estimation using multiple sensors, which may be geographically, dispersed thus providing spatio-temporal sampling. In classical multi-sensor detection and estimation, it is assumed that all the local sensors communicate the sampled data to a central processor that performs optimal detection and tracking of targets based on conventional statistical techniques [28]. In distributed processing, some preliminary processing of data r_i is carried out at each sensor s_i and condensed information u_i is sent from each sensor to other sensors and ultimately to the central processor which is often known as the fusion center [27]. In the terminology of distributed sensor networks, we say that the network has intelligence at each node.

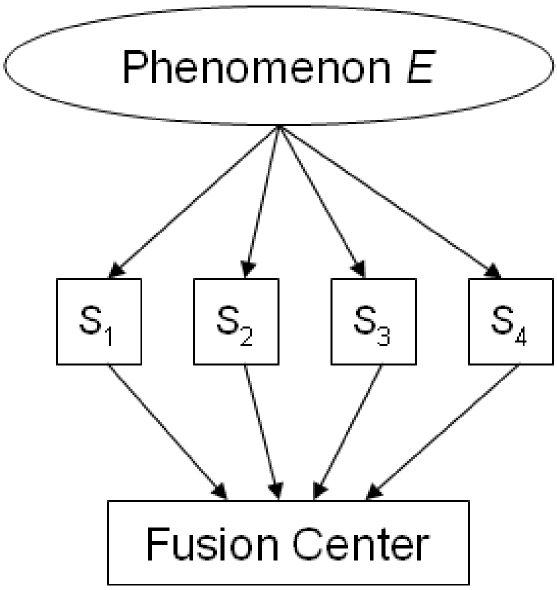


Figure 2.1: Multi-Sensor Detection: Parallel Topology

The centralized scheme fares poorly in terms of scalability, and in certain scenarios, may be impractical too. For example, in case of non-overlapping coverage areas, it is possible that an event is observed by only some of the sensors. In such cases, an optimum scheme would have to be based on decentralized processing of the observations at the sensors. Some of the advantages of distributed signal processing schemes are reduced communication bandwidth requirement, increased reliability, and reduced cost. In addition, distributed system architecture may yield a better response to rapid changes in background scenario. Unlike the central processor in centralized systems, the fusion center of a decentralized system has only partial information as communicated by the sensors. This results in a loss of performance in decentralized systems as compared to centralized systems, which can be reduced by optimal design of algorithms.

Although the distributed scheme is highly scalable and quickly adapts to changes, it requires sending the data to fusion centers for arriving at a decision. This means continued involvement of all the nodes in the detection process. This may lead to certain nodes (especially the fusion centers) to be depleted in resources, mostly energy, sooner than others, compromising the robustness of system. To alleviate this problem requires dynamic election of fusion centers, which results in an additional communication cost.

In general, a distributed sensor network has to address the issues of choice of topology, ability to reconfigure the structure in the case of sensor/link failures, existence of communication between sensors and feedback communication between the fusion center and the sensors, and robustness of signal processing algorithms. Of these issues, the ability of a network to re-configure itself is of paramount importance. Such Self-Organizing networks can continue operating in view of frequent link and node failures virtually unattended. A self-organizing network, for example, shall be able to automatically elect fusion centers, in response to energy consumption in certain nodes, or even due to rapid changes in the sensed phenomena E.

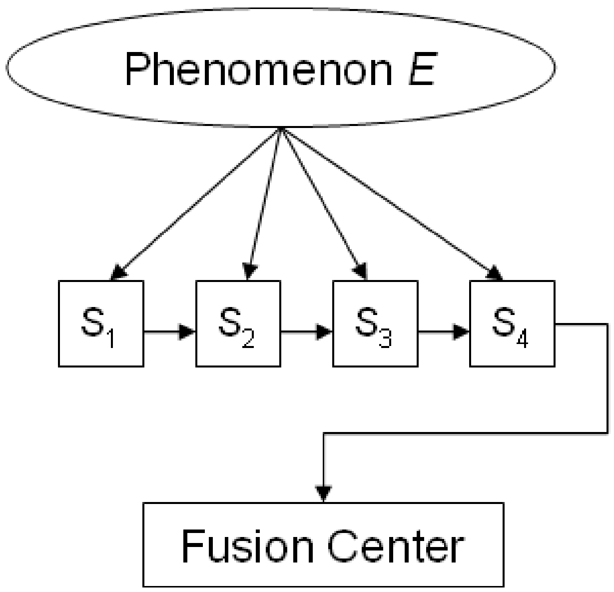


Figure 2.2: Serial Topology

2 .4 Binary Detection

In Binary Detection Problem [27] the observations at each node correspond to either presence (E_1) or absence (E_0) of the phenomena E. This is also known as threshold probing. Here, we assume that the sensors do not communicate with each other and that there is no feedback from the fusion center to any sensor.

Let r_i denote either a single observation that is available at s_i, or, in the case of multiple observations, a sufficient statistic that might exist for the given binary hypothesis testing problem. The i^{th} sensor node, s_i, employs a specific mapping rule and passes the quantized information u_i to the fusion center. Based on the received information the fusion center arrives at a global decision favoring either (E_1) or (E_0). The application of a specific rule by a sensor node may be or may not be done independently of other nodes. A dependent decision may lead to serial or tree topologies which are shown below.

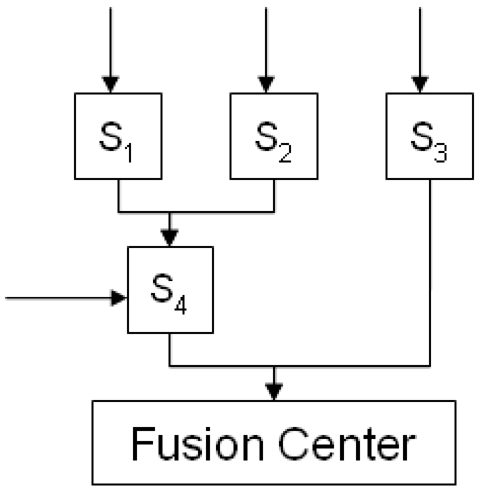


Figure 2.3: Tree Topology

In both serial and tree topologies the information from one sensor is further processed by another sensor and the last sensor receives the quantized information from all sensors to arrive at the global decision. A disadvantage of such an approach vis-à-vis parallel topology is the level of indirection needed to arrive at the global decision. Moreover in case of node failures, the closer a given node is to the fusion center the greater the loss in accuracy. Of all the three topologies considered above, we shall focus on the parallel topology as it provides the best fault tolerance. The choice of decision rules is also a challenging task if the statistical information of the sensed phenomena is not known. Such problems are frequently encountered in spatio-temporal sampling of environmental phenomena.

2.5 Spatio-Temporal Sensing

Sensor Networks are characterized by the dense deployment of sensor nodes that continuously observe physical phenomena. Due to high density in the network topology, sensor observations are highly correlated in the space domain. Furthermore, the nature of the physical phenomena constitutes the temporal correlation between each consecutive observation of a sensor node. These spatial and temporal correlations along with the collaborative nature of the sensor networks bring significant potential advantages for the development of efficient communication protocols which may provide sensing capabilities in space and time that surpass the traditional sensing approaches [1].

Sensor networks are event-based systems that rely on the collective effort of densely deployed sensor nodes which continuously observe physical phenomenon. The main objective of a sensor network is to reliably detect/estimate event features from the collective information provided by sensor nodes. Therefore, the energy and hence processing constraints of small wireless sensor nodes are overcome by this collective sensing notion

which is realized via their networked deployment. While the collaborative nature of sensor nodes brings significant advantages over traditional sensing including greater accuracy, larger coverage area, and extraction of localized features; the spatio-temporal correlation among the sensor observations is another significant and unique characteristic of the sensor networks which can be exploited to drastically enhance the overall network performance.
The characteristics of the correlation in sensor networks can be summarized as follows:

• **Spatial correlation.** Typical sensor network applications require spatially dense sensor deployment in order to achieve satisfactory coverage. As a result, multiple sensors record information about a single event in the sensor field. Due to high density in the network topology, spatially proximal sensor observations are highly correlated with the degree of correlation increasing with decreasing inter-node separation.

• **Temporal correlation.** Some of the applications such as event tracking may require sensor nodes to periodically perform observation and transmission of the sensed event features. The nature of the energy-radiating physical phenomenon constitutes the temporal correlation between each consecutive observation of a sensor node. The degree of correlation between consecutive sensor measurements may vary according to the temporal variation characteristics of the phenomenon.

In addition to the collaborative nature of sensor networks, the existence of above mentioned spatial and temporal correlations bring significant potential advantages for the development of efficient communication protocols well-suited for the sensor networks paradigm. For example, intuitively, due to the spatial correlation, data from spatially separated sensors is more useful to the base station than highly correlated data from nodes in proximity. Therefore, it may not be necessary for every sensor node to transmit its data to the base station; instead, a smaller number of sensor measurements might be adequate to report the event features within a certain reliability level. Similarly, for a certain event tracking application, the measurement reporting frequency, at which the sensor nodes transmit their observations, can be adjusted such that temporal-correlated phenomenon signal is captured at the fusion center within a certain distortion level and with minimum energy-expenditure.

2.6 Energy Efficiency

Sensor nodes comprise of a host of hardware modules that include processor, radio and various sensors [30]. The nodes are powered by batteries which can supply energy only for a limited period of time. Power consumption limits the utility of sensor networks. Replacing batteries every week in building networks is a laborious task and replacing them in a less friendly environment may not be possible at all. A failed node can sometimes, severely hamper the functioning of the sensor network, for example a failed fusion center will have drastic consequences than a failed data collection node. It, therefore, makes sense to judiciously use the available energy in delegating the tasks. Energy efficiency can be achieved by designing low power consuming hardware devices in the first place and developing energy efficient algorithms for further optimizing the power usage.

2. 6.1 Power States

Most of the modern hardware devices, such as processor, memory, radio etc. allow a range of power states wherein a trade off between functionality and power consumption is achieved.

A power saving algorithm's main task is to switch the device to the low power state when it is not in use and switch it back to active state when the usage is required. An additional transition cost is required to switch between the states. The algorithm has to decide when to transition to lower power state to optimize the energy usage. If the transition is done too soon, the node pays high transition costs frequently and if it is done too late, the node spends more time in high power state. The power saving at any node will be affected by how each of the devices i.e. processor, radio, sensors etc. are used with respect to their available power states. Table 2.1 [1] looks at typical current consumption in various components of Mica2 sensor node.

Note that there is a significant reduction in amount of current drawn by various devices in their low power states. This provides a strong case for designing power aware applications and protocols for sensor networks.

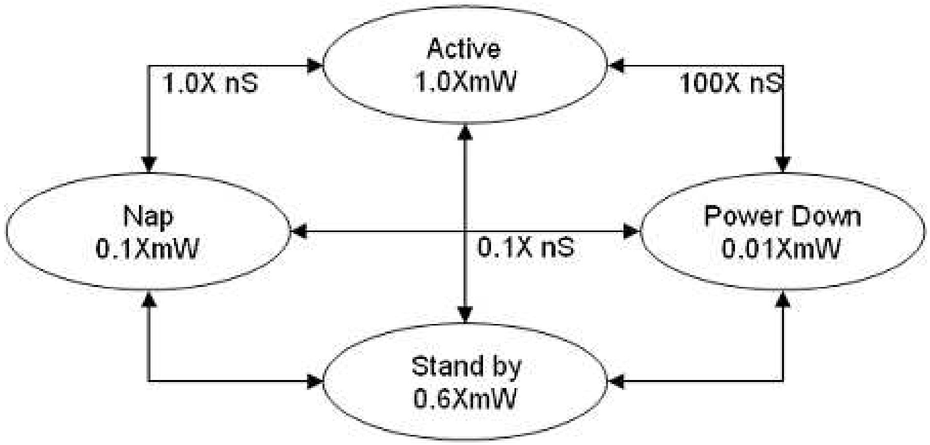


Figure 2.4: Typical Power States, with Running and Transition Costs

2. 6 .2 Radio Usage

As is clear from the above table, Radio communication is the major source of power consumption in a sensor node. At the communication distances typical in sensor networks, listening for information on the radio channel is of a cost similar to transmission of data, so unnecessary radio operation must be reduced to increase node lifetimes. In addition, the energy cost for a node in idle mode is approximately the same as in receive mode. Therefore, protocols that assume receive and idle power are of little consequence, are not efficient for sensor networks. Idle listening, the time spent listening while waiting to receive packets is a significant cost, so, to reduce power consumption in radios, and the radio must be turned off during idle times.

Mode	Current	Mode	Current
CPU		*Radio*	
Active	8.0 mA	Rx	7.0 mA
Idle	3.2 mA	Tx(-20 dBm)	3.7 mA
ADC Noise Reduce	1.0 mA	Tx(-19 dBm)	5.2 mA
Power-down	103 μA	Tx(-15 dBm)	5.4 mA
Power-save	110 μA	Tx(-8 dBm)	6.5 mA
Standby	216 μA	Tx(-5 dBm)	7.1 mA
Extended Standby	223 μA	Tx(0 dBm)	8.5 mA
Internal Oscillator	0.93 mA	Tx(+4 dBm)	11.6 mA
LEDs	2.2 mA	Tx+6 dBm	13.8 mA
Sensor Board	0.7 mA	Tx+8 dBm	17.4 mA
EEPROM ACCESS		Tx+10 dBm	21.5 mA
Read	6.2 mA		
Read Time	*565 μs*		
Write	18.4 mA		
Write Time	*12.9 ms*		

Table 2.1: Power Model for Mica 2

Turning the radio off though may, because packets destined for the node to be missed. Many ad hoc network protocols use forwarding agents, the nodes that receive the packet on behalf of a sleeping node. When the sleeping node wake up the forwarding agent transmits the packets it received for that node.

Radio transmissions costs can be saved by reducing the number of transmissions and/or using a multi-hop strategy. Most protocols reduce the number of transmissions to a sink (i.e. a fusion center or a base station) by aggregating packets. A multi-hop strategy can also save considerable amount of power, since in radio transmissions the power consumed varies exponentially with respect to transmission distance. A draw back of this approach is the additional cost of processing the packets at each of the intermediate nodes.

2.7 Adaptive Sampling

As highlighted in previous sections, one of the roles of a sensor network is to present an accurate picture of a spatio-temporal signal (or phenomena). Since the events are non-uniformly distributed in the environment, individual sensor nodes should alter their sampling rates to match with that of phenomena. Such an approach increases the sampling rate in case the phenomena changes at a higher rate, thus giving a more accurate picture of the signal. By decreasing the sampling rate when signal changes slowly the nodes avoid over sampling which causes increased energy consumption without significantly improving the results. [1] looks at adaptive sampling as an alternative to fixed sampling and presents a control model for varying the sampling rate (Figure 2.5) .

l sensor nodes should alter their sampling rates to match with that of the phenomena.

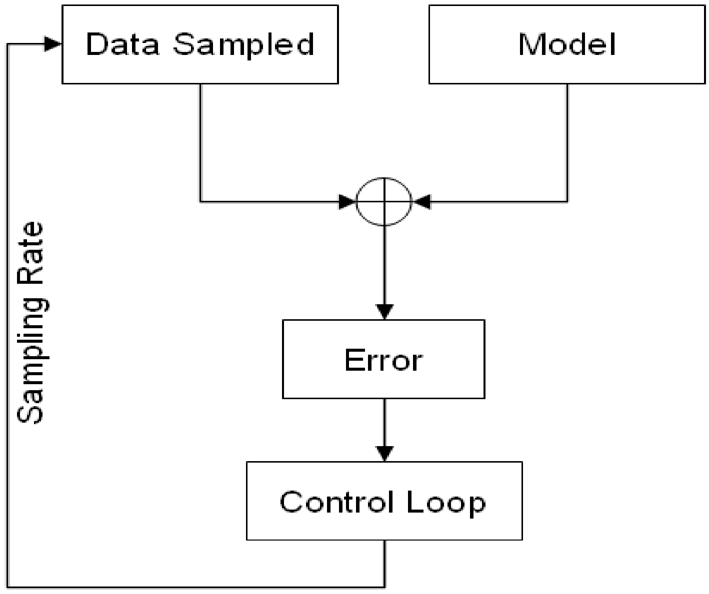


Figure 2.5: Control Loop for Adaptive Sampling within a node

In this chapter we had looked at work done in the field of deployment and relocation of self organizing sensor network. In next chapter we surveys what self organization is, and how can we apply the principles of biological self organization to design sensor networks.

Chapter 3

Self-Organization

3.1 Self-Organization in Sensor Networks

Self-organization means that the system can achieve the necessary organizational structures without requiring external intervention. In case of sensor networks this becomes applicable in multiple ways. Sensor networks are self sustaining systems of nodes that co-ordinate amongst themselves autonomously but, their development is hindered by the constraints of the devices used. Firstly, sensor nodes are (usually) battery based, and it is difficult or unpractical to recharge each node. This also makes energy efficient communication essential. Nodes can die and are also prone to failure. This implies frequent topology change even in case of a static (i.e. nodes are fixed) network. Nodes also have limited computing power, preventing sophisticated network protocols from being run, and limited bandwidth which constraints the amount of communication.

