People's Democratic Republic of Algeria
Ministry of Highly Education and Scientific Research

جامعة بجاية
Tasdawit n'Bgayet
Université de Béjaïa

A.MIRA-BEJAIA University
Faculty of Exact Sciences
Computer Science Department

Dissertation

Presented by

BELHADJ Asma

Submitted in Partial Fulfillment of the Requirement for Magister Degree

Branch: Computer Science

Option: Cloud Computing

Theme

Area Coverage in wireless sensor network with presence of mobile nodes

Supported on: 10/03/2016

In Front to the Jury composed from:

Last/First Name	**Grade**		
Mr. MENDIL Boubekeur	Professor	Univ. of Bejaïa	President
Mr. BEGHDAD Rachid	Professor	Univ. of Bejaïa	Supervisor
Mr. IDOUGHI Djilali	M.C.A	Univ. of Bejaïa	Examiner

University year: 2015/2016

Acknowledgement

We first thank Allah the Almighty and merciful God, who gave us the strength and patience to accomplish this modest work.

We would like to thank my Supervisor: Dr Rachid BEGHDAD for his guidance, countless hours of reflecting, reading, encouraging, and most of all patience throughout the entire process and during my dissertation work.

We wish to thank my committee members, Mr. MENDIL Boubekeur, Mr. BOUBETRA Abdelhak and Mr. IDOUGHI Djilali who were more than generous with their expertise and precious time, and who offered guidance and support.

Thanks to the University of Bejaia for awarding me a Dissertation Completion Fellowship, providing me with all means to complete this research project.

Special thanks go to our professors, and to administrators in the computer science department and in the faculty of exact sciences.

I would also like to thank my family for their support, patience, and understanding during this labor when I had to be “working on my Dissertation”. A very respectful thanks to numerous friends, to their presence, who endure this long process with me, always offering support and help.

Finally, This Work would not have been possible without the support of many people made the completion of this research an enjoyable experience.

We also wish to thank all those who participated directly or indirectly to the realization of this work.

Dedication

To my dear parents,

To my brother and my sisters,

To my friends and colleagues,

To all my teachers,

I dedicate this modest work.

Résumé

Plusieurs algorithmes ont été proposés dans la littérature pour résoudre le problème de la couverture dans les RCSF, qui sont basées sur des stratégies différentes. Pour un réseau contenant des nœuds mobiles, la stratégie de mouvement de nœud a été utilisé où l'objectif est de tenir compte de nœuds mobiles pour contrôler la couverture du réseau. Les nœuds mobiles changent les caractéristiques de couverture du réseau via ce déplacement vers les emplacements désirées.

Ici, on désigne un nouvel algorithme distribué basé sur la méthode de mouvement sur grille appelé CBNMS (*Coverage Based Node Movement Strategy*) afin d'assurer les exigences de la couverture et de la connectivité. L'algorithme divis la zone cible en m × m grilles de carrés en utilisant les concepts mathématique de l'espace de Hilbert, les capteurs dispersés densément et uniformément en grilles; et un header de grille doit sélectionner par grille comme un super-nœud placé dans chaque grille.

Notre méthode de mobilité permet de déplacer les nœuds d'une grille de haute densité vers celle de faible densité ; Nœuds de capteurs se déplacent à travers les sous-carrés de Hilbert avec un chemin limité. Notre solution proposée est localisée. Aucune découverte du voisinage n'est nécessaire. Aucune connaissance préalable sur les positions des nœuds. La zone cible plane sans obstacles.

Les résultats de simulation montrent que CBNMS atteint un taux de couverture élevé en comparaison à certains protocoles existants.

Mots-clés: *Réseaux de capteurs sans fil, couverture, connectivité, mobilité, approche basée sur les grilles, espace de Hilbert.*

Abstract

Several algorithms were proposed in the literature to solve the coverage problem in WSN, which are based on different strategies. For network containing mobile nodes, node movement strategy has been used where the objective is to leverage mobile nodes to control network coverage. Mobile nodes change network coverage characteristics via moving to the desired locations.