Under these (and similar other) constraints, sensor networks face a host of tasks in providing an end to end application. These include:

•**Dynamic networking**: As discussed before, the nodes are randomly deployed densely and rapidly in inaccessible terrains, for example in their thousands from an aircraft. As a result, they need to set up a network dynamically, in an ad-hoc manner that would be flexible enough to respond to frequent topological changes. These changes are the result of potential sensor failures, node mobility or additions that should prompt the network to re-organize itself, to deal with the respective loss or gain of a system resource.

•**Self-calibration**: The nodes need to calibrate themselves automatically and adapt to the changes in their environment independently. Manual operation of the network could be made difficult by the weather or location, so unattended independent operation is imperative. The devices need to divide among themselves, the task of monitoring, while adapting to the resources at their disposal.

•**Peer to peer communication**: The nodes need to be able to talk among themselves, to develop a multi-dimensional view of the sensing environment. A centralized approach would not provide the vast scalability expected from a sensor network, thus distributed and localized algorithm is needed, where information is passed between sensors in the same vicinity. The data can then be compressed and aggregated to give an accurate global representation.

• **Mobility**: Since sensor nodes have limited sensing range, they need to be located at the regions where spatial activity is of particular interest in order to optimally fulfill their purpose. Such mobility or self-organization is dependent on local information, set of rules, internode communication, and coordination among nodes to achieve the overall system goals.

Implementing sensor networks involve methods that allow the nodes to make decisions based on their local environment and their own individual state that would result in the global purpose of the network being fulfilled. With this respect we define self-organization as

> " . . . a process in which pattern at the global level emerges solely from numerous interactions among the lower-level components of the system. Moreover, the rules specifying interactions among the system's components are executed using only local information without reference to the global pattern. "

The pattern, as such, is an emergent property of the system, rather than a property imposed on the system by an external ordering influence [20]. In this regard, a distributed system is not a self-organizing system, in true sense. This is because the local processing at nodes is done keeping in view the overall goal of multi-sensor detection [28] i.e. a node sensing some phenomena is aware and hence needs to communicate with another node or a fusion centre in order for the system to arrive at a global decision. Note that, a distributed system arrives at a global decision (or pattern) where as in a self-organizing system the pattern emerges. Also, the nodes that interact with each other are, supposedly, not aware of the holistic view of the problem, as they are generally concerned about processing the local information as efficiently as possible and interacting with each other mostly without any restrictions (imposed order) .

This definition of self-organization can be listed into a set of features:

- The system is composed of units that individually respond to local information, based on predefined rules.
- These units collaborate and communicate to divide the task among them.
- The system, as a whole, achieves the goals more efficiently.

The rules that define local information processing decisions and inter-node communication are also very simple and the emergent pattern itself cannot be merely expressed as a sum total of local interactions. The emergent pattern can be viewed as the overall application goal, which in our case is spatio-temporal sensing, energy efficiency and distributed actuation. To fully appreciate the role self-organization can play in achieving our goal we look at self-organization in biological and artificial systems and discuss how knowledge from these realms can be reused in sensor networks.

To summarize, implementing sensor networks involves methods that allow the nodes to cooperatively make decisions based on their local environment, their own individual state and predefined rules, and their mutual communication that would result in the global purpose of the network being fulfilled. Thus self organization among sensor nodes can be listed into a set of features.

- The system is composed of units that individually respond to local information, based on predefined rules.
- These units collaborate and communicate to divide the task among themselves.
- The system, as a whole, achieves the goals more efficiently.

3.2 Biological Self-Organization

The concept of self-organization in biological systems can be explained through counter examples. A marching band forming alphabets on the field provides one such example. Here the band's members are guided in their behavior by a set of externally imposed instructions for the movements of each band member that specify precisely the final configuration of the whole band. A particular member of the band may know that the instructions are to march to the center line and then turn left and move in straight line. In such a case the member may blindly follow the instructions and often ignore the local information such as his position relative to other members, this is clearly not a case of self-organization [20].

Ant colonies, on the other hand, are an example that show functionality and adaptation on a global scale that result from locally-applied underlying rules and amplified collective behavior. The large-scale spatial patterns found in the clustering behavior of ants suggest that the ants carry out very simple steps without the need for central manager that does more than the rest of the group. There are many other examples of self - organization, emergence and complexity in biology that involve lower-level entities making local decisions and unconsciously affecting the global pattern.

3.2 .1 Schooling of Fish

It is quite an amazing sight to watch a school of fish moving in parallel in same direction. When the school changes its direction, all its members rapidly respond, moving cohesively as if they were a single organism. This behavior suggests that a school possesses special group level properties. Such coordinated movements help the prey in avoiding their predators, and the predators in surrounding their prey. One of the evasive movements, a prey performs, is the flash or rapid expansion and another is fountain expansion.

A likely explanation of schooling in fish can be explained using a simple, self organizing model. This model incorporates the known behaviors and sensory capabilities of a fish as it moves in the school, making approximate, moment-to- moment responses based on the local information. This model can be enumerated by following set of assumptions:

1. Each fish in the school follows the same set of rules as other fish, thus there are no leaders or followers.

2. To decide where to move, each fish uses a weighted average of the position and orientation of its neighbors, more so, the fish responds to neighbors in a probabilistic manner.

3. In response to a neighbor, the fish exhibits one of the four behaviors.

(a) Repulsion: To avoid collision with other fish.

(b) Attraction: To swim towards the school.

(c) Parallel Orientation: To orient in same the direction as one of its neighbor.

(d) Searching: To randomly turn towards the school or a neighbor, if the fish gets isolated. It should be noted that the fountain or flash expansion (Figure 3.1) effects demonstrated by the school are not at all obvious from the above rules. This clearly emphasizes the fact that a pattern at global level emerges from local interactions. We shall now see another example of self-organization in nature, where organisms collaborate to carry out foraging.

3.2.2 Nectar Source Selection by Honey Bees

Honey Bees have been found to focus their foraging activities around more profitable nectar sources. A collaborative effort of several bees maintains a steady flow of food-supply in the bee hive. It has been verified that the foraging activities emergent at the colony level occur without any central leadership or a global rule, per se. The overall nectar source selection process can be understood with respect to the behavior of individual bees [20].

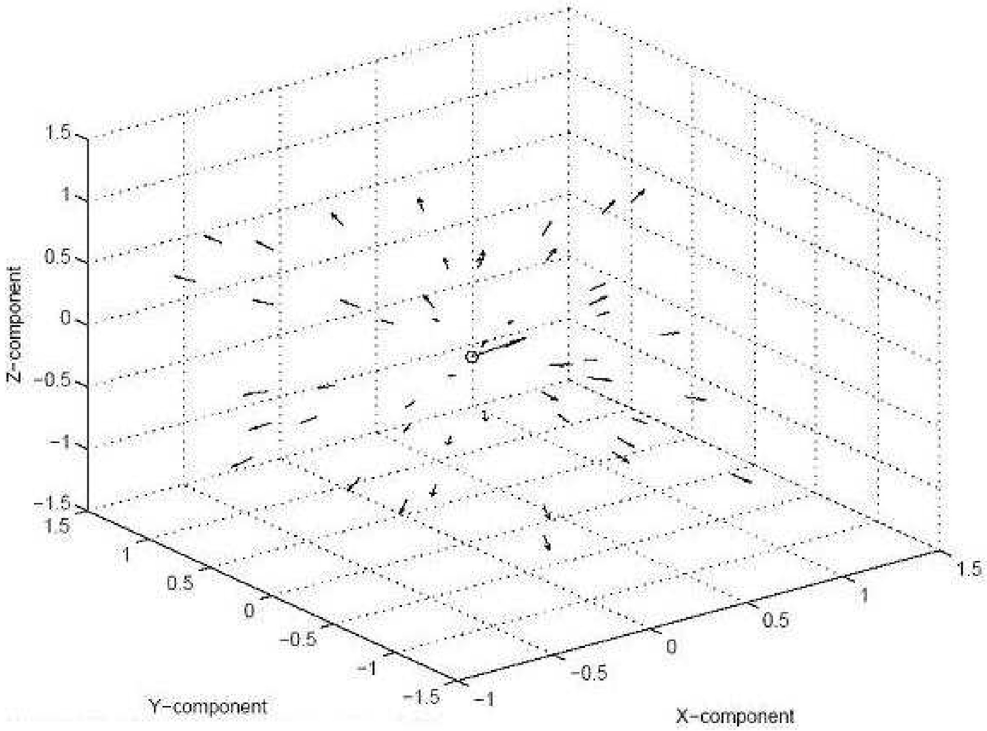


Figure 3.1: Flash expansion in fish

(The fish (arrows) move rapidly away from the predator (circle) as it approaches the school, leaving the predator with a sea void of any fish.).

3.2.3 Employed and Unemployed Foragers

Employed foragers are those bees that are currently engaged in bringing nectar to the hive and conveying the location of a food source, while unemployed foragers receive information about food sources and then search for one of the advertised work sites. Every time an employed forager flies into the hive, it deposits the nectar it brought from the flowers and performs a waggle dance to share the knowledge (quality, distance and route) about the food source with other bees. Only the information about high quality sources is shared. An employed bee may also give up foraging activity if the nectar source is depleted beyond exploitation.

Unemployed foragers observing the dance have an opportunity to become well informed about the various food sources exploited by their colony. Based on the observed dance, the unemployed forager decides whether to get involved in foraging for that particular source or wait for other dances. It has been deduced that the decision to forage is based on the information received during a small time frame and as such does not involve and accumulation of knowledge over successive dances by different bees. An unemployed forager actually observes a randomly chosen dancer and samples it before deciding to exit the hive. Thus we can say that the overall foraging activity of the entire hive is carried out using only local information and without any reference to a global template.

3.3 Self-configuring Wireless Network

Wireless or sensor networks employ a self-organizing scheme for coordinating the channel access, since conventional approach such as TDMA fare poorly when not all the nodes are within the range of each other. To alleviate this, nodes only collaborate with their neighbors to adjust their transmission schedules. Highly sophisticated schemes have already been developed for sensor networks.

Our work is also an example of a self-organizing wireless sensor network which is partially inspired from the foraging in bees and has been theoretically described using a Digital Hormone Model. We enumerate the characteristics of a self-organizing system and try to draw some analogies that can be applied to spatiotemporal sensing and form the basis for the theoretical framework.

3.4 Characteristics of Self-Organizing Systems

Feedback Mechanism : A feedback mechanism forms the basic mode of interaction between components in a self-organizing system. Negative feedback tends to resist changes in the organism. For example, in human body a rise in blood glucose level triggers insulin release this causes a decrease in glucose level. Similarly, in the example above, when a fish gets too close to a neighbor, it receives a negative feedback and moves away from it to avoid over crowding. In contrast to this, positive feedback promotes changes in system. The cascading effect of positive feedback takes an initial change in a system and amplifies that change in the same direction as the initial deviation. A common example that establishes positive feedback is the fact that each generation of human has more than reproduced itself. The attraction behavior shown by fish is also an example of positive feedback. Together, the negative and positive feedbacks (in a fish school) keep the size of school from becoming too large or small.

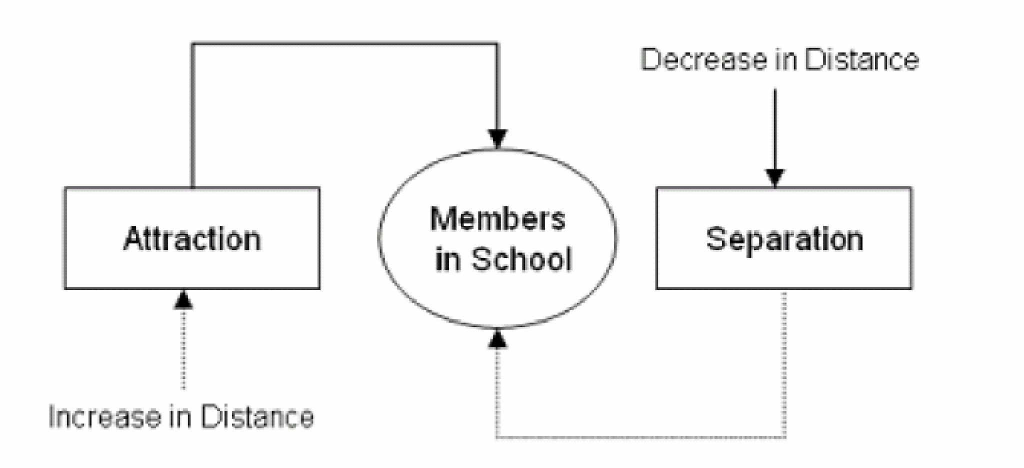


Figure 3.2 : Feedback Mechanism in a School of Fish

These feedback mechanisms come into play when an individual acquires and acts on local information. In essence, positive feedback creates patterns, negative feedback keeps it under control, thus converting environmental randomness to organizational structure.

- **Dynamism:** The multiplicity of interactions that characterize self-organizing systems, emphasize that such systems are dynamic and require continual interactions among individual components to produce and maintain structure. This dynamic process gives rise to properties that emerge from the ongoing pattern formation. All the examples; schooling of fish, foraging in bees, swarm intelligence clearly exhibit this trait and, in fact, make it imperative for a self-organizing system to be dynamic.

- **Emergent Properties:** Emergence refers to a process by which a system of interacting components acquires qualitatively new properties that cannot be understood as a sum total of their individual contributions. Most of such properties emerge spontaneously from local interactions as exemplified by

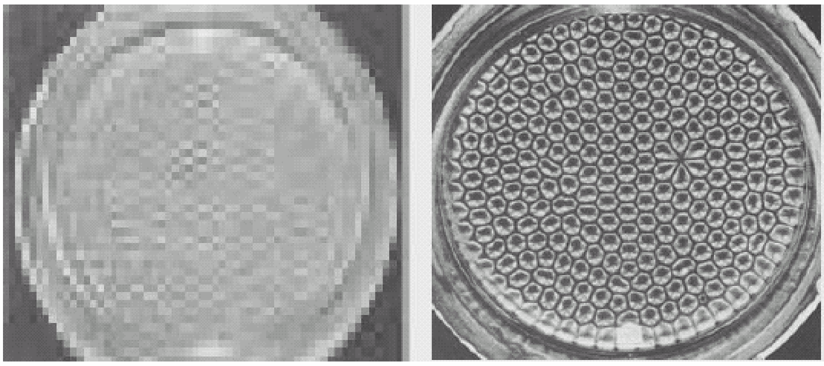


Figure 3.3: Emergent pattern in B´enard Convection
Source: http://www.catea.org/grade/mecheng/images/BenardConvection-MVanDyke.gif

B´enard convection cells (Figure 3.3). Here an initially homogenous layer of fluid becomes organized into a regular array of hexagonal cells of moving fluid. More so, this pattern does not develop gradually, but rather appears suddenly at a moment determined by the amount of heat applied to the bottom of the fluid layer.

- **Parameter Tuning:** Such a sudden change occurs as one of the inputs or parameters are varied even slightly. By making small changes in such parameters one can induce large changes in the state of the system. In B´enard convection, the tunable parameter is the amount of heat applied to the lower surface of the dish containing the fluid. Such a quantitative change in parameters that lead to a qualitative change in behavior are called bifurcations. These bifurcations make a self-organizing system adaptive and flexible towards changing environment and changing goals of the system, and are also the key to understanding the complex behavior that emerges from simple rules.

3.5 Analogy from Honey Bee Foraging

The foraging activity of bees can be divided into two parts; discovery of nectar sources and sharing the information with unemployed bees. It is clear from the previous section that at any time, a bee is either foraging or observing the waggle dance and the decision to forage or observe is taken by that bee alone. Such an approach prevents wastage of hive resources, as not all bees go out to forage. The task allocation among bees is also based on local information only.

With respect to the sensor networks one can consider a sensor node akin to a bee. The process of searching nectar sources by employed foragers is analogous to sampling of environment. Although in normal sense the sensor nodes monitor the environment around them i.e. within their sensor range but in this respect whether the bee goes to the event (the nectar source) or the event occurs at the sensor node are relatively same. If we consider a mobile sensor node that moves around, to sample a region, then a closer parallel can be drawn.

Multiple bees carrying out nectar source searching can be regarded as sampling in space, where as the searching activity of a single bee is equivalent to sampling in time. The analogies from bee foraging can be summarized as:

- Self-Organization through task allocation among employed and unemployed foragers.
- Spatio-temporal sensing through nectar source searching.
- Energy Efficiency achieved by having only part of the bees involved in sampling.

We have seen that the simplicity of a self-organizing system is its greatest strength. Simple changes in rules can cause drastic changes in overall system behavior. Much of the natural systems are also self-organizing and we intend to reuse some of this knowledge in building a artificial self-organizing system.

Chapter 4

Sensor Deployment

Sensor networks are self sustaining systems of nodes that co-ordinate amongst themselves autonomously but, their development is hindered by the constraints of the devices used. Firstly, they are power constrained which makes device failure inevitable and energy efficient communication essential. They also have limited computing power, preventing sophisticated network protocols from being run, and limited bandwidth which constraints the amount of communication. Human intervention to keep the network up and running, in such conditions, is at the least a tedious job and mostly infeasible. It is for this reason that there is a continued effort to make sensor networks as autonomous as possible.