Herein we design a novel distributed algorithm based on square-grid movement method called CBNMS (*Coverage Based Node Movement Strategy*) in order to ensure the coverage and connectivity requirement. The algorithm divides target areas into m × m square grids using the concept of space-filling curve with scattered sensors densely and uniformly into grids; and selects a grid header in each grid as a super-node placed in each grid.

Our movement method consists of migrating nodes from high density to low density grid; Sensor Nodes move within their Hilbert sub-squares with limited distance. Our proposed solution is localized. No neighbor discovery is needed. No previous knowledge about nodes positions. The plane target area with no obstacles.

Simulation results show that CBNMS achieves a high coverage ratio in comparison to some existing protocols.

Keywords: *Wireless Sensor Networks, Coverage, Connectivity, Mobility, grid-based approach, Hilbert space-filling.*

ملخص

توجد عدة خوارزميات مقترحة من أجل حل مشكلة التغطية في شبكات الاستشعار اللاسلكية ، والتي تقوم على استراتيجيات مختلفة. من أجل شبكة تحتوي على عقد متنقلة، تستخدم استراتيجية تحريك العقد حيث تهدف لاستغلال العقد المتنقلة لضبط تغطية الشبكة. العقد المتنقلة تغير من خصائص تغطية الشبكة وذلك عبر تنقلها إلى الأماكن المطلوبة.

في هذه المذكرة، قمنا بتصميم خوارزمية ترتكز على طريقة الحركة في شبكة مربعات ، تسمى CBNMS (*Coverage Based Node Movement Strategy*) من أجل ضمان متطلبات الربط و التغطية . الخوارزمية تقسم المناطق المستهدفة إلى م × م شبكات مربعة باستخدام مفهوم فضاء هيلبرت مع نشر أجهزة الاستشعار بكثافة وبشكل منظم ما بين الشبكات المربعة. مع اختيار مسؤول في كل شبكة مربعة بمثابة عقدة متخصصة.

طريقة حركة العقد ما بين مربعات شبكة هيلبرت تكون من مربع الشبكة الاكثر كثافة إلى الاقل كثافة و بمسافة محدودة. الحل المقترح محلي. ليس هناك حاجة إلى اكتشاف الجار. لا معرفة سابقة عن مكان العقد. المنطقة المستهدفة من دون عقبات.

أظهرت النتائج التجريبية لـ CBNMS نسبة تغطية عالية بالمقارنة مع بعض البروتوكولات الموجودة.

كلمات البحث: *شبكات الاستشعار اللاسلكية، التغطية ، الربط، التنقل، طريقة مربع الشبكات، فضاء هلبرت.*

Acronym's List

ADC	**A**nalogue-to-**D**igital **C**onverters
ASIC	**A**pplication-**S**pecific **I**ntegrated **C**ircuit
BCP	**B**order **C**overage **P**rotocol
BS	**B**ase **S**tation
CBNMS	**C**overage Based **N**ode **M**ovement **S**trategy
CCSID	**C**onnected **C**over **S**et based on **ID**entity of nodes
CDSC	**C**entralized **D**ominating **S**et for **C**overage
CH	**C**luster **H**ead
CPU	**C**entral **P**rocessor **U**nit
CSD	**C**ritical **S**ensor **D**ensity
DCovPDS	**D**istributed **Cov**erage **P**reserving based on **D**ominating **S**et
GPS	**G**lobal **P**ositioning **S**ystem
LLC	**L**ogical **L**ink **C**ontrol
MAC	**M**edia **A**ccess **C**ontrol
MCDS	**M**inimum **C**onnected **D**ominating **S**et
MULE	**M**obile **U**biquitous **L**AN **E**xtensions
OGDC	**O**ptimal **G**eographical **D**ensity **C**ontrol
QoS	**Q**uality **o**f **S**ervice
RN	**R**endezvous **N**ode
SMART	**S**can-based **M**ovement-**A**ssisted senso**R** deploymen**T**
US	**U**nit **S**quare
VFA	**V**irtual **F**orce **A**lgorithm
WSN	**W**ireless **S**ensor **N**etwork
ZI	**Z**one of **I**nterest

Table of Contents

Chapter4: *Simulation and performance Analysis*

Figures

Figures

Tables

General Introduction

Sensors are devices that convert physical stimulus into recordable signals. Sensors have facilitated people to understand, monitor, and control machines and environments for many decades. A sensor node consists of not only sensor unit but also microcontroller unit, communication unit, storage unit, and power supply for producing, collecting, storing, processing, and delivering sensory data. The size and cost of a single sensor node has been reducing with the continuous advances of micro-electro-mechanical systems (MEMS) techniques. The miniaturization of sensor nodes has promoted the emergence of sensor networks, which normally consists of a large number of sensor nodes collaborating to accomplish advanced tasks.

Sensor nodes are usually deployed in a field of interests to monitor some physical phenomena. The nodes have constrained batteries which cannot be recharged or replaced after they are deployed over networks. In the environment such as batter areas, harsh regions and disaster areas, we cannot deploy the nodes manually. Thus the nodes need randomly to be scattered by vehicles such as aircraft [**1**], [**2**]. Such a field of interests is called a sensor field, and the sensor nodes form a sensor network. Diverse scenarios exist in the architecture and management of sensor net-works. Numerous sensor network applications are expected to emerge in the near future. Applications of sensor networks are in a wide range, including battlefield surveillance, environmental monitoring, biological detection, smart space, industrial diagnostics, etc. Despite promising applications, there are also great challenges in designing, implementing, and operating sensor networks. Many research issues have been studied, and many solution approaches have been proposed for sensor networks [**3**].

One of the most active research fields in wireless sensor networks is that of coverage. Coverage is usually interpreted as how well a sensor network will monitor a field of interest. It can be thought of as a measure of quality of service. Coverage can be measured in different ways depending on the application. In addition to coverage it is important for a sensor network to maintain connectivity. Each node has a communication range which defines the area in which another node can be located in order to receive data. This is separate from the sensing range which defines the area a node can observe. The two ranges may be equal but are often different [**4**].

In wireless sensor networks, all nodes share common sensing tasks, which imply that not all sensors are required to perform the sensing tasks during the whole system lifetime. Making some nodes sleep does not affect the overall system function as long as there are enough working nodes to assure it. Therefore, if we initially deploy a large number of sensors and schedule them to work alternatively, system lifetime can be prolonged correspondingly, i.e., redundancy is exploited to increase system lifetime.

For sensor networks containing mobile nodes, the node movement strategy has been used where the objective is to leverage mobile nodes to control network coverage. Mobile nodes change network coverage characteristics via moving to the desired locations. Although network coverage can be greatly improved, the product cost and the moving cost of mobile nodes need to be minimized. The design of node movement strategy should balance between network coverage and movement cost.

In this dissertation we interest about mobile sensor network, one of the movement objectives is to maximize area coverage. We design and evaluate a novel distributed algorithm for mobile sensor networks to maximize the network coverage; called *Coverage Based Movement Strategy* “CBNMS”. Nodes are uniformly scattered in the initial deployment. In such environment, the algorithm divides the target areas into square grids, which have a grid header respectively. These grids are obtained using the concept of space-filling curves, with scattered sensors densely and uniformly into grids; and selects a grid header in each grid as a super-node placed in each grid. The grid header is responsible for controlling and communicating nodes located in its region, in order to ensure that only one node per region is required to be active during a round. To ensure a whole coverage of the covered area, where at the same time connectivity is maintained, a specific Hilbert order is chosen for Hilbert trajectory.

Only few related works use of space filling curves to solve the problem of efficient coverage of the monitored in networks composed of mobile sensors. More related to the use of space-filling curves for coverage in wireless sensor networks are the works [**5**,**6**,**7**],all these works use the space filling curves with sinks and relays mobility (Sink, collectors or single mobile). While herein we interest about *sensor mobility* in mobile WSN and their impact on coverage and network topology.

For our coverage purpose, only one node per US is required to be active during a round (e.g., one node lifetime) to ensure a whole coverage of the covered area, where at the same time connectivity is maintained. We use a scheduling activity to schedule the activation and deactivation of nodes’ sensor units, based on the selection of an appropriate subset of sensor nodes that must remain active; such that each cell reach the center of the grid to achieve optimal precision and accuracy of network coverage .