Self-organizing sensor networks may be built from sensor nodes that may spontaneously create impromptu network, assemble the network themselves, dynamically adapt to device failure and degradation, manage movement of sensor nodes, and react to changes in task and network requirements. Reconfigurable smart sensor nodes enable sensor devices to be self-aware, self-reconfigurable and autonomous. The main benefits of these features are:

1. Support tactical and surveillance applications using reconfigurable sensor network nodes that are capable of forming impromptu network, being deployed incrementally, and assembling themselves without central administration.

2. Provide capabilities for sensor networks to adapt dynamically to device failure and degradation and changes in task and network requirements.

3. Integrate various application-specific network and system services provided by mixed types of sensor nodes and embedded defense applications.

Modeling of SOSN can be addressed from various aspects, such as sensing coverage, node placement, connectivity, energy consumption, etc. The present work aims at modeling the sensor network from the coverage point of view. We divide the problem of sensor coverage into two phases one is the deployment phase followed by the relocation phase. The geographical space is divided into regular hexagons; we propose a distributed deployment algorithm for the deployment phase which ensures the uniform distribution of the sensor nodes through out the sensing field, i.e, each cell in the region must be occupied by at least one sensor node. In the relocation phase the redundant idle sensors are moved to the target location for tracking the multiple events which improves the coverage of the sensor network. We propose a frame-work for the sensor relocation which is explained in the next chapter in detail.

4.1 Deployment Phase

Sensor deployment has received considerable attention recently. When the environment is unknown or hostile such as remote harsh fields, disaster areas and toxic urban regions, sensor deployment cannot be performed manually. To scatter sensors by aircraft is one possible solution. However, using this technique, the actual landing position cannot be controlled due to the existence of wind and obstacles such as trees and buildings. Consequently, the coverage may be inferior to the application requirements no matter how many sensors are dropped. Moreover, in many cases, such as during in-building toxic-leaks detection, chemical sensors must be placed inside a building from the entrance of the building. In such cases, it is necessary to make use of mobile sensors, which can move to the correct places to provide the required coverage.

4.2 Cellular Based Management Model

In this project, we used a Cellular-Based management model for mobile Ad-hoc Sensor Network [2]. In the Cellular-Based Management, the mobile Ad-hoc WSN is geographically partitioned into several disjoint and equal-sized cellular regions as shown in the Figure 4.1 Each cell is then assigned a unique Cellular-ID, as shown in Figure 4.1. Head of one cell can directly communicate with heads of the neighboring cells.

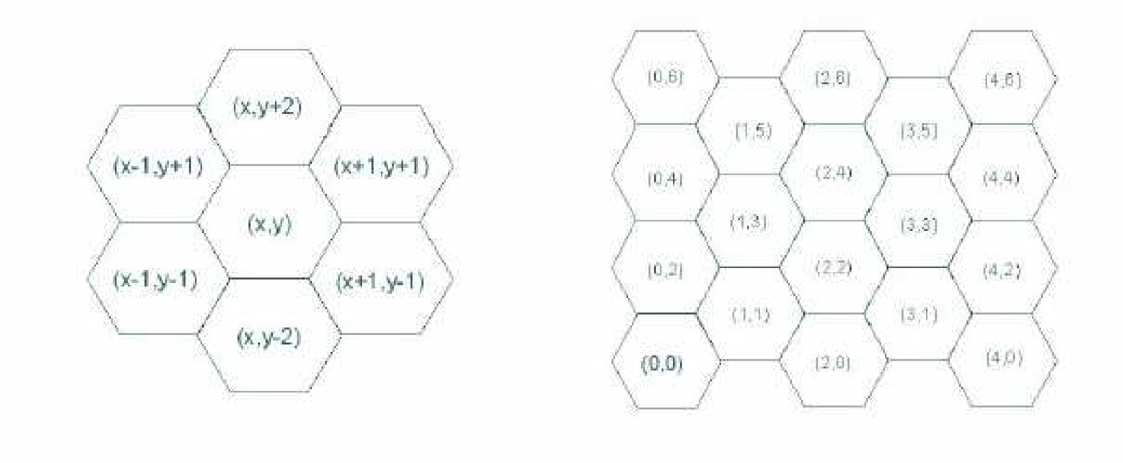


Figure 4.1: Cellular-ID of a partitioned Sensor Network

Let the signal radius of each host be Rc. Under the proposed management model, mobile sensor host can construct a stable communication path with fewer flooding messages and smaller number of hop count. To ensure that each cell head can directly communicate with neighboring cell heads, in the worst case, the maximal distance of two neighboring cells by the value of R_c, is given from the formula:

$$\mathbf{R_c = \sqrt{13}S}$$

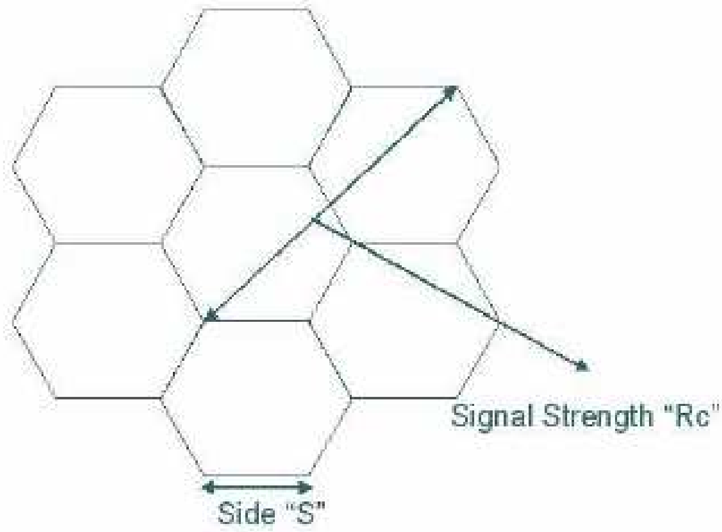


Figure 4.2 : Range for Signal Strength

As given in[2] the proposed algorithm needs area of sensing field as reference for localization of sensing nodes. This can be given as the range of Latitude and longitude in case of an open field or as coordinates in case of a closed field. The granularity of the hexagonal grid is decided from the communication range of sensor being used. The coordinates of each dropped node can be taken from satellite (GPS) or using 3Beacons placed locally as shown in fig(4.3).Once the node gets it's coordinates it can calculate the corresponding cell_id (C_x, C_y) of the hexagonal tessellation which is being logically conceived within the sensing field as shown in fig(4.1). The sensor nodes in each cell will then select its master / head node and all other nodes will become slave. Then the master node present in each cell can directly communicate with master node of neighboring cell. Each cell will be broadcasting its coordinates, its cell_id, the sensor reading and the no. of slaves present in the cell. Let the communication range of each host be R_c.

The size of hexagon is calculated from the maximum communication range of a sensor which will ensure direct communication between neighboring cell heads even in the worst case.

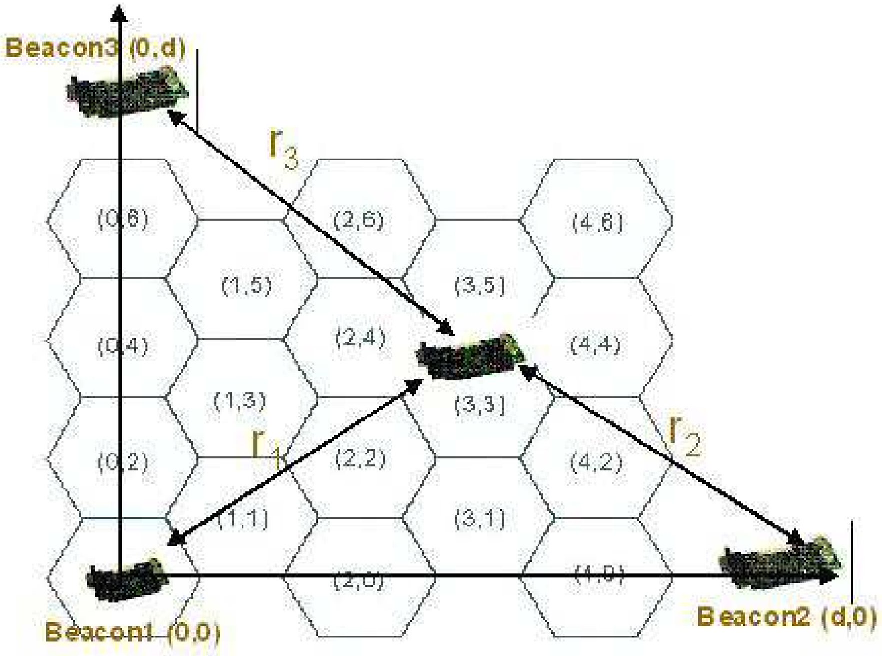


Figure 4.3: Localization System

4.2.1 Advantages of Cellular Model

• If we consider the sensing and communication pattern as omni directional then, with the hexagonal representation we can cover the given field with minimum number of nodes with minimal overlapping area.

• Mobile sensor host can construct a stable communication path with fewer flooding messages.

• Smaller number of hop count to ensure that each cell head can directly communicate with neighboring cell heads.

4.3 Deployment Algorithm

In this section, we describe a distributed algorithm for sensor deployment, considering that the space is divided in to several disjoint and equal-sized hexagonal cellular regions. Each node first elects itself as a cell head with a pre-defined probability P_{head}. If one node elects itself as a cell head, it broadcasts its location to its neighbors. Otherwise, the node listens to the messages from cell head and the node will send its location to its corresponding cell head. If no head is elected with in a cell then all the nodes will be in the normal state and will respond to the message transmitted by the head node while learning step of the deployment process which means that the cell is occupied. In this each cell head is capable of communicating with neighboring cell heads/normal nodes. If once a head is elected, all the nodes in that cell will become slaves to that head node. Once the head is elected then, it will execute the deployment algorithm given in the following sections. The figure 4.4 represents the state transition diagram of a node during the deployment phase.

4.3.1 Assumptions

The key assumptions are:

a. Each node has the ability to determine its positional co-ordinates with respect to a reference axis.

b. Each node can autonomously navigate from its current location to a commanded goal location.

c. All nodes can communicate within a short range.

d. The sensor network contains large number of nodes.

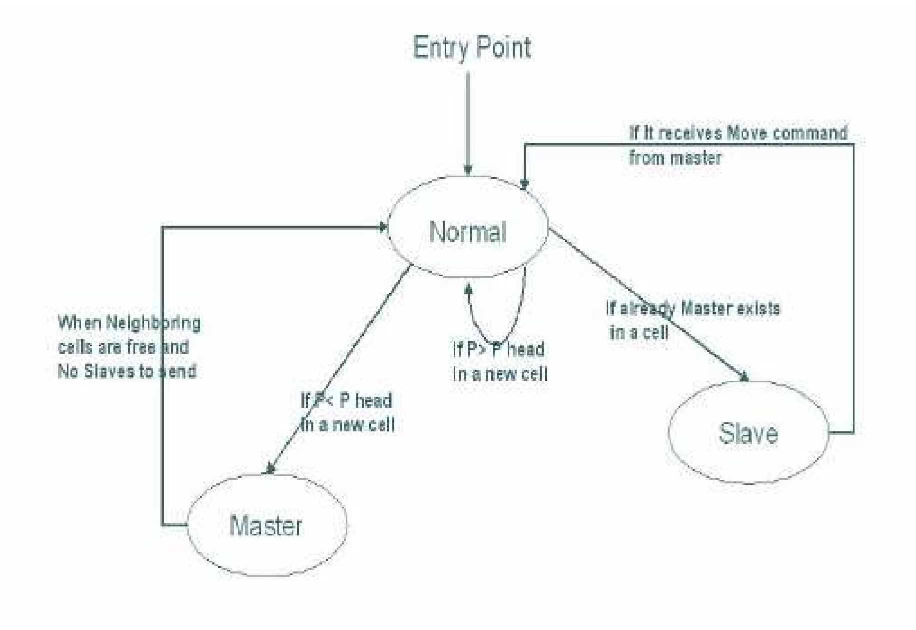


Figure 4.4: Node State Transition Diagram In Deployment Phase

4.3.2 Algorithm

H_i^k represents the head node of a cell k at i th iteration and P is the set of current node locations. A_i^k is the adjacent cell information of the head cell H_i^k and V_i^k are the victim nodes that are targeted to the empty neighoring cells of the cell C_k by the head node H_i^k

0: Partition the sensing field into small sub cells C_k having a regular pattern (hexagonal);

1: i=0;

2: **While** termination conditions are not satisfied do

3: Select a cell head H_i^k from the nodes which are moved to a cell C_k;

4: Each node *n* within a cell sends position P[*n*] to the cell head;

5: Each cell head H_i^k learn the neighboring cells information $\forall C_j \in C_{nbrs}$ and constructs the adjacency list A_i^k;

6: Each cell head H_i^k selects the victim nodes V_i^k that are to be sent to the neighboring cells C_j € C_{nbrs};

7: Assign each victim node a new position;

8: Notify the adjacent cells with positions of the victim nodes;

9: All the Victim nodes will move to the new cells;

10: i = i + 1 ;

11: **end**

The distributed algorithm is executed by the master nodes, it will first learn the neighboring cells information i.e., whether the cells are occupied/ unoccupied, there might be some cells which doesn't contain head cell but still occupied with nodes, for this case the normal nodes present in that cell will respond to the messages transmitted by their neighboring cell heads. After learning the neighboring cell information the head cell will look for the slave nodes that can be sent to the unoccupied cells, the victim nodes are selected based on the Euclidean distance between the center of the empty cell to the slave nodes positions, the node which is closest to the empty cell will be chosen as a victim node. If no slaves are present, then the head node itself will move to the closest empty cell which creates a void cell in that region which draw the slave/head nodes from the neighboring cell. If none of the neighboring cells have nodes, then the node will oscillates but these oscillations will not last longer as the void cells are slowly occupied by the nodes as number of iterations increases.

4.4 Analysis

As a part of analysis we would like to show that the proposed deployment algorithm will converge and will attain a stable state after certain number of iterations. For this we define a term called occupancy and then show that this will increase or will remain constant as time progresses by taking some constraints on the properties of the sensor node. Occupancy is defined as the total number of cell's occupied at a given iteration. First we will deal with case in which two nodes are targeted to a single void cell, which leads to the decrease in the occupancy and later with case where there will be always increase in the occupancy for every k iterations on an average.

'N' be the total number of cells in the given area whose side is given by 'S' O_i be the occupancy at a iteration i with a given probability of electing head be P_{head}

$$O_i = \sum_{k=1}^{N} C_i^k \quad (4.1)$$

C^k_i = 1 If the cell is occupied

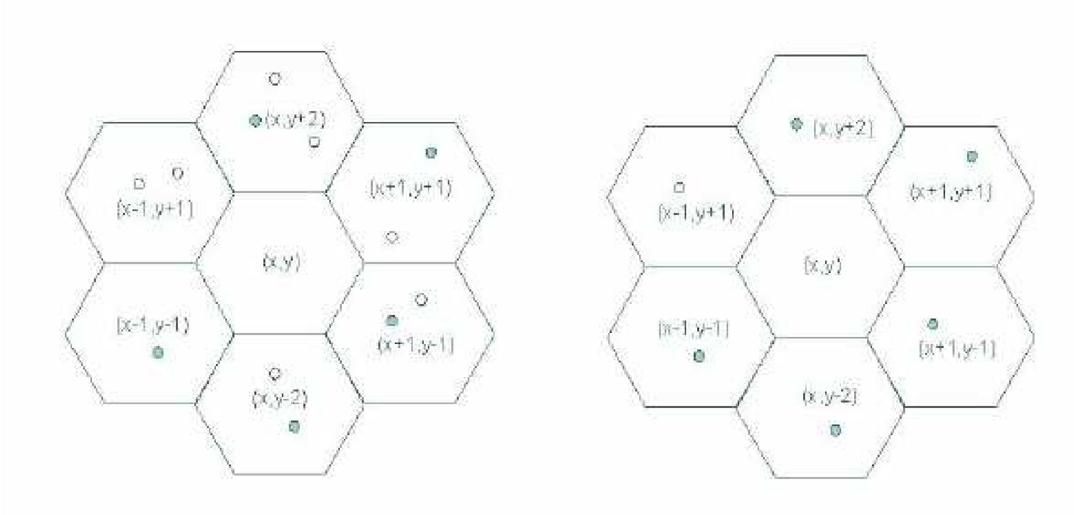
0 otherwise

O_0 is the initial occupancy

We would like to show that the following inequality satisfies at any time instant

$$O_i \leq O_j \qquad \forall i \leq j \tag{4.2}$$

Case a : The two important cell configuration's at a iteration i is given by the figure 4.5 in which all the neighboring cells of the empty cell is occupied by at least one head/normal node. At a iteration i there is a possibility that more than one head node will send victim to the empty cell. For case 1 shown in figure 4.5(a) there isn't any problem as if the occupancy increases by 1 for the iteration i+1 but for the case 2 shown in the figure 4.5(b) the problem of merging comes in to picture as if there is a possibility that two head nodes will move to a single empty cell which decreases the occupancy at a iteration i+1. But this state can be avoided by taking the communication range of the sensor node as shown in the figure 4.6. When this condition is imposed in the network, during step 8 of the algorithm the head nodes will send the notify message which will be processed by the head node which is in the scheduling phase will come out of that phase.