Moreover; our algorithm uses a square grid-based movement strategy in order to leverage mobile nodes to enhance network coverage; it consists of migrating nodes from high density to low density grid. Sensor Nodes move within their Hilbert sub-squares with limited distance; to reduce the movements of sensor nodes to save the energy which results in increasing network lifetime. No previous knowledge about nodes positions, all sensor nodes are not equipped with GPS system. We assume that every Rendezvous node RN is equipped with GPS system. Each RN stores the distance vector estimated during localization phase between each node and h-keys of its square grid. That is, every node has the ability to know its own location and the number of deployed nodes in Neighbor's grid by requesting their RN. Our algorithm assumed a plane target area with no obstacle. In most of the existing work researcher assumed a target field with no obstacle. Our proposed solution is localized, and asynchronous. No neighbor discovery is needed. No previous knowledge about nodes positions.

This Dissertation is organized in four chapters:

Chapter 1: In this chapter, we provide some backgrounds and introduction about sensor nodes, and sensor networks. Then we present coverage problem formulation, their objectives, coverage taxonomy, the main types of coverage and their design choices.

Chapter 2: gives the related work. A review of research that has already been done that relates to our topic will be covered. This will cover a review of papers that deal with the sensing coverage of wireless sensor networks. We describe different coverage algorithms in WSN, and synthesis of different area coverage protocols of interest in static and mobile WSN.

Chapter 3: a discussion of our coverage protocol, giving the detailed description of the proposed solution within both static and mobile sensor networks, in order to see the impact of mobility to maximize the network area coverage.

Chapter 4: Finally, following Java programming language, we present the experimental results and performance evaluations of our protocol, this by comparing it with different existing area coverage solutions.

At the end; lessons learned and conclusions to be drawn from the research and future work which is given.

Chapter 1

Introduction of Coverage Problem in WSN

1.1. Introduction

Sensors are devices that convert physical stimulus into recordable signals. Sensors have facilitated people to understand, monitor, and control machines and environments for many decades. Sensor nodes are usually deployed in a field of interests to monitor some physical phenomena. Such a field is called a sensor field, and the sensor nodes form a sensor network. Diverse scenarios exist in the architecture and management of sensor networks. Numerous sensor network applications are expected to emerge in the near future. Many research issues have been studied, and many solution approaches have been proposed for sensor networks.

Coverage problems have become an important research topic in wireless sensor networks in recent years. They are addressing the following fundamental question*: how well do the sensors observe the physical space?* Many coverage problems have been addressed by the research literature due to a variety of sensors and applications. They vary on the subject to be covered (area, discrete points), sensor deployment mechanism (random, deterministic), sensor mobility (stationary, mobile), event structure (simple, composite), and network properties (connectivity, energy efficiency.

In this chapter, we provide some backgrounds and introduction about sensor nodes, and sensor networks. Then we present coverage problem formulation, their objectives, coverage taxonomy, the main types of coverage and their design choices.

1.2. Sensor Nodes Architecture

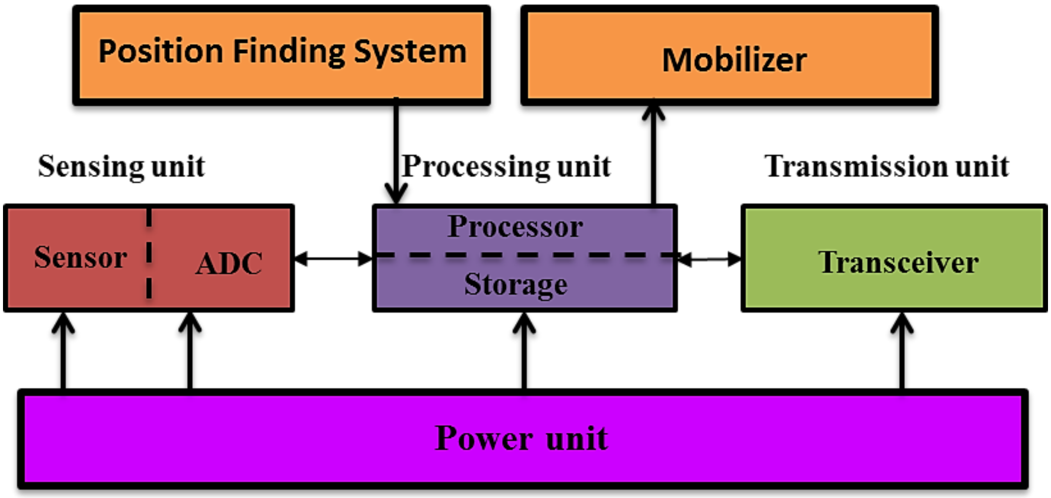

Figure 1. 1: A typical sensor node hardware architecture [8]

Generally speaking, a sensor is a device which responds to physical stimulus (such as heat, light, sound, pressure, magnetism, etc.) and converts the quantity or parameter of a physical stimulus into recordable signals (such as electrical signals, mechanical signals, etc.) [**3**]. Sensor nodes are tiny devices equipped with one or more sensors, one or more transceivers, processing and storage resources, and, possibly, actuators. Sensor nodes have limited resources: they have finite battery resources, low CPU speed, little memory, and small transmission range.A typical architecture of a sensor node is shown in Figure 1. 1, which consists of sensor unit, communication unit, microcontroller unit, and memory and power unit.

A sensor node may also include some other units, depending on application scenarios and requirements, such as a locomotive unit that enables a sensor node to move around, an energy scavenge unit that obtains energy from the physical world, a GPS unit that acquires node geographical location, etc [**3**].

1.3. Sensor Networks

The wireless sensor network (WSN) is a special type of Ad-hoc network where the communication infrastructure and centralized management are absent.

Sensor nodes are usually deployed in a field of interests to monitor some physical phenomena. Such a field is called a sensor field, and the sensor nodes form a sensor network. Diverse scenarios exist in the architecture and management of sensor networks. Numerous sensor network applications are expected to emerge in the near future [**3**].

A wireless sensor network (WSN) in its simplest form can be defined as [**2**] a network of (possibly low-size and low-complex) devices denoted as nodes that can sense the environment and communicate the information gathered from the monitored field (e.g., an area or volume) through wireless links; the data is forwarded, possibly via multiple hops

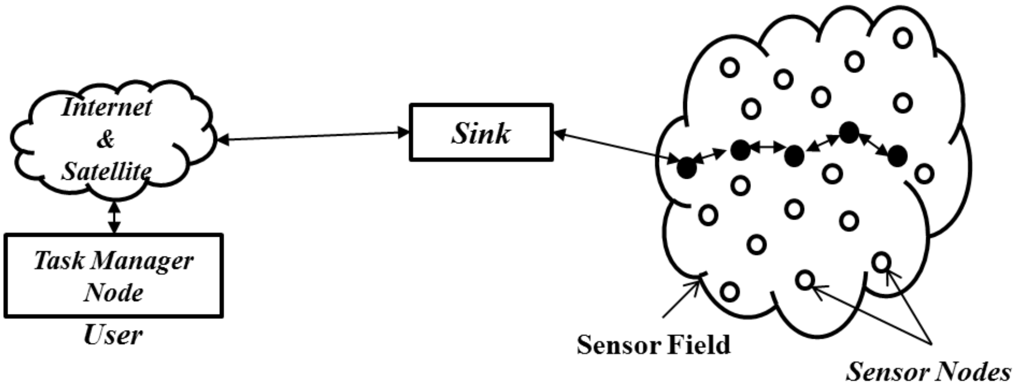

Figure 1. 2: Architecture of a sensor network Example [2]

relaying, to a sink(sometimes denoted as controller or monitor) that can use it locally, or is connected to other networks (e.g., the Internet) through a gateway.

The nodes can be stationary or moving. They can be aware of their location or not. They can be homogeneous or not. This is a traditional single-sink WSN (see Figure 1. 2).