(a) case 1 (b) case 2

Figure 4.5: possible cell configurations at iteration *i*

Even if both the heads try to move the same time there will be only one head that will be sending the notify message first as the communication channel is same, hence avoiding the merging of two nodes. The figure 4.7 shows the nodes configuration at a iteration *i*+1. Hence, critical communication range for a given hexagon side 'S' is given by

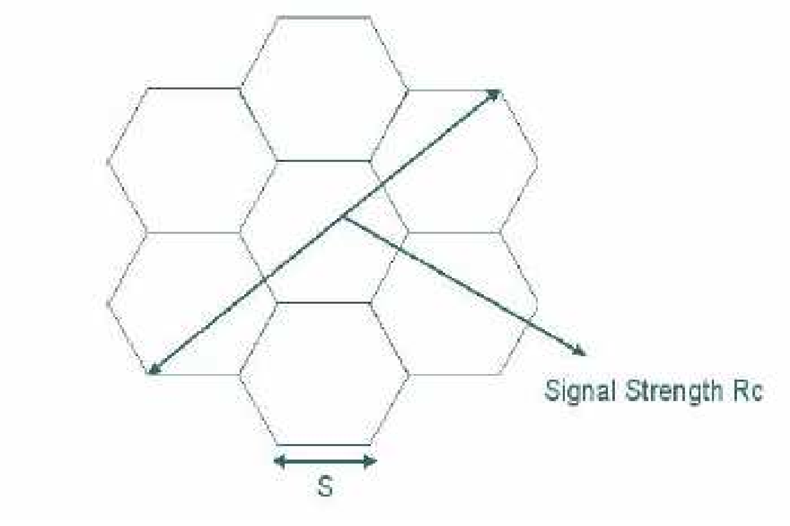
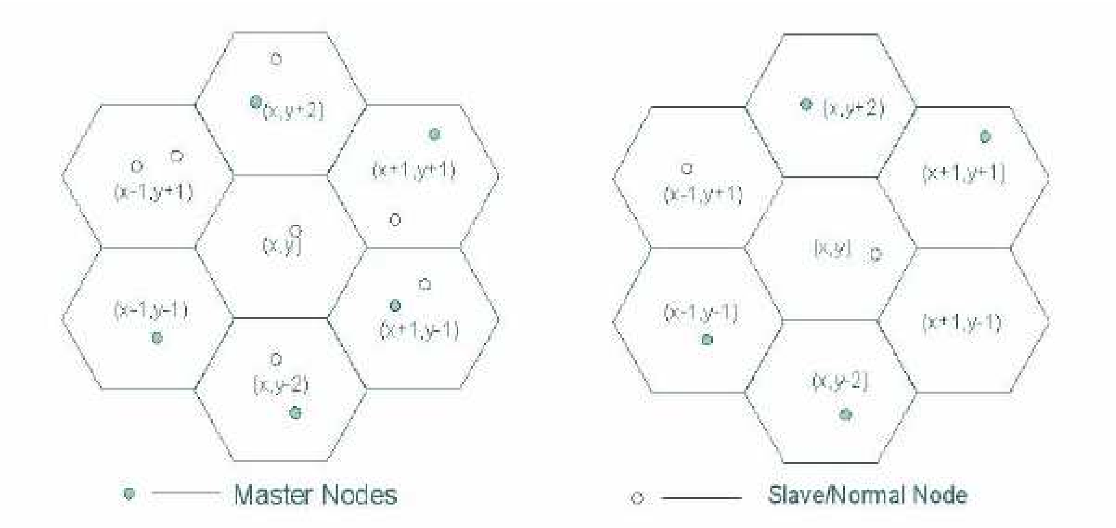


Figure 4.6: Critical Signal Strength for Convergence

$$R_{Critical} = \sqrt{28}S \qquad (4.3)$$



(a) case 1 (b) case 2

Figure 4.7: possible cell configurations at iteration *i*+1

Hence we can say that

$$O_i \not> O_j \qquad \forall i < j \qquad (4.4)$$

Case b: In this we would like to show that there will be always increase in the occupancy by at least one for every k iterations on an average, if there are unoccupied cells in the network. Let us consider a case that at a iteration i, all the neighboring cells of the void cell are having nodes in normal state, since the normal nodes doesn't executes the step 5 of the deployment algorithm the void cell will remain unoccupied until a head node is elected in the neighboring cells. So even the iterations are going on there will not be any increment in the occupancy which should not be the case.

Given probability of the node being head is $P_{head.}$ In the cell the probability of the node to in normal state is P_{normal}.

Hence, we have $$P_{head} + P_{normal} = 1 \quad (4.5)$$

$$P_{normal} = 1 - P_{head} \quad (4.6)$$

After k iterations the probability of the node being the head atleast once is given by

$$P_{(atleast\ once\ head)} = 1 - (P_{normal})k \quad (4.7)$$

$$P_{(atleast\ once\ head)} = 1 - (1 - P_{head})k \quad (4.8)$$

Hence, for large value of k the probability will goes to 1 and hence at least one node will become head, then that head node will go through the steps 5-9 which will fill the empty cell hence increasing the occupancy. Hence from case a and case b, we can say that

$$O_i \leq O_j \qquad \forall i \leq j \quad (4.9)$$

Larger will be the probability of the head election smaller will be the iterations taken for the convergence of the algorithm.

Finally we would like to say that algorithm proposed is unique in the following ways:

1. Our approach makes no assumptions on the initial distribution of the sensor nodes and the deploying environment.
2. Our algorithm is distributed and it can be applied to a system consisting of large number of nodes.
3. Doesn't need high computational resources.

We addressed one phase of the coverage problem, say deployment in this chapter. Next chapter deals with event detection model and the relocation of the redundant sensors.

Chapter 5

Event Coverage and Sensor Relocation

Modeling of SOSN can be addressed from various aspects, such as sensing coverage, node placement, connectivity, energy consumption, etc. The present work aims at modeling the sensor network maximizing event coverage. In the relocation phase the redundant idle sensors are moved to the target location for tracking the events which improves the QoS of the sensor network. We propose a frame-work for the sensor relocation which is explained in this chapter in detail. In many potential working environments, such as remote harsh fields or disaster areas sensor deployment cannot be performed manually or precisely. In addition, once deployed, sensor nodes may fail or respond to new events, requiring nodes to be moved to overcome the coverage hole created. In these scenarios, it is necessary to make use of mobile sensors, which can move to provide the required coverage. Reconfigurable smart sensor nodes enable sensor devices to be self aware, self-reconfigurable and autonomous. In this project we address the problem of sensor relocation, i.e., moving previously deployed sensors to respond to an occurring event that requires that a sensor be moved to its location with out creating any holes in the region of interest.

5.1 Gradient Based Event Tracking

In the first phase, we considered an algorithm developed in [1] to track a light source with single node. Light source can be moving or stationary. It will present the time varying nature of the events in space. In order to move node in the direction of light source, we will take the history of the events in the space. In [1] three consecutive values of the light in past time and space have been taken. To take the decision in which direction it has to move so that to reach the light source in minimum number of iterations.

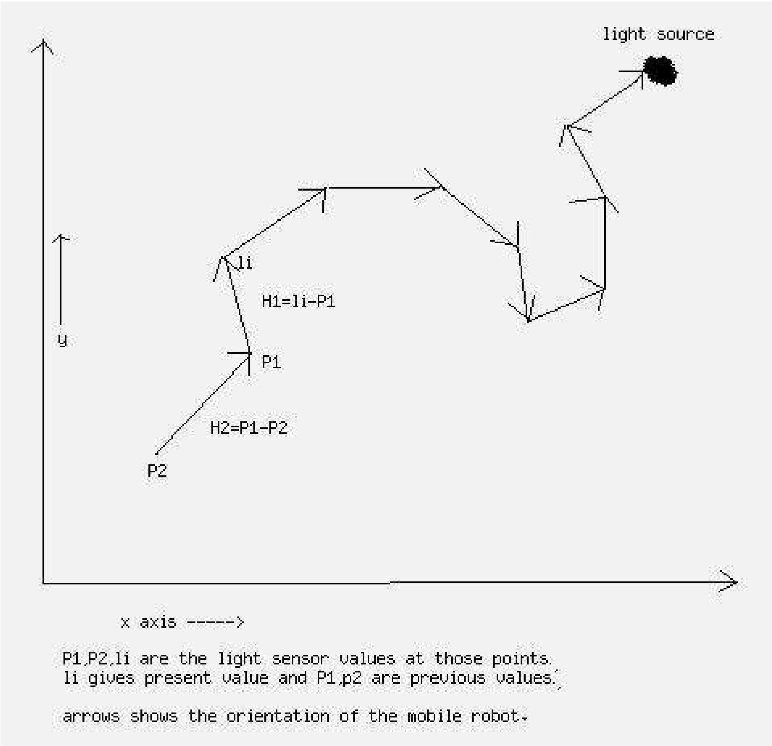


Figure 5.1: Position of Bee Bot while tracking light source

5.1.1 Algorithm:

In the implemented algorithm of light tracking, it will store the three previous sensor values. Decision for the further movement of the mobile node is decided by the difference between two successive sensor values taken (H1,H2) and the previous orientation of the mobile node motion_dir. Motion_dir gives the direction in which mobile robot has to be moved or to take the value which mobile robot is taken.

Light Tracking

Algorithm for Light Signal Tracking is given below.

1: Initialise H1=0,P1=0,H2=0,P2=0, motion_dir=1(1=FWD LEFT,0=FWD RIGHT);

2: Move FWD_ LEFT and FWD_ RIGHT and take the two sensor values into P2 and P1 at the beginning of movement respectively;

3: **While** termination conditions are not satisfied **do**

4: sense the present value into l;

5: calculate the difference between the successive points in which mobile node moved as given below

6: H2=l-P1; H1=P1-P2;

7: if H1 < 0 and H2 < 0

Then move the mobile in the direction given by motion_dir for three successive intervals;

8: else if H1 < 0 or H1 < H2

Toggle motion_dir and move the node in the direction of Motion_dir

9: else

Move the mobile node in direction motion dir

10: **end.**

5.2 Event Detection Model

This event detection model is used to detect the events in the network, we used the DHM developed in [1] for detecting the events in the network and take necessary actions when an event is detected.

5.2.1 DHM

A Digital Hormone Model (DHM) [1] encapsulates dynamic networks of mobile nodes or robots that use hormone like messages to communicate, collaborate, and achieve their goals. The hormones received by a node influence, but do not determine, their local behaviors, as it is the function of both the type of hormone and local state of the node. Thus, different nodes react differently to the hormones despite the fact that all are running the same protocol. The DHM consists of three components; A Dynamic Network of Mobile Nodes, A Probabilistic Function for node behavior and a set of rules and equations for hormone (message) reaction and propagation.

Some of the important rules used in our project from the DHM model are explained below. In this model the messages in the network are treated as hormones which change the state of the mobile sensor node. The rule B0 is used to detect the event, rule B2 transmits the message (Hormone) by a node which detects the event and rule B5 is adaptive sampling which is a power saving rule. We assume that sensor nodes with isotropic radial sensors of range R_s which is less than the side of a cell and the quality of sensing is constant within R_s and is zero outside the sensing range, i.e. it follows a binary model.

Terminology Used

* The variable S can have three states: Probe, Listen or Move

* $r_i(t_x)$: sensor reading from sensor i of node at sampling instance t_x
* H denotes the hormone that is transmitted from a probing node to a listening node. Each H is contains the (C_x, C_y), (x,y) coordinates of the transmitting node as well as the sensor reading $r_i(t_x)$.

• B0 (State Change rule)

If S = Probe AND $r_i(t_x) < LTH_i$ for entire T then S = Listen

OR

If S = Listen AND H = 0 for entire T then S = Probe
Where LTH_i is a threshold parameter for sensor i.

• B2 (Hormone Transmission rule)

A node with S = Probe transmits a hormone $H(C_{xi}, C_{yi}$, xi, yi, $r_i(t_x))$

if there is zero-crossing between $r_i(t_{x-1})$ and $r_i(t_x)$

OR

$r_i(t_x) > HTH_i$

Where HTH_i is a threshold parameter for sensor i.

• B5 (Adaptive Sampling rule)

This rule is key to saving energy while taking sensor readings. It defines how often to take samples during the sampling period T when S=Probe. A node predicts when to take the next sample based on the past values of sensor readings. The next sampling instance t_{x+1} is given by:

$$t_{x+1} = t_x + (HTH_i - r_i(t_x))((t_x - t_{x-1})/(r_i(t_x) - r_i(t_{x-1}))) \qquad (5.1)$$

Here, $0 < t_i <= T$.

Finally we would like to say that the controlled spatial behavior of the nodes will be achieved by the algorithm explained above and the timely behavior of the sensor nodes are influenced by the DHM. Hence we can track the spatial as well as the time varying events effectively with the framework proposed. Instead of Gradient Based Event Tracking we would like to recommend DHM based algorithm.

5.3 Relocation System Model

The deployment protocol which we proposed can be used for sensor relocation. For example, when an event is detected by a sensor, the sensors neighboring that node can be drawn to the event location (pull scenario). However, moving the sensors from the neighboring cells may create new holes in that area. To heal these new holes, more sensors must move. This process continues until some cell having redundant sensors is reached and the sensors leaving this cell do not create new holes. Using the method, sensors may move several times, wasting energy. In addition, since many sensors are involved, it may take a long time for the algorithm to terminate. Hence based on this observation, we first find the locations of the redundant sensors, and then design an efficient route for them to move to the destination.

To determine which sensor(s) is redundant is a challenging problem. It is hard for a single sensor to independently decide whether its movement will generate a coverage hole. To make such a decision, the sensor requires information about whether its neighbours will move or not. More specifically, a number of sensors located closely must determine the redundant sensors among themselves. A cell-based architecture is one solution for this problem. Already we have divided the target field into cells as a part of deployment algorithm. The cell head is responsible for collecting the information of its members, and determining the existence of redundant sensors based on their locations. The cell head monitor its group members and initiate a relocation process in case of new event. Further this organization can facilitate data aggregation, routing, etc., in addition to finding the redundant sensors.

With the cell-based model, the sensor relocation problem can be reduced to two sub-problems: finding the redundant sensors and then relocating them to the target location. Figure 5.2 illustrates the sensor relocation problem when cells are used, the black nodes are used to represent cell heads. Each cell is indexed by a tuple, whose first number is used to represent the column and the second number is used to represent the row. Cells (0, 6), (1, 5), and (0, 4) have redundant sensors. When there is an event at cell (4, 0), its cell head first needs to locate the redundant sensor and then relocate some sensor to cover that event. For the first problem, we can use a Cell-Quorum solution to quickly identify the redundant sensors. For the second problem, we use a cascaded movement solution to relocate sensors in a timely and energy efficient way.

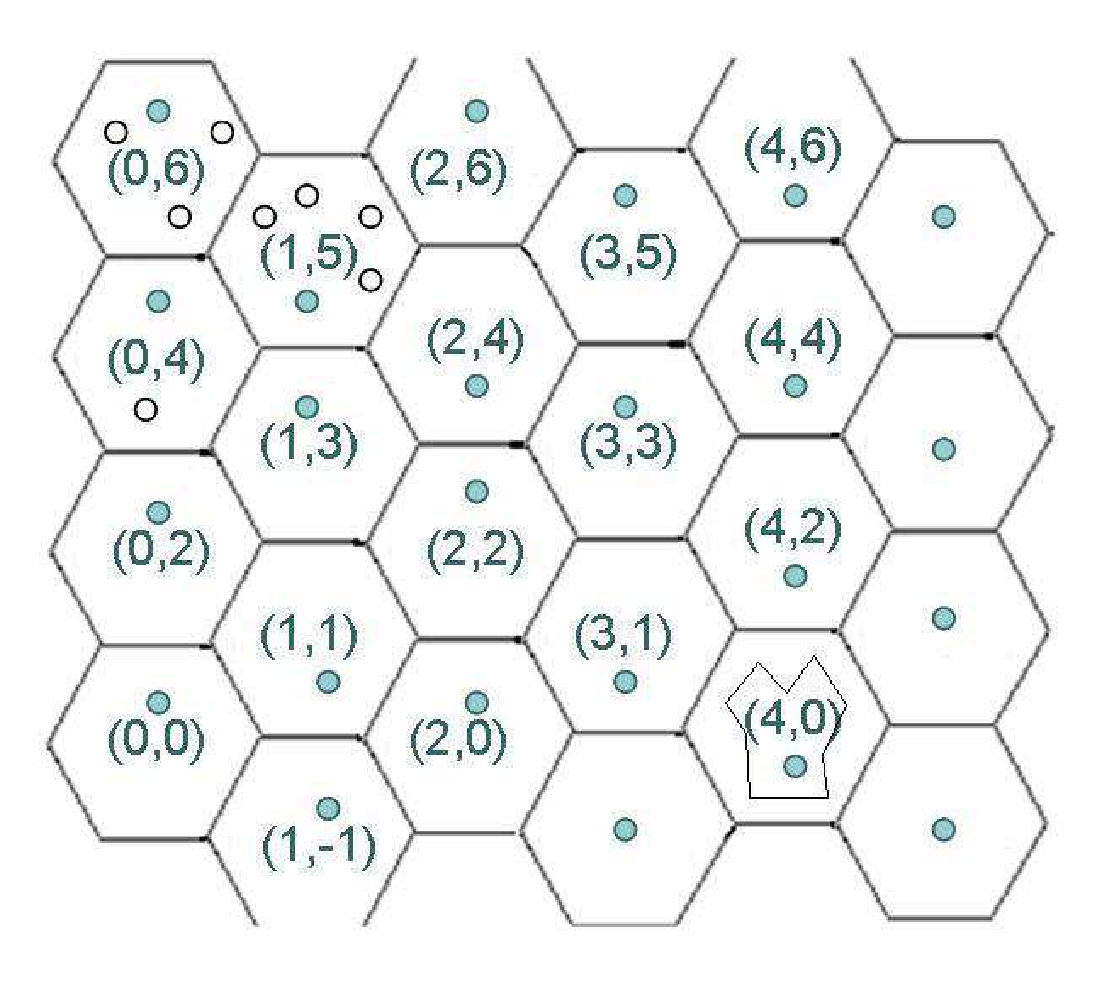


Figure 5.2: Hexagonal System Model

5.4 Finding Redundant Sensors

The problem of finding redundant sensors has some similarity to the publish/subscribe problem, where the publisher advertises some information and the subscriber requests the information. Mapping the terminology to our problem, the cells that need more sensors are the subscribers, and the cells that have redundant sensors are the publishers. In the publish/subscribe system, the matching of a request to an advertisement is called matchmaking. Generally, there are three types of solutions for matchmaking:

- Matchmaking occurs at the subscriber, which is referred as “broadcast advertisement”. In our problem, this is similar to letting the cells having redundant sensors flood this information. Later, when some cell needs redundant sensors, it can get the information quickly.
- Matchmaking occurs at the publisher, which is referred as “broadcast request”. In our problem, this is similar to letting the cells that need sensors flood the request. The cell that has redundant sensors replies after receiving the request.
- Matchmaking happens in the middle of the network. In our problem, this is similar to letting the supply cell advertise the information to some intermediate cells from which the demand cell obtains the information.