A sink is often assumed as resource abundant, without energy supply limit and with advanced computation capability. Aside from data collection, a sink is also a central controller which can execute network management algorithms and instruct sensor nodes only the computation results [**3**].

1.4. Sensor Network Applications

As almost all kinds of physical phenomena can be sensed by different types of sensors, it is not surprised to find applications of sensor networks in almost any environment. In what follows, B. Wang [**3**] present a brief introduction and example applications of sensor networks.

1.4.1. Military Applications

Sensor networks can be integrated into the military command, control, communications, and computing systems. Wireless sensor networks, which can be rapidly scattered to critical terrains, routes, paths, and straits, can provide battlefield intelligence, such as the location, numbers, movement, and identity of troops and vehicles, and can be used to detect, classify, - and track the trajectory of enemy vehicles.

1.4.2. Environmental Applications

Some environmental applications of sensor net-works include tracking the habitat of birds and animals, observing bio-diversity and bio-complexity, monitoring environmental conditions for precision agriculture, detecting disasters such as forest fire detection and flood detection, studying air pollution, etc.

1.4.3. Industrial Applications

Sensor networks can also find many industrial applications, such as manufacture automation, warehouses and products management, automated reading for gas, water, and electric meters, lighting control, smoke and nocuous gas detection, etc. Sensor networks can also be used to monitor the structural health of building, bridges, and roads and to check the physical condition of water and gas pipes. At present, structures are monitored primarily through manual inspections. Untethered sensors can help to automate the monitoring process by providing timely and detailed information about incipient cracks or about other structural damage.

1.4.4. Home Applications

Sensor networks can help to create a smart environment. With the advances of technology, tiny yet smart sensors can be buried in appliances, such as vacuum cleaners, micro-wave ovens, and refrigerators. These sensors inside the domestic devices can interact with each other and with external networks such as the Internet. This allows users to manage their devices locally and remotely.

1.4.5. Medical Applications

Physical sensors and bio-sensors can monitor body temperature, pulse rate, perspiration, oxygen and glucose level in blood, and other physical and biochemical parameters. These sensors can be wear in clothes or even implanted into human bodies. A wireless body sensor network integrates these wear-able or implanted sensor nodes into a network and connects them with external networks for monitoring patients with chronic disease, hospital patients, and elderly patients.

1.5. Sensor Network Challenges

Several architectures and design goals/constraints have been considered to adapt WSNs to the context defined by the related applications. In what follows, we briefly discuss the most important challenges in sensor networks.

1.5.1. Deployment

The deployment of sensor nodes in the physical environment may take several forms. Nodes may be deployed at random (e.g., by dropping them from an aircraft) or installed manually. Deployment may be a one-time activity, where the installation and use of a sensor network are strictly separate activities. However, deployment may also be a continuous process, with more nodes being deployed at any time during the use of the network for example, to replace failed nodes or to improve coverage at certain interesting locations [**9**].

1.5.2. Mobility

In a sensor network [**9**], three types of mobility can be distinguished:

- ✓ *Event mobility:* Typical in tracking applications (gathering information about target like speed, direction, and size of an object). For this, we need the collaboration of many sensors.
- ✓ *Node mobility:* Sensor nodes may change their location after initial deployment. Mobility can result from environmental in influences such as wind or water, sensor

nodes may be attached to or carried by mobile entities, and sensor nodes may have the capability to adjust their position to over better results for the task.

- ✓ *Sink mobility:* the requester of information from a sensor network can be mobile when the sink node is an external element like a PDA or Laptop.

1.5.3. Cost, Size, Resources, and Energy

Varying size and cost constraints directly result in corresponding varying limits on the energy available (i.e., size, cost, and energy density of batteries or devices for energy scavenging), as well as on computing, storage, and communication resources. Hence, the energy and other resources available on a sensor node may also vary greatly from system to system [**9**]. These resource constraints limit the complexity of the software executed on sensor nodes.