The message complexity will be very high if we use the broadcast advertisement approach, which requires two network-wide broadcasts for each redundant sensor: one for advertisement and the other for data update after the redundant sensor moves. For the broadcast request approach, the delay is relatively long since it is on-demand. Therefore, we prefer the third solution, which can achieve a balance between message complexity and response time.

By organizing cells as quorums, each advertisement and each request can be sent to a quorum of cells. Due to the intersection property of quorums, there must be a cell which is the intersection of the advertisement and the request. The cell head will be able to match the request to the advertisement. A simple publisher quorum can be constructed by choosing the nodes in a row and a column and subscriber quorum can be constructed by choosing the nodes in diagonal cells.

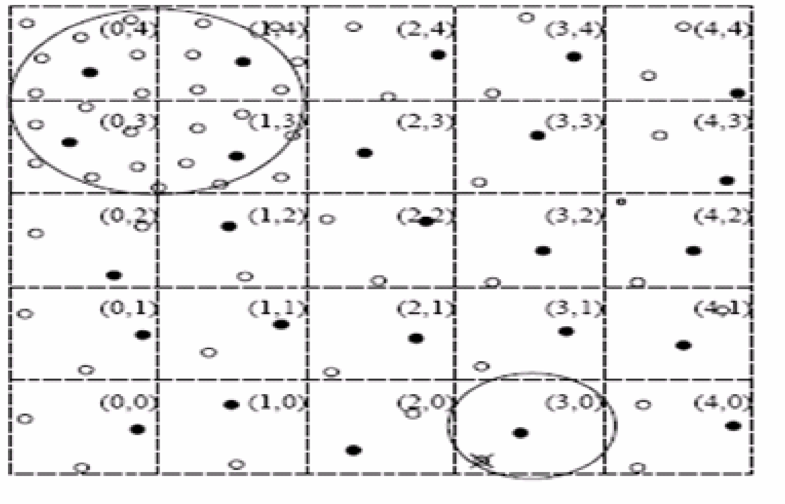


Figure 5.3: Grid Quorum System model

For example, as shown in Fig. 5.2, suppose cell (1,5) has redundant sensors, it only sends the advertisement to cells in a row ((1,5), (3,5), (5,5), (7,5), (9,5)) and a column ((1,3), (1,1), (1,-1)).When cell (4,0) is looking for redundant sensors, it only needs to send a request to diagonal cells ((4,0), (3,1), (2,2), (1,3)) and ((3,-1), (4,0), (5,1)). The intersection node (1, 3) will be able to match the request to the advertisement. Suppose N is the number of cells in the network. By using this quorum based system, the message overhead can be reduced from O(N) to $O(N^{.5})$. The message overhead is very low compared to flooding. We can further reduce the message overhead by observing the specialty of our problem.

This matchmaking technique not only suits for the single event but for also for the multi event detection. Let us consider a case that an event is detected at a cell (0,0) in the figure 3.1 then the match making will takes place directly at the node (0,0) itself as it is a member in the quorum, For the event that was detected simultaneously at a cell (4,2), match making will be takes place at the cell (2,4). In the following section we have discussed the Grid Quorum idea for sensor relocation.

5.4.1 Grid Quorum

In this section, we first give the background and motivation of the Grid-Quorum idea. Then, we present the detailed solution and illustrate its advantage in terms of message complexity and response time. The problem of finding redundant sensors has some similarity to the publish/subscribe problem [32], [13] where the publisher advertises some information and the subscriber requests the information. Mapping the terminology to our problem, the grids that need more sensors are the subscribers, and the grids that have redundant sensors are the publishers. In the publish/subscribe system, the matching of a request to an advertisement is called matchmaking. Generally, there are three types of solutions for matchmaking which we have discussed above.

Different from the traditional publish/subscribe problem, the information in our system is not reusable. The information about the redundant sensor can only be used once, since it may be changed after the redundant sensor moves to the requesting place. Due to this special property, the message complexity will be very high if we use the broadcast advertisement approach, which requires two network-wide broadcasts for each redundant sensor: one for advertisement and the other for data update after the redundant sensor moves. For the broadcast request approach, the delay is relatively long since it is on-demand. Therefore, we prefer the third solution, which can achieve a balance between message complexity and response time. In this type of solution, a structure like that in [13]can be used to facilitate the matchmaking between the advertisement and the request. Since the data may not be re-used, this structure should be simplified compared to that in [13], [33]; otherwise the benefit may not be worth the cost. Therefore, we need a simple and low-cost structure for matchmaking.

By organizing grids as quorums, each advertisement and each request can be sent to a quorum of grids. Due to the intersection property of quorums, there must be a grid which is the intersection of the advertisement and the request. The grid head will be able to match the request to the advertisement. A simple quorum can be constructed by choosing the nodes in a row and a column. Instead of flooding the network with advertisements or requests, the request and the advertisement are only sent to nodes in a row or column. For example, as shown in Fig. 5.3, suppose grid (0,3) has redundant sensors, it only sends the advertisement to grids in a row ((0,3), (1,3), (2,3), (3,3), (4,3)) and a column ((0,4), (0,3), (0,2), (0,1), (0,0)).When grid (3,0) is looking for redundant sensors, it only needs to send a request to grids in a row ((0,0), (1,0), (2,0), (3,0), (4,0)) and a column ((3,4), (3,3), (3,2), (3,1), (3,0)). The intersection node (0,0) will be able to match the request to the advertisement. Suppose N is the number of grids in the network. By using this quorum based system, the message overhead can be reduced from O(N) to $O(N^{.5})$. Although the message overhead is very low compared to flooding, we can further reduce the message overhead by observing the specialty of our problem.

The Grid-Quorum Solution :

In our Grid-Quorum system, we do not require the intersection of any two quorums. Instead, we deploy two coterie, called supply coterie and demand coterie separately, and only require that the quorum belong to the supply coterie intersects with all quorums in the demand coterie, and vice versa. The formal definition is as follows. Given a nonempty set U, there is a supply coterie C_s and a demand coterie C_d, which are the sets of U's subsets. Each subset P_s in coterie C_s is called a supply quorum and each subset P_d in coterie C_d is called a demand quorum.

To construct a Grid-Quorum, the grid heads belong to the grids in one row are organized into one quorum, called supply quorum and the grid heads belong to the grids in a column are organized into one quorum, called demand quorum. All the supply quorums compose the supply coterie, and the demand quorums compose the demand coterie. In this way, the natural geographic relation ensures that every supply quorum has intersection with all the demand quorums and vice versa. When a grid has redundant sensors, the grid head propagates this information through the supply quorum to which it belongs. When any grid wants more sensors, the grid head needs only to search its demand quorum. Since every demand quorum has intersection with all supply quorums, the grid head can get all the information about redundant sensors. We can see that using the geographic information reduces the cost of building Grid-Quorum to almost zero. Still using the example of Fig. 5.3, Grids (0,4), (1,4), (0,3) and (1,3) have redundant sensors, while grid (3,0) needs more sensors. The grid head of (1,3) propagates its redundant sensor information through its supply quorum ((1,4), (1,3), (1,2), (1,1), (1,0)). The grid head in grid (3,0) searches its demand quorum ((0,0), (1,0), (2,0),(3,0), (4,0)). Grid (1,0) can reply the information about redundant sensors. Compared to using the quorum in the last example, using grid-quorum cuts the message by half.

Grid quorum system is used for self organizing and self replacement of damage sensors, where as we are using the same principle for relocation of sensors after event detection.

5.5 Sensor Relocation

Having obtained the location of the redundant sensor, we need to determine how to move the sensor to the target location (destination). Moving it directly to the destination is a possible solution. However, it may take a longer time than the application requirement. For example, a sensor monitoring a strategic area dies and the application specifies that the maximum tolerable time for such a sensing hole is thirty seconds. If the redundant sensor is 100 meters away and it takes at least one minute for the sensor to reach its destination, the application requirement cannot be met. Moreover, moving a sensor for a long distance consumes too much energy.

If the sensor dies shortly after it reaches the destination, this movement is wasted and another sensor has to be found and relocated.

We propose to use a cascaded movement to address the problem. The idea is to find some cascading (intermediate) nodes, and use them for relocation to reduce the delay and balance the power. As shown in Fig. 5.6, instead of letting the redundant sensor s_3 move directly to the destination, s_1 and s_2 are chosen as cascading nodes. As a result, s_3 moves to replace s_2, s_2 moves to replace s_1, and s_1 moves to the destination. Since the sensors can first exchange communication messages (i.e., logically move), and ask all relevant sensors to (physically) move at the same time, the relocation time

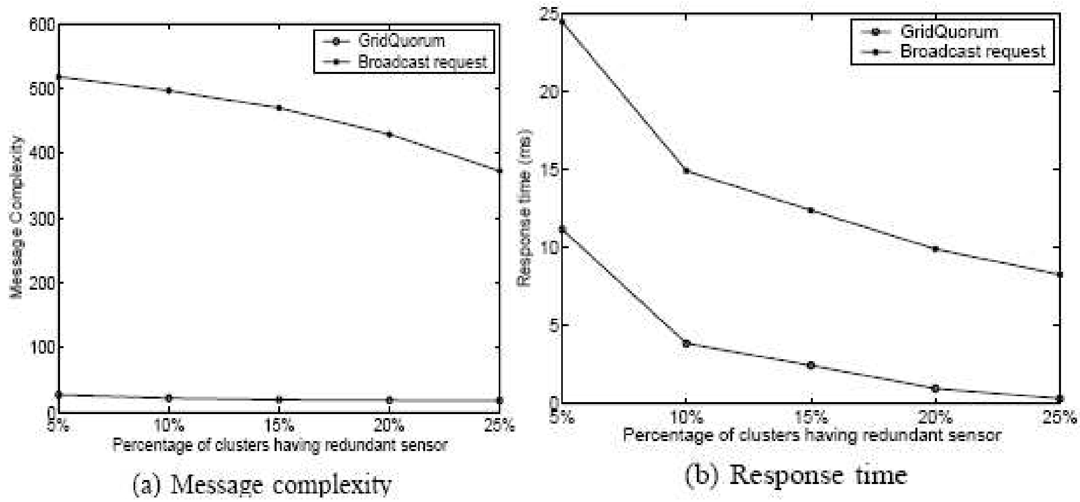
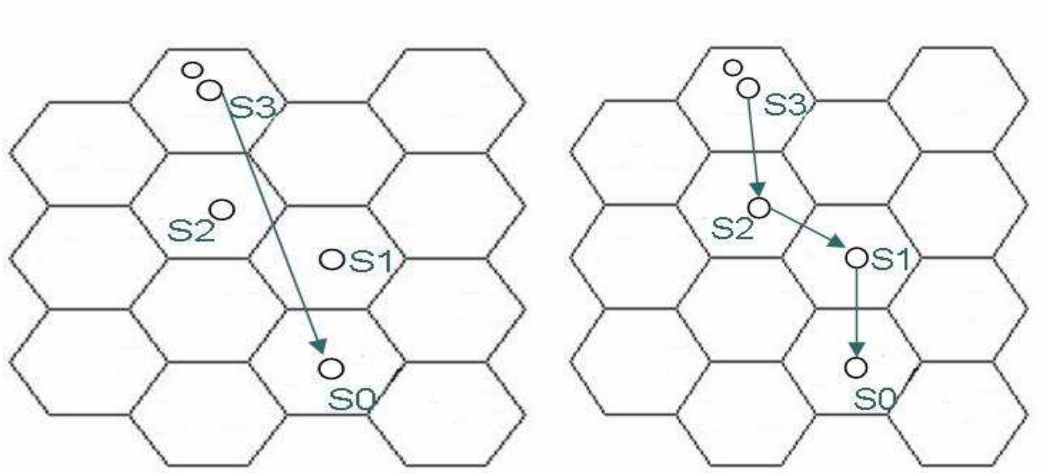


Figure 5.4 Comparison between the Grid-Quorum solution and the “Broadcast Request” approach



a. Direct movement b. Cascaded movement

Figure 5.5Cascaded movement

is much shorter. A node s_i which moves to replace another node s_j, is referred to as s_j 's successor, and s_j is referred to as s_i's predecessor. In Fig. 5.6 s_3 is s_2's successor and s_2 is s_3's predecessor. We also introduce a virtual node s_0, which is used to represent the target location. It may represent the failed sensor or the location where an extra sensor is needed to increase the sensing accuracy. In Fig. 5.6, we say s_0 is s_1's predecessor and s_1 is s_0's successor. Selecting cascading nodes is not easy since the sensor nodes may be used by some application and their movement may affect the sensing or communication tasks they are performing. To ensure that this effect is within application's requirement, each sensor s_i is associated with a recovery delay T_i. After s_i's movement, its successor must take its place within T_i. T_i is determined by the application based on the critical level of s_i's sensing task, the size of the coverage hole generated by s_i's movement, and other application factors. We use T_0 to represent the recovery delay of the relocation event. It can be the maximum recovery delay of the failed sensor or the time limit for an additional sensor being placed at s_0. The T value imposes restrictions on the spatial relationship and departure time of the cascading nodes. We use t_i to denote the departure time of s_i and d_{ji} to denote the distance between s_i and s_j . The following Inequality must be satisfied if s_j is s_i's successor.

$$d_{ji}/speed - (t_i - t_j) \leq T_i \qquad (1)$$

For simplicity, ti is normalized to be the time period after the relocation request is sent and t_0 (for s_0) is set to be 0. Based on Inequality (1), whether s_j can be the the successor of s_i is not determined solely by its distance to s_i, but also s_i's departure time. If s_i moves at t_0 (0), s_j must be within speed _ T_i from s_i; if s_i moves after another t minutes, s_j can be farther away from s_i as long as d_{ji} _ speed _ (T $_{i+t}$). Whether s_i can stay at its place for this t minutes or must move immediately is determined by its own predecessor. For example, if s_i is the successor of s_0, and d_{i0} is shorter than speed_T_0, s_i can flexibly move between (0, $T_0 - d_{i0}$/speed). In this case, we normally let s_i move at T_0 − di0/speed (the upper limit) such that more sensors can be chosen as s_i's successor and we can choose the best one. The set of cascading nodes for relocation and their departure time together is defined by a cascading schedule. For example, in Fig. 5.6 Choice 2, $s_3(t_3)$ => $s_2(t_2)$ => $s_1(t_1)$ =>s0 is a cascading schedule, which can be used to recover a sensor failure. Certainly, the cascading scheduling should make sure that the recovery delay is satisfied; i.e., Inequality (1) is satisfied. For example, Choice 1 is not a cascading schedule since the sensor failure cannot be recovered within the required time.

5.5.1 The Metrics to Choose Cascading Nodes

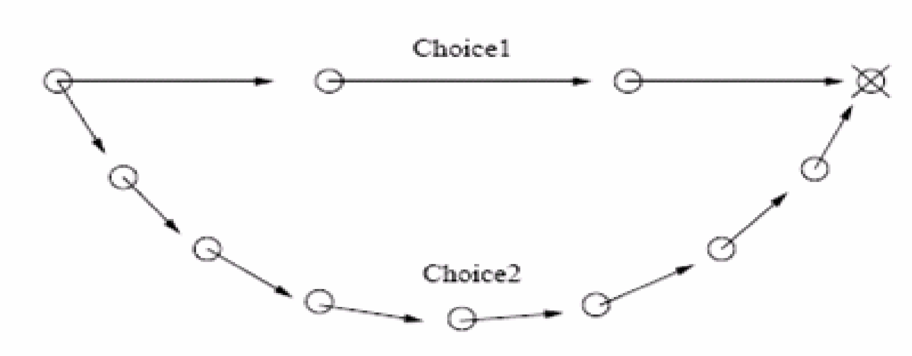


Figure 5.6 cascaded movement

The cascading schedule should minimize the total energy consumption and maximize the minimum remaining energy so that no individual sensor is penalized. However, in most cases, these two goals cannot be satisfied at the same time. As shown in Fig. 5.7, suppose all sensors have the same amount of power. Choice1 consumes less energy, but the involved sensors will have lower remaining energy. Sensors in Choice2 have higher remaining energy, but the total energy consumption of Choice 2 is higher than that in choice1. There is a tradeoff between minimizing the total energy consumption and maximizing the minimum remaining energy, and we want to find a balance between them. Before presenting our solution, we first show some observations. Based on the sensor deployment result generated by running VOR, we randomly choose some sensor and deplete its energy. Then, all cascading schedules to recover the failed sensor are enumerated and compared in terms of the total energy consumption and the minimum remaining energy. Here, the recovery delay (T_i, $i \neq 0$) is relaxed for better observation, but the relocation time (T_0) is calculated for reference. The cascading schedules which are worse than some other schedule in both metrics (total energy consumption and minimum remaining energy) will be ignored; that is, we only keep the cascading schedules which perform better than others at least in terms of one metric. Fig. 5.8 shows the total energy consumption and the minimum remaining energy of these schedules in an increasing order. As shown in the figure, the total energy consumption is almost flat at the beginning and then significantly increased, whereas the minimum remaining energy has a steep increase at the beginning and then becomes flat. This observation motivates us to achieve a good balance between minimizing the total energy consumption and maximizing the minimum remaining energy.

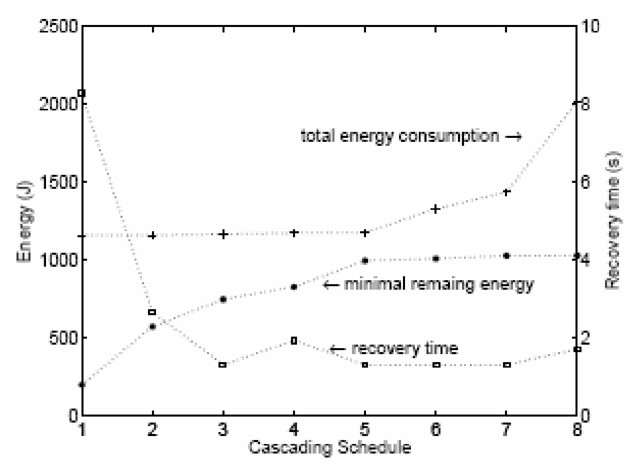


Figure 5.7 : Tradeoff

From the observation, we can see that it is possible to continuously spend a little more energy for a much higher minimum remaining energy until a turning point after which the cost is higher but the gain is less. The cascading schedule just before this turning point should be the best schedule. In other words, the best schedule is the schedule with the minimum difference between the total energy consumption and the minimum remaining power. This new metric can be explained in a mathematical way.

Suppose there are two cascading schedules with E1 and E2 as their total energy consumption, and E_{min1} and E_{min2} as their minimum remaining energy. Schedule 1 is chosen since E1 −Emin1 ≤ E2 −Emin2. This inequality can also be expressed as E1−E2 ≤ Emin1−Emin2; i.e., the cascading schedule with more advantage and less disadvantage should be chosen.

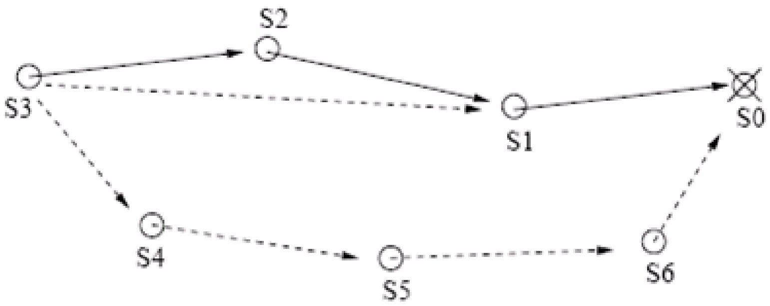
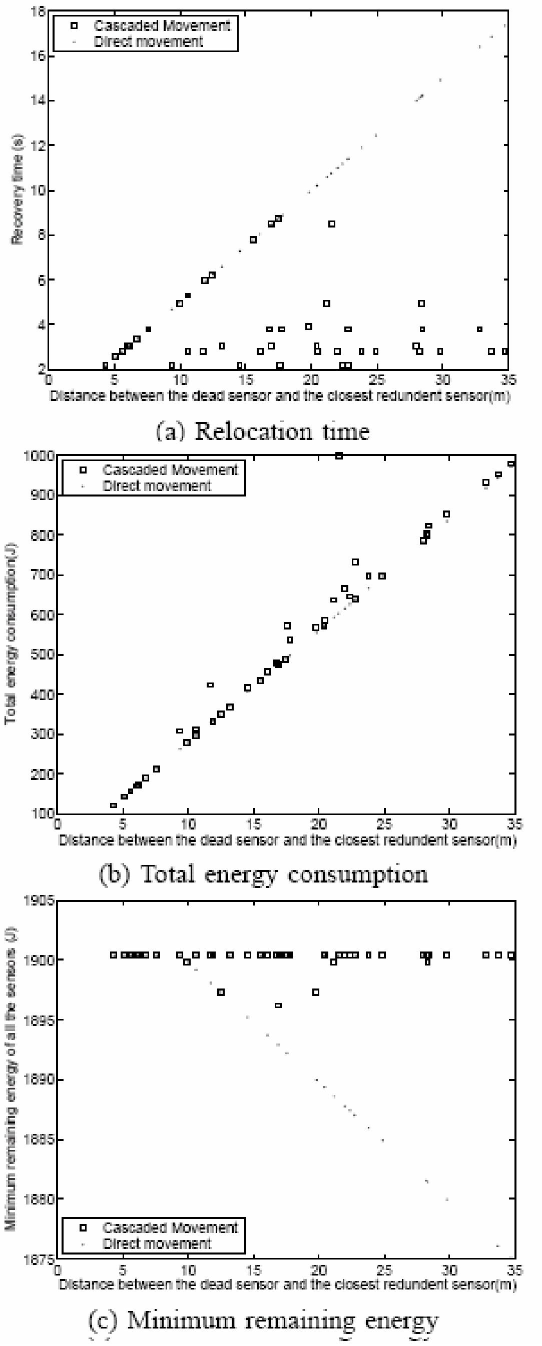


Figure 5.8 : Cascaded movement

Fig.5.9 uses an example to further explain the reason. In Fig. 5.9, moving s3 directly to the target location is the most energy efficient solution. However, in this way, s3 will be penalized, and its minimum remaining energy will be significantly reduced. If s1 is added as a cascading node, the load of s3 can be shared and the minimum remaining energy can be improved. Since the total length of the zigzag line s3s1s0 is only a little bit longer than the length of s3s0, only a slightly more power is needed. If more sensors close to the line s3s0 are chosen as cascading nodes, the load can be further shared and the minimum remaining power can be further improved. Certainly, if some sensor close to this line has very low energy, it should not be selected for cascading. When all eligible sensors close to this line have been chosen as cascading nodes, a balanced and efficient schedule is obtained. Starting from this point, if we want to further improve the minimum remaining energy, faraway sensors such as s4, s5 and s6, have to be chosen. However, in this way, the total energy consumption will be significantly increased, and then it may not be a good solution. So the cascading schedule with minimum difference between the total energy consumption and minimum remaining power is referred to as the best cascading schedule. The cascading schedule with the least total energy consumption is referred to as the shortest schedule.

5.5.2 Cascaded Movement vs. Direct Movement

In this section, we compare the cascaded movement approach with the direct movement approach, which moves the redundant sensor directly to the target location. Simulation results are shown in Fig. 5.10. As can be seen (Fig.5.10 (a)), the relocation time can be significantly reduced in the cascaded movement approach. As for energy consumption, direct movement is better, but its advantage over cascaded movement is very limited (Fig5.10 (b)). This proves that cascaded movement is energy efficient. On the other hand, the minimum remaining energy of using cascaded movement is much better than that of direct movement. If the redundant sensor has relatively high power, moving it directly to the target location may not affect the minimum remaining power; otherwise, it may significantly reduce the minimum remaining power, especially when the moving distance is long. This explains why the minimum remaining energy drops proportionally as the distance increases in the direct movement approach (see Fig. 5.10 (c)). In our solution, the metric used to get the best cascading schedule is to minimize the difference between the total energy consumption and the minimum remaining power. Since there are other objective metrics existing, we compare our solution to other alternatives. Since we have shown (see Fig. 5.10 (b)) that the total energy consumption of our approach is similar to the direct movement approach, which is optimal, we only compare our approach with another alternative that maximizes the minimum remaining power. Comparisons when the remaining energy is very different, meanwhile, the minimum remaining energy is at most slightly lower than its alternative. Between these two settings, our approach saves more energy when the remaining energy is similar. The reason is as follows. Sensors with relatively more energy must be involved to maximize the minimum remaining energy. When the remaining energy is similar, it is not likely to find nearby sensors with high energy. Then, faraway sensors are more likely to be involved, and more energy will be consumed compared to our solution. On the other hand, when the remaining energy is similar, the disadvantage of our solution is a little bit larger since only nearby sensors are involved in the relocation. These sensors may become the sensors with minimum remaining energy after relocation and the minimum remaining energy among all the sensors is reduced consequently. When the remaining energy is very different, both approaches have similar minimum remaining energy since a sensor with minimum remaining energy is more likely not involved in the relocation and the minimum remaining energy of the network does not change after the relocation.



(a) Relocation time

(b) Total energy consumption

(c) Minimum remaining energy

Figure 5.9: Comparison between cascaded movement and direct movement

5.6 Event Coverage

Our project consists of integrating the both the DHM model and COVERAGE to implement a distributed senor network system to maximize the coverage of events by multiple sensors. The DHM model will make to reconstruct the spatio temporally varying signal by giving mobility to sensor nodes. The Algorithm becomes effective with inherent communication capabilities of the sensor networks. Here the digital hormone model is implemented with the coverage constraints so that the events generated in any region cannot be missed. For implementing we have developed master slave protocol.

- Each node will four states MASTER, NORMAL, SLAVE, PROBING.
- When node turns on it will makes itself NORMAL node and bids to become head of that cell if no other node is head of that node.
- Master node is responsible for covering the events that may occur in that region over time.
- If there is another Master Node in that cell it will register itself as slave node for it.
- When it is in slave mode it will wait for some random time. If no event is occurring in that region over time it will move from that cell and signal the MASTER node and it will follow the Digital Hormone Model (DHM).

• Master node will know about the occupancy of the neighboring cells. If any cell is vacant it will instruct one of its registered slave nodes to that cell.

• When ever an event occurs in that cell the Master of that cell will broadcast Event trigger message. In response to these the neighboring cells Master will send the details of the slave nodes which are nearer to that cell. The Master node of event generating cell will take one of the mobile node which is nearer to it will be instructed to track the event.

• If the Master node of that cell has to move to track the event in the neighbor cell it will notify in that cell about it departure. So new head will be elected by bidding.

• If the only master node in that cell is moving away to track the event then it will inform its neighbors about the coverage hole in the cell. By this using coverage algorithm the cell is filled with the redundant sensors. Even if there is no redundant sensor the master node itself will move to cover that cell. It may create oscillations but it will not sustain for long as are assuming more redundant sensors in the region.

5.6.1 Algorithm

The distributed algorithm will be executed by master nodes to detect the events that occur in space. Master nodes will learn about the neighboring cells and will send the nodes to cover that area if it is not covered before. When event occurs in a cell Master node will detect it and request the adjacent cells to monitor the events occurred in that place. If no event is occurring for some time the slave node will move randomly and senses the environment and gathers information about the environment (DHM).

Algorithm for implementing Event Coverage is given below.

Algorithm 1 (Event Coverage Algorithm)

0. Partition the sensing field into small sub cells C_k having a regular pattern (hexagonal);

1. i=0;

2. While terminating condition not satisfied do

3. Select a cell head H_i^k from the nodes which are moved to a cell C_k ;

4. Each node n within a cell sends position P[n] to the cell head ;

5. Each cell head H_i^k learn the neighboring cells information A C_j € Cnbrs and constructs the adjacency list A_i^k ;

6. Each cell head $H_i{}^k$ selects the victim nodes $V_i{}^k$ that are to be sent to the neighboring cells C_j € Cnbrs ;

7. Assign each victim node a new position;

8. Notify the adjacent cells with positions of the victim nodes;

9. All the Victim nodes will move to the new cells;

10. Each cell head $H_i{}^k$ will continuously sense the light strength $L_k(t)$ at time t;

11. When Light strength $L_k(t)$ HTH high threshold it will request a node into from Cnbrs and constructs available list.

12. Event generating Cell Head $H_i{}^k$ will select victim whose distance d_i^k is less and assigns its a new position.

13. Victim V_i^k will notify its movement to the head of that cell.

14. When there is no event for long time Slave nodes will take a random direction to move and notify its Head H_i^k

15. i = i + 1 ;

16. end

The above algorithm can be modified for detecting multiple events.

Multiple Event Coverage

0. Partition the sensing field into small sub cells C_k having a regular pattern (hexagonal);
1. i=0;
2. While terminating condition not satisfied do

3. Select a cell head H_i^k from the nodes which are moved to a cell C_k ;
4. Each node n within a cell sends position P[n] to the cell head ;
5. Each cell head H_i^k learn the neighboring cells information A C_j € Cnbrs and constructs the adjacency list A_i^k ;
6. Each cell head H_i^k selects the victim nodes V_i^k that are to be sent to the neighboring cells C_j € Cnbrs ;
7. Assign each victim node a new position;
8. Notify the adjacent cells with positions of the victim nodes;
9. All the Victim nodes will move to the new cells;
10. Each cell head H_i^k will continuously sense the light strength $L_k(t)$ at time t;
11. When Light strength $L_k(t)$ HTH high threshold it will request a node into from Cnbrs and constructs available list.
12.Event generating cell head H_i^k checks if any other node is already in active state;
13.If No, Go To step 20;
Else
14.Declare H_i^k as upper or lower ,by comparing i and k.
(if I < =k H_i^k assigns as H_{upper} ;If i>k assigns as H_{lower}).
15.Declare the new head H_i^knew as H_{upper} or H_{lower} using step14.
16.If H_i^k and H_i^knew are different , H_i^k and H_i^knew will work separately following steps from step20.
Else
17. H_i^k go to step20.
18. H_i^knew go to wait state,
19.Check H_i^k is active or not
If active go to step18.
Else
H_i^knew => H_i^k
20. Event generating Cell Head H_i^kwill select victim whose distance d_i^k is less and assigns its a new position.
21.Victim V_i^k will notify its movement to the head of that cell.
22.When there is no event for long time Slave nodes will take a random direction to move and notify its Head H_i^k
23. i = i + 1 ;
24. end

The distributed algorithm will be executed by master nodes to detect the event that occur in space. Master nodes will learn about the neighboring cells and will send the nodes to cover that area if it is not covered before. When event occurs in a cell Master node will detect it and request the adjacent cells to monitor the events occurred in that place. At the same time more than one event occurs, master node will detect whether the event occurs in the same triangle (upper or lower of the diagonal elements). If the events are in the different triangle, the nearby master nodes will monitor the events and send nodes to cover that areas simultaneously. If the events are in the same triangle, the priority will be given to the master node of the first event, the others will go to wait state, till first one's process completes.

If no event is occurring for some time the slave node will move randomly and senses the environment and gathers information about the environment (DHM).

The main advantage of our solution is only a minimum number of nodes are involved and many other sensors are not affected. The energy consumption is low and the remaining energy is high. In summary, first finding the redundant sensor and then relocating it to the target location is much better for sensor relocation.

Chapter 6

Simulation Results

Important criteria for our framework are the mobility models used by the mobile sensor nodes. In the predefined mobility model, we set the initial and final (x,y) co-ordinates of the mobile node along with the path to follow and once simulation starts the nodes will move till, end of the simulation. But in our framework the node locations are calculated dynamically and the nodes will move with intermediate stops.

We used java simulator for validating our framework. The simulator can also be used to view the topology generated while executing the algorithm. The simulator assumes no packet collisions. It also assumes that there are no packet errors during transmission and reception. In other words, we assume a perfect wireless channel. Figure 6.1 shows the GUI panel for the simulator in which it provides two select buttons which can be used for selecting different initial distribution of the nodes.

The three different initial configurations of the nodes are:

1. “Uniform Distribution” of nodes in which all the cells are occupied by at least one node.

2. “Random Distribution” of nodes in which all the nodes are distributed through out the field in a random fashion.

3. “Single Distribution” of nodes in which all the nodes are placed in a single cell.