1.5.4. Heterogeneity

A sensor network can be non-hierarchical or at in the sense that every sensor has the same detection and tracking, some sensors may be designated as the fusion centers: they collect the reports from the sensors in their neighborhood, make a decision regarding whether an object has been detected, and send a report to the base station. Many prototypical systems consist of a variety of different devices. Nodes may differ in the type and number of attached sensors: some computationally powerful nodes, limited sensing nodes, some localized nodes equipped with special hardware such as a GPS receiver ; some nodes may act as gateways to long-range data communication networks . The degree of heterogeneity in a sensor network is an important factor that affects the complexity of the sensor's software [**9**].

1.5.5. Network Topology

One important property of a sensor network is its diameter, that is, the maximum number of hops between any two nodes in the network. In its simplest form, a sensor network forms:

- ✓ A *single*-hop network, with every sensor node being able to directly communicate with every other node. An infrastructure-based network with a single base station forms a star network with a diameter of two.
- ✓ A *multi*-hop network may form an arbitrary graph, but often an overlay network with a simpler structure is constructed such as a tree or a set of connected stars.

The topology affects many network characteristics such as latency, robustness, and capacity. The complexity of data routing and processing also depends on the topology [**9**].

1.5.6. Network Autonomy

In a remote or dangerous field, randomly scattering nodes might be the only way to deploy a sensor network. In such cases, the untended nodes should self-organize into an autonomous network to decide the structure and topology of the network. Such an autonomous network should be able to schedule sensing tasks and to arrange delivery routes all by itself. It is also required that an autonomous sensor network should be able to monitor its own health and status and to adapt its operational parameters in different situations. Sometimes, a sensor network should also be able to interact with external maintenance mechanisms or external network interfaces for better operation [**3**].

1.5.7. Coverage

The effective range of the sensors attached to a sensor node defines the coverage area of a sensor node. Network coverage measures the degree of coverage of the area of interest by sensor nodes.

- ✓ With sparse coverage, only parts of the area of interest are covered by the sensor nodes.
- ✓ With dense coverage, the area of interest is completely (or almost completely) covered by sensors.
- ✓ With redundant coverage, multiple sensors cover the same physical location.

The actual degree of coverage is mainly determined by the observation accuracy and redundancy required. High coverage is a key to robust systems and may be exploited to extend the network lifetime by switching redundant nodes to power-saving sleep modes [**9**].

1.5.8. Connectivity

The communication ranges and physical locations of individual sensor nodes dene the connectivity of a network. If there is always a network connection (possibly over multiple hops) between any two nodes, the network is said to be connected.

Connectivity is intermittent if the network may be occasionally partitioned. If nodes are isolated most of the time and enter the communication range of other nodes only occasionally, we say that communication is sporadic. Note that despite the existence of partitions, messages may be transported across partitions by mobile nodes. Connectivity mainly in influences the design of communication protocols and methods of data gathering [**9**].

1.5.9. Network Size

The number of nodes participating in a sensor network is mainly determined by requirements relating to network connectivity and coverage, and by the size of the area of interest. The network size may vary from a few nodes to thousands of sensor nodes or even more. The network size determines the scalability requirements with regard to protocols and algorithms [**9**].

1.5.10. Lifetime

Depending on the application, the required lifetime of a sensor network may range from some hours to several years. The necessary lifetime has a high impact on the required degree of energy efficiency and robustness of the nodes [**9**].

1.5.11. Information Security

Information security, which is a basic yet common requirement in almost all types of networks, requires that sensing data should be accessed, transmitted, and processed securely and privately. However, security algorithms are usually resource demanding and should be modified to adapt to resource-constraint sensor networks [**10**].

1.6. Wireless Sensor Network with mobile elements

Wireless Sensor Network (WSN) is a collection of sensor nodes able to sense their environment, collect and process various data, and communicate among each other. The introduction of mobility of sensor nodes can significantly affect and improve the overall network performances. Sensor nodes may move by self-driving (e.g. Mounted on wheels) or by being attached to transporting devices (e.g. robots, people, vehicles, or animals) resulting in longer network lifetime, better area coverage, and dynamic adaptation to different system functionalities and requirements [**8**].

Each sensor node in the WSNs comprises sensing unit, transceiver, processing unit, and power supply (usually battery) as mentioned in sensor node architecture above. If mobile, sensor nodes can be equipped with various locomotion devices and, additionally, they may contain different position finding systems.