By default it will have the random distribution of sensor nodes. It provides a time slider at the bottom of the panel which can be used to slide the time epochs, we can drag the time slider to a point so that it will show the corresponding node distribution at the time epoch. The simulator runs for 2500 time epochs. At every time epoch we will get the previous state of the node and the node will executes the algorithm based on the previous state, then the current state will be saved again.

6.1 Simulation Details

The distributed algorithm for dynamic sensor network has been simulated using the simulator explained in the previous section. Figures 6.2 6.3 and 6.4 are the results of an example simulation run with the following simulation parameters:

1. Number of nodes = 140.
2. Area of the simulation region of WSN = 1025X850 sq.units.

The side of the hexagon is varied from 70 units to 100units and for each case we consider 3 different initial configurations of the nodes Along with these three confi urations we also changed the probability of electing head Phead from .5 to .7 and side of the polygon. We have taken the uniform distribution of nodes as an important test case for our algorithm because for the uniform distribution of nodes even the algorithm executes the nodes should not move for covering the space as there is an 100% initial coverage. For the remaining two configurations as the algorithm executes the alignment of the nodes approximates the uniform distribution after a sufficient of iterations. Figures 6.2 shows the coverage plot for the hexagon of side S=90 units for three configurations explained above. For Random deployment the initial number of cells occupied is 47 and after 8 iterations the occupancy increased to 54 which is actually the total number of cells present in the given field for side S=90 units. Number of iterations for complete coverage in case of side=90 units are less as the initial occupancy is high but the number of iterations will increases with decrease in the initial occupancy for this random distribution for a given P_{head}.

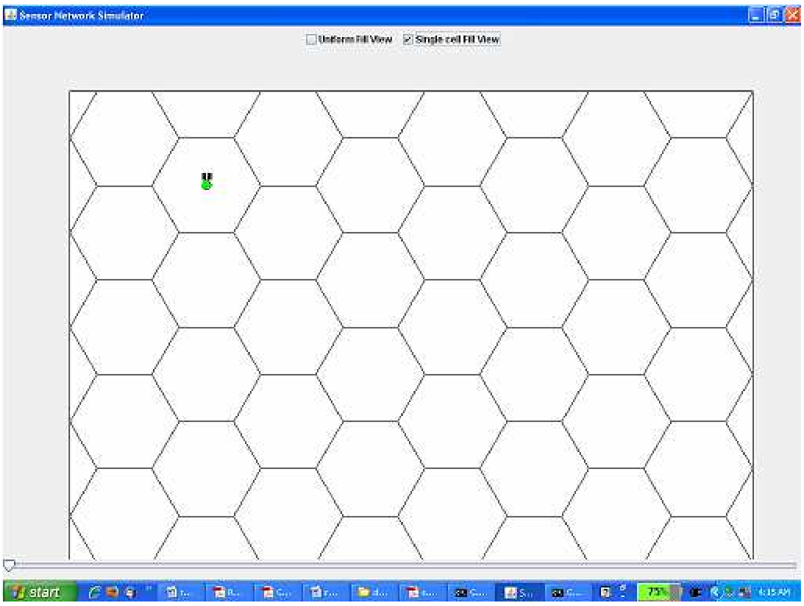


Figure 6.1: Simulation Panel

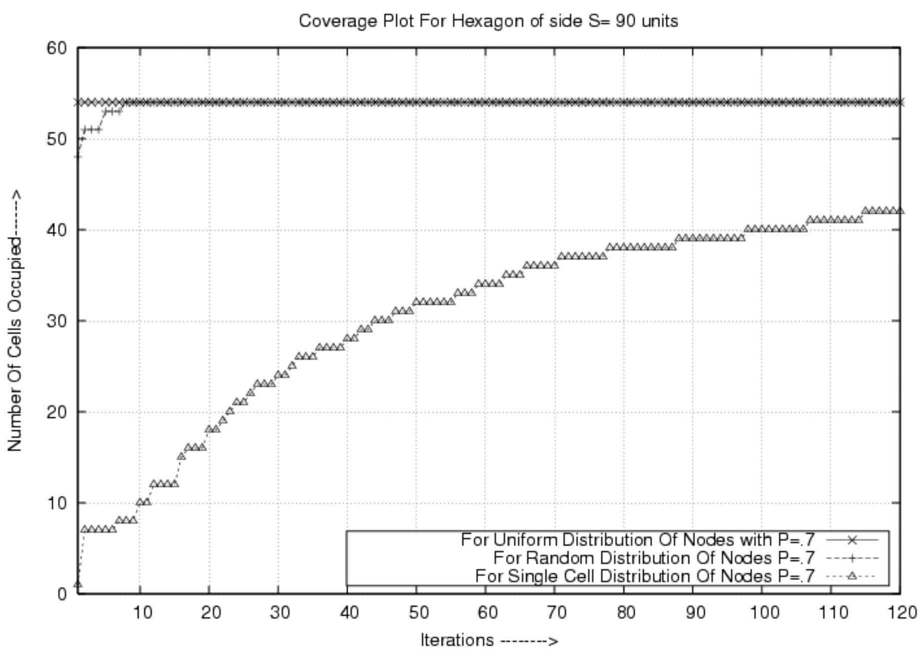
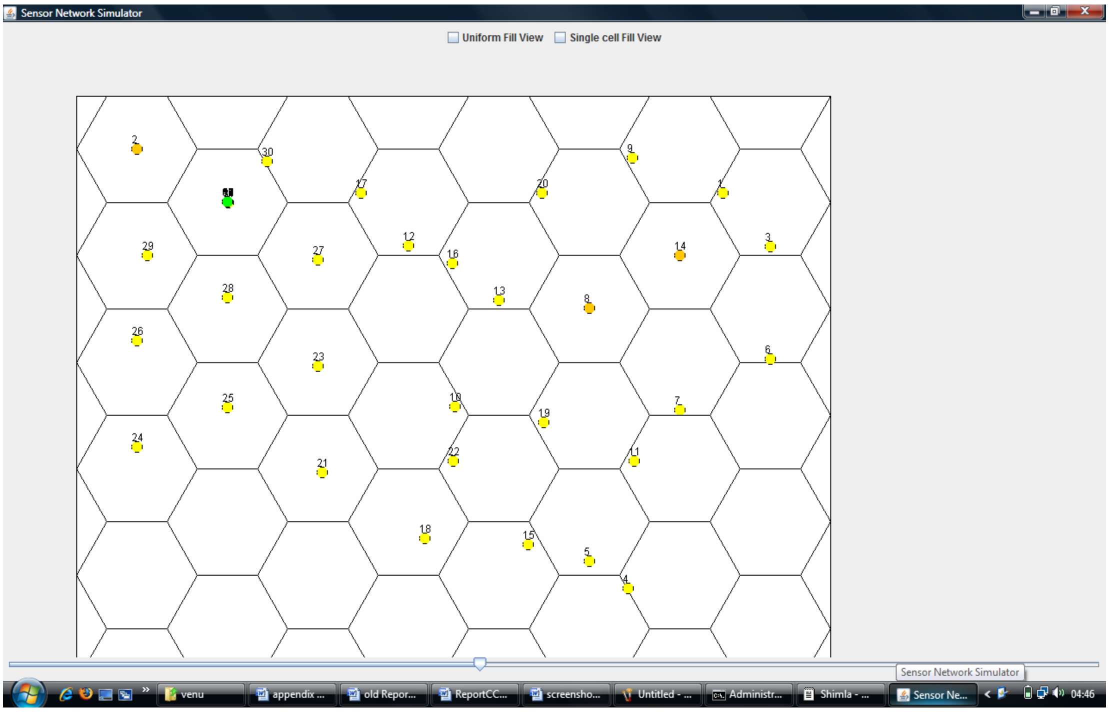
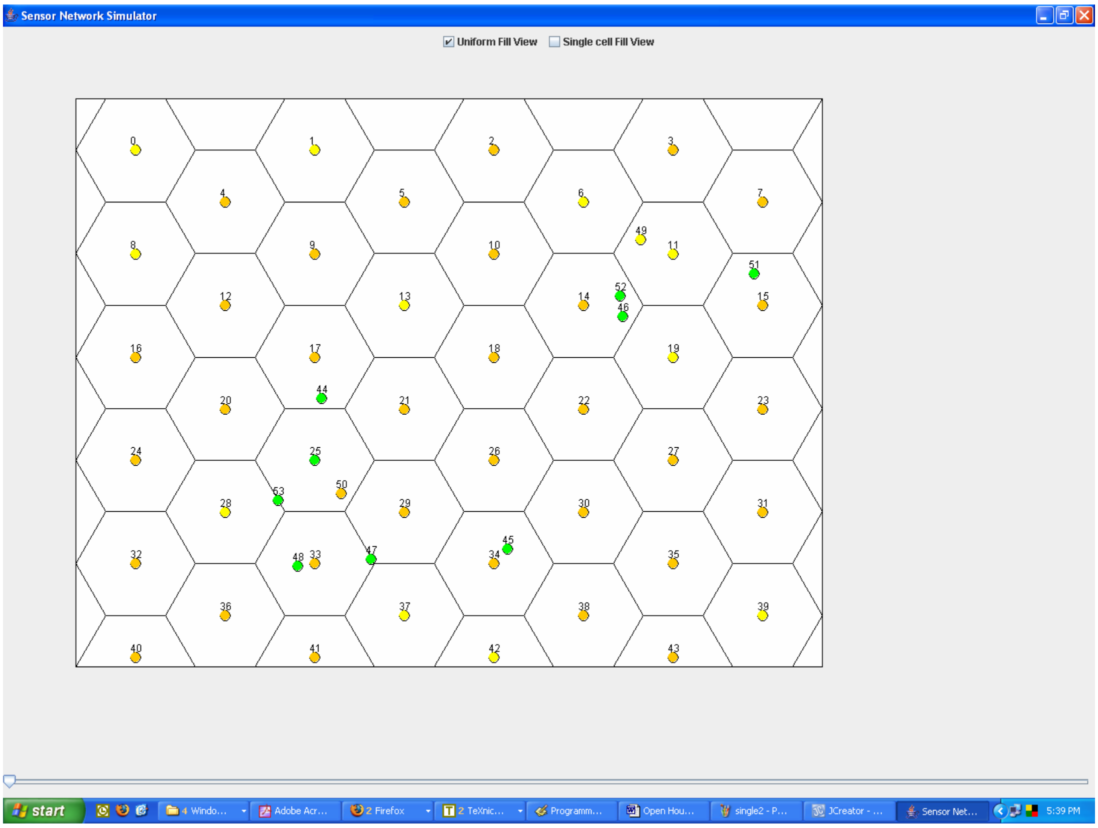
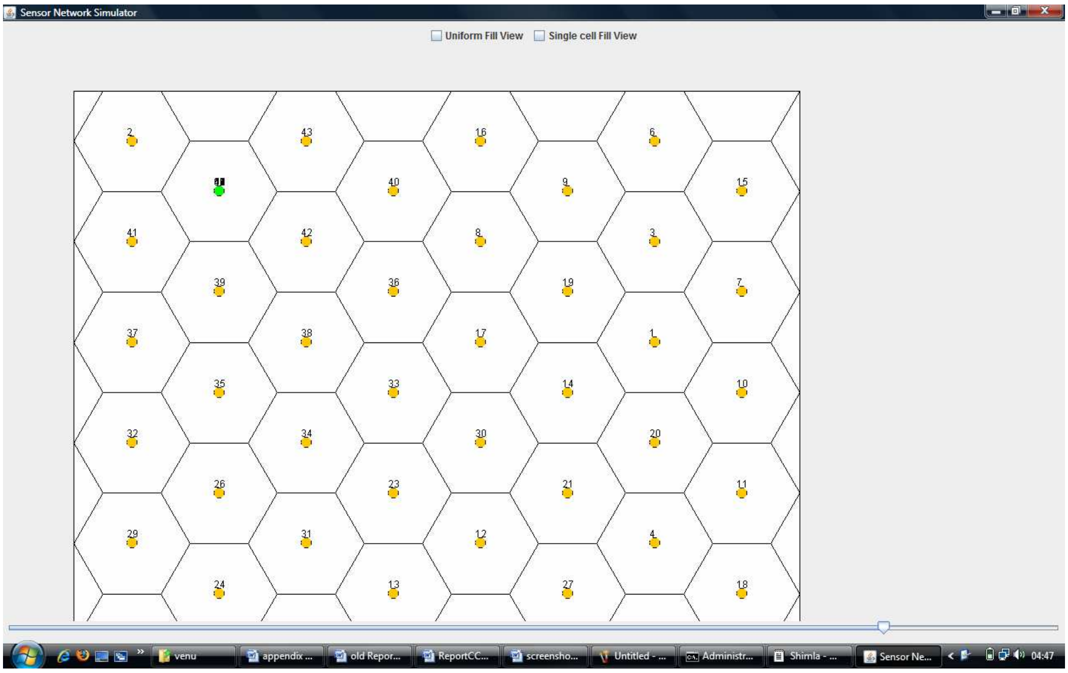


Figure 6.2: Coverage plot for side 90 units for uniform, random and single cell distributions



(a) Initial Distribution (b) Intermediate stage

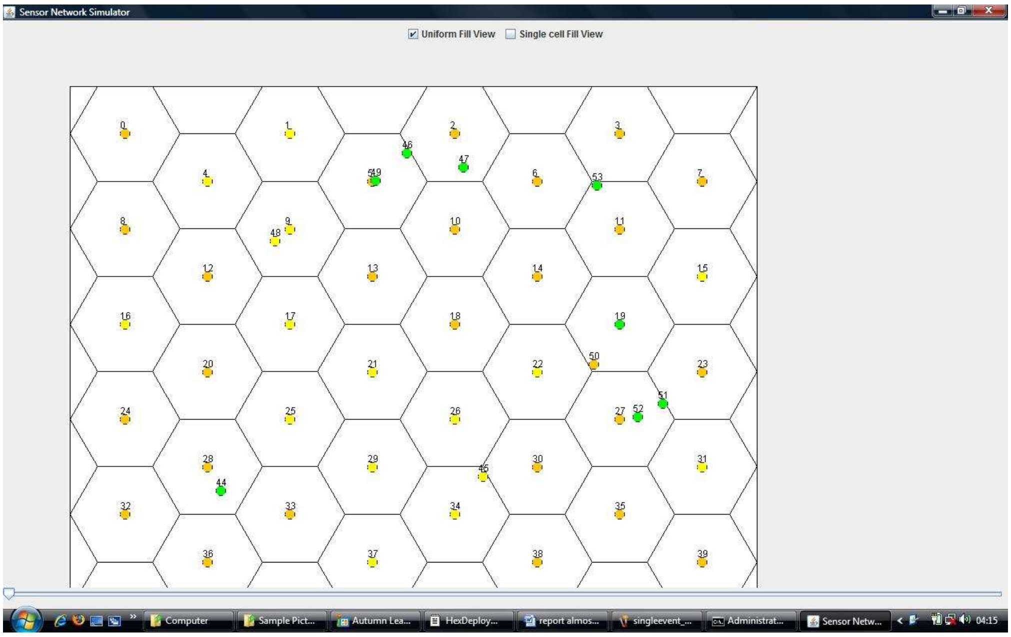




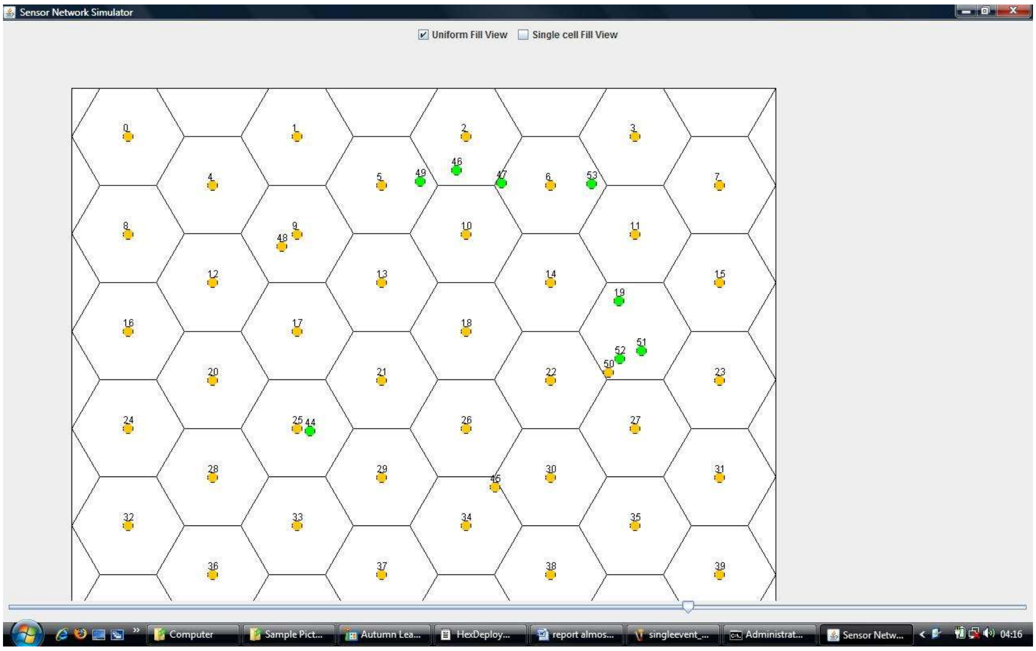
(c) Final stage of deployment phase

Figure 6.3: Deployment phase

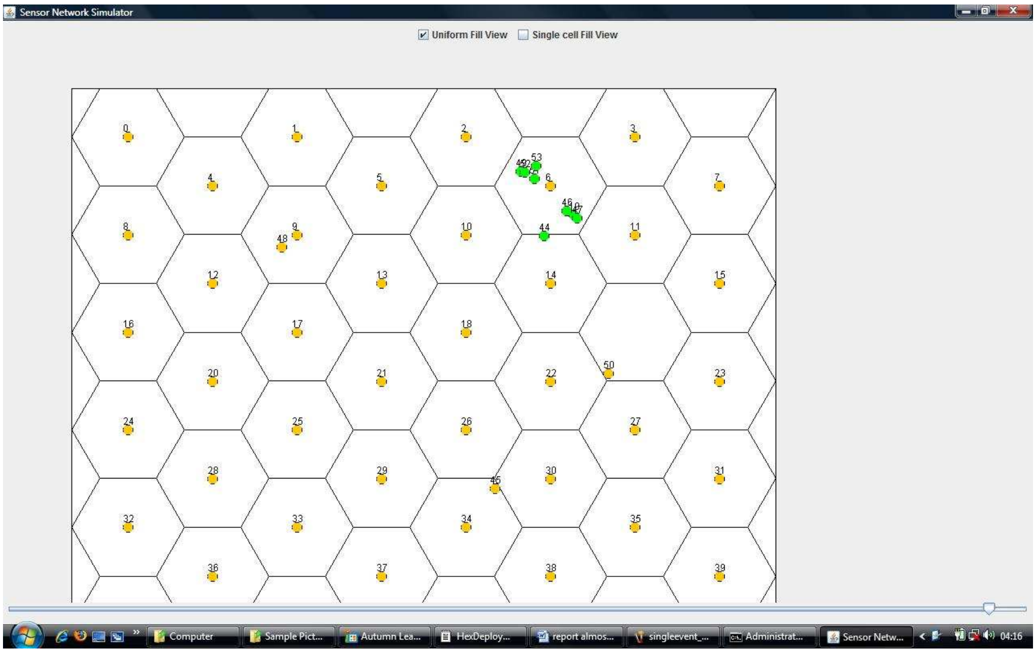
Single event generation



(a) Initial Distribution



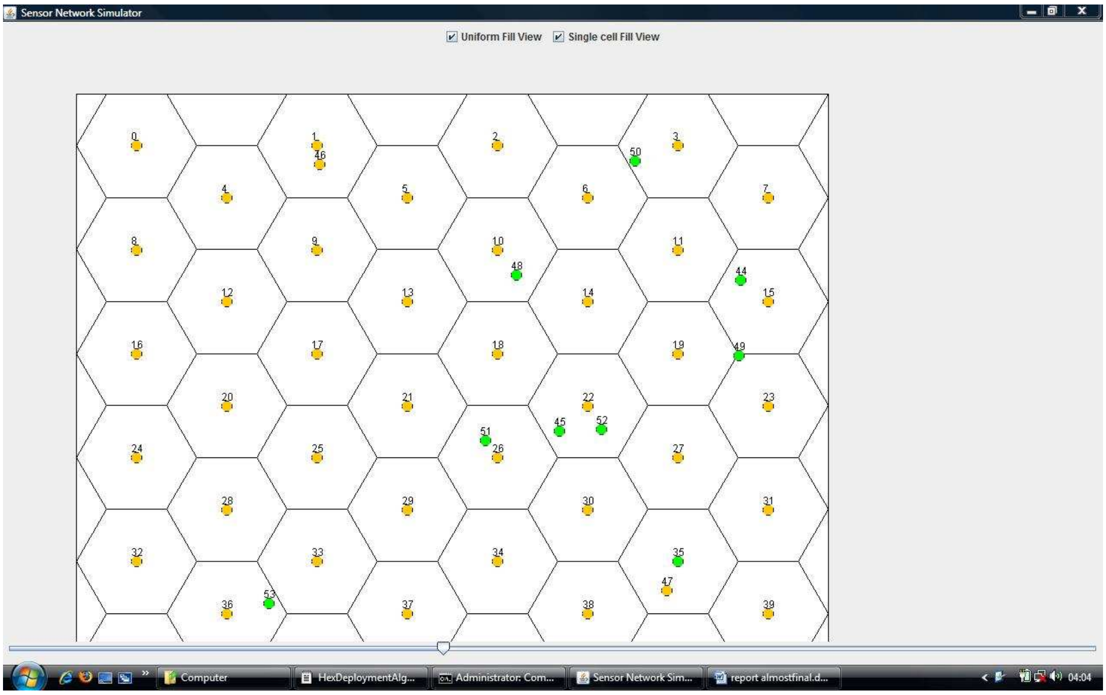
(b) Intermediate stage



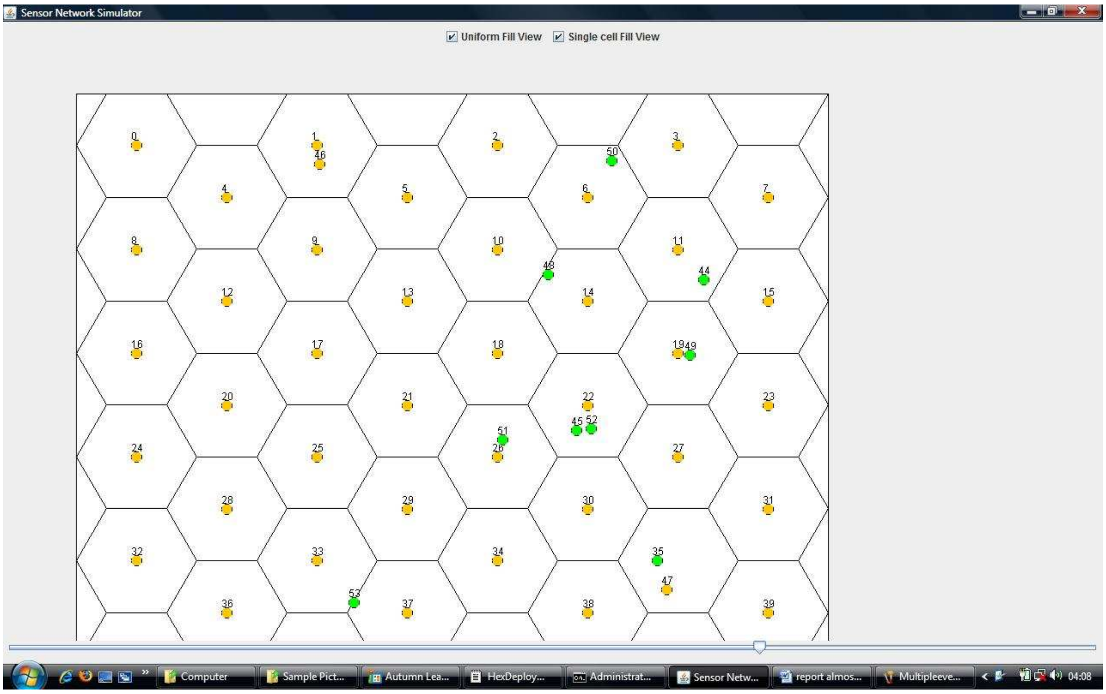
(c) Final stage

Figure 6.4 Single Event Generation

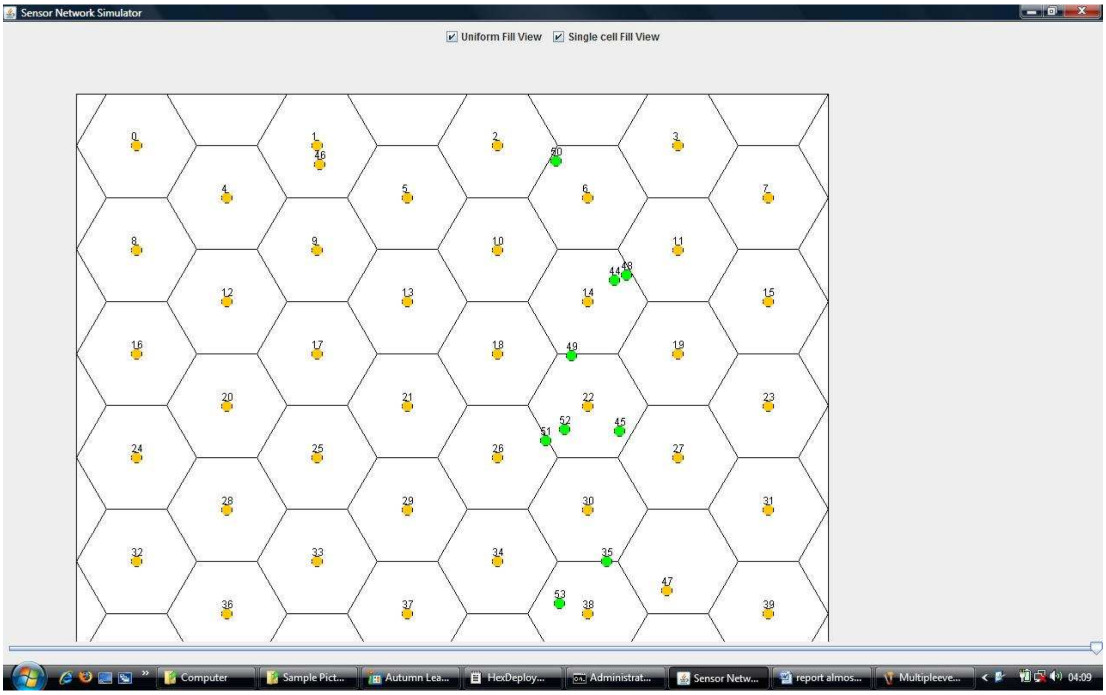
Multiple Event Generation:



(a) Initial Distribution



(b) Intermediate stage



Final stage

Figure 6.5 view of distribution of nodes

Chapter 7

Conclusion

In this project we have looked at the possibility of reusing the knowledge from natural systems in the domain of computer science and self organizing sensor networks. The discussion begins with a survey of sensor networks, its applications and various challenges. We looked at the properties of self organizing sensor networks in the view of covering the given area with mobile wireless sensor nodes. The project addressed one of the fundamental problems of dynamic sensor network which is coverage. Algorithm was proposed and simulated using Java Simulator. The results shown that for a single cell distribution of nodes, the algorithm is taking very large number of iterations where as for the random distribution of nodes it is performing very well.

7.1 Contributions

The contributions of this project can be enumerated as follows:

- Survey on self organizing sensor networks.
- Survey on sensor networks coverage.

- Development of a distributed deployment algorithm for integrating Hexagonal coverage with digital hormone model (DHM).

- Framework for the mobile sensor relocation for tracking multiple events.

- Development of a Java Simulator for validating the proposed deployment algorithm.

7.2 Future Work

Although the current work is complete to a certain extent, the complex and fast developing field of sensor networks provides many challenges and opportunities. The proposed algorithm can be implemented on Cricketmotes and BeeBot mobile platform.

References

[1] Manohar Adari . ”Configuring MobileSensors For Maximizing Event Coverage”, M.Tech Thesis submitted under the guidance of Prof.SantanuChaudhury, Electrical Engineering Department, IIT Delhi.

[2]. S.Indu, Venu Kesam, Santanu Chaudhary and Asok Bhattacharyya, “Self Configuring Deployment Algorithm for Mobile Sensor Network”.

[3] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B.Srivastava, “Coverage Problems in Wireless Ad-hoc Sensor Networks,” In INFOCOM, pages 1380–1387, 2001.

[4] A. Howard, M. J. Mataric, and G. S. Sukhatme, ”An incremental self deployment algorithm for mobile sensor networks,” Autonomous Robots, Special Issue on Intelligent Embedded Systems, September 2002.

[5] Maxim A. Batalin and Gaurav S. Sukhatme, “Spreading Out: A Local Approach to Multi-robot Coverage,” In Proceedings of the 6th International Symposium on Distributed Autonomous Robotics Systems, pages 373–382, June 25–27, 2002.

[6] S. Poduri and G. S. Sukhatme. ” constrained coverage for mobile sensor networks”. In IEEE International Conference on Robotics and Automation, pages 165–172, 2004.

[7] Bin Zhang and G. S. Sukhatme. ” controlling sensor density using mobility”. In Proceedings of the Second IEEE Workshop on Embedded Networked Sensors, pages 141 – 149, 2005.

[8] Z. Bulter and D. Rus, “Controlling Mobile Sensors for Monitoring Events with Coverage Constraints,” In Proc. of the IEEE International Conference on Robotics and Automation, pages 1568–1573, 2004.

[9] M. Batalin and G. S. Sukhatme. ”Coverage, exploration and deployment by a mobile robot and communication network”. Telecommunications Systems Journal, special Issue On Wireless Sensor Networks, 26:181–196, 2004.

[10] F. Aurenhammer ”Voronoi diagrams a survey of a fundamental geometric data structure,” ACM Computing Surveys, vol. 23, pages 345-406, 1991.

[11] G. Wang, G. Cao, and T. L. Porta. ”Movement-assisted sensor deployment”. Proceedings of 23rd International Conference of IEEE Communications Society, Mar 2004.

[12] Mohammad Ilyas, Imad Mahgoub. Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems

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

[14] Deepak Ganesan, Alberto Cerpa, Wei Ye, Yan Yu, Jerry Zhao, and Deborah Estrin. ”Networking Issues in Wireless Sensor Networks,” Journal of Parallel and Distributed Computing (JPDC), Special issue on Frontiers in Distributed Sensor Networks, To Appear, Elsevier Publishers.

[15] Alan Mainwaring, Joseph Polastre, Robert Szewczyk, David Culler, and John Anderson. ”Wireless Sensor Networks for Habitat Monitoring,” Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications, pages 88-97, September 2002

[16] Y. Zou and K. Chakrabarty, ”Sensor deployment and target localization based on virtual forces,” in Proceedings of INFOCOM Systems, March 2003.

[17] Joseph A. Paradiso, Thad Starner. ”‘Energy Scavenging for Mobile and Wireless Electronics”’, IEEE Pervasive Computing, pages 18-27, January 2005.

[18] Weixiong Zhang, Zhidong Deng, Guandong Wang, Lars Wittenburg, and Zhao Xing. ”Distributed Problem Solving in Sensor Networks”, In Proceedings of the first international joint conference on Autonomous agents and multiagent systems, pages 988-989, ACM Press 2002.

[19] K. S. J. Pister, J. M. Kahn, and B.E. Boser. ”Smart dust: Wireless networks of millimeter-scale sensor nodes”, Highlight Article in 1999 Electronics Research Laboratory Research Summary, 1999.

[20] Scott Camazine et al.” Self-Organization in Biological Systems”, Princeton Studies in Complexity, Princeton University Press, 2001.

[21] Y.Yao and G.B. Giannakis.” Energy Efficient Scheduling for Wireless Sensor Networks” IEEE Transactions Communications, Vol. 53, No. 8,August 2005.

[22] S.S. Dhillon, K. Chakrabarty and S. S. Iyengar “Sensor Placement for Grid Coverage under Imprecise Detections” .

[23] M. Rahimi, H. Shah and G.S. Sukhatme “Studying the Feasibility of Energy Harvesting in a Mobile Sensor Network”.

[24] J. Cortes, S. Martinez, T. Karatas and F. Bullo “Coverage control for mobile sensing networks” IEEE Transactions on Robotics and Automation.

[25] Wei-Men Shen, Peter Will and Aram Galstyan Hormone Inspired Self Organization and Distributed Control of Robotic Swarms, Autonomous Robots 17, pp 93-105, Kluwer Academic Publishers, 2004.

[26] Chee-Yee Chong, Srikanta P. Kumar. ” Sensor Networks: Evolution, Opportunities, and Challanges”. Proceedings of the IEEE, Vol. 91, No. 8, August 2003.

[27] Jean-Franois Chamberland, Venugopal V. Veeravalli. Decentralized Detection in Sensor Networks IEEE Transactions On Signal Processing, Vol. 51, No. 2, February 2003.

[28] Ramanarynan Vishwanathan, Pramod K. Varshney.Distributed Detection With Multiple Sensors: Part IFundamentals Proceedings of the IEEE, Vol. 85, No. 1, January 1997.

[29 J. H. Reif and H.Wang, “Social Potential Fields: A Distributed Behavioral Control for Autonomous Robots,” Robotics and Autonomous Systems, vol. 27, pages 171–194, 1999.

[30] Holger Karl, Andreas Willig. ” Protocols and Architectures for Wireless Sensor Networks”. John Wiley & Sons, 36–44, 2005.

[31] C. Y. Chang and C. T. Chang. ” Hierarchical Cellular-Based Management for Mobile Hosts in Wireless Ad-Hoc Networks”. Computer Communications, Vol. 24, pages 1554–1567, 2001.

[32] P. Eugster, P. Felber, R. Guerraoui, and A. Kermarrec, “The Many Faces of Publish/Subscribe,” ACM Computing Surveys, vol. 35, no. 2, pp. 114–131, June 2003.

[33] W. Zhang, G. Cao and T. La Porta, “Data Dissemination with Ring-Based Index for Wireless Sensor Networks,” The 11th IEEE International Conference on Network Protocols (ICNP), November 2003.