Mobile WSNs may include sensor nodes, actuators, relays, and sinks as shown in Figure 1. 3.The main components of WSN with mobile elements as in [**11**] WSN-MEs are the following:

- ***Sensor nodes:*** are the sources of information. Such nodes perform-sensing as their main task. They may also forward or relay messages in the network, depending on the adopted communication paradigm.
- ***Sinks (base stations or central Gateway):*** are the destinations of information. They collect data sensed by sensor nodes either directly (i.e., by visiting sensors and collecting

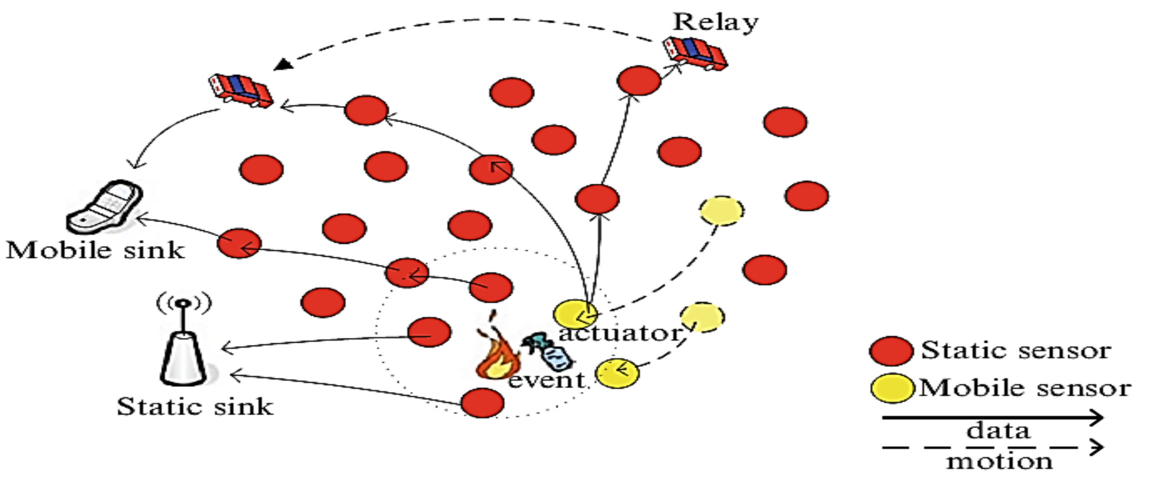

Figure 1. 3: WSN, mobile entities and possible reaction to events [8**]**

- data from each of them) or indirectly (i.e., through intermediate nodes). They can use data coming from sensors autonomously or make them available to interested users through an Internet connection.
- ***Special support nodes:*** perform a specific task, such as acting as intermediate data collectors or mobile gateways. They are not sources nor destinations of messages, but exploit mobility to support network operation or data collection .We distinguish :
  - ✓ *Mobile relays* can inherit other sensors to improve network connectivity. Relays usually forward the information from the sensors to the so called sink (a data collector and a possible gateway to the backbone network)
  - ✓ *Actuators* are nodes that may dynamically act upon received information, both on sensors and on environment.

1.6.1. Taxonomy of Mobility in WSN

Mobility has a large impact on the expected degree of network dynamics and hence influences the design of net-working protocols and distributed algorithms. The actual speed of movement may also have an impact, for example on the amount of time during which nodes stay within communication range of each other [**9**].

Mobility includes different functionality in wireless sensor network like coverage optimization, better lifetime of network, better use of resources and relocation . Sensor nodes may change their location after initial deployment. Mobility may apply to all nodes or only to subsets of nodes. M. Marks [**12**] give taxonomy of mobility as shown in Figure 1. 4

1.6.1.1. Sensor network topology

Somme applications introduce an even greater degree of dynamism, due to the need to support physically mobile devices. Mobility may (or may not) manifest itself in different ways. Mottola and Picco [**13**] distinguish three classes static, mobile nodes and mobile sinks.

- In static applications neither nodes nor sinks move once deployed. This is by far the most common case in current deployments.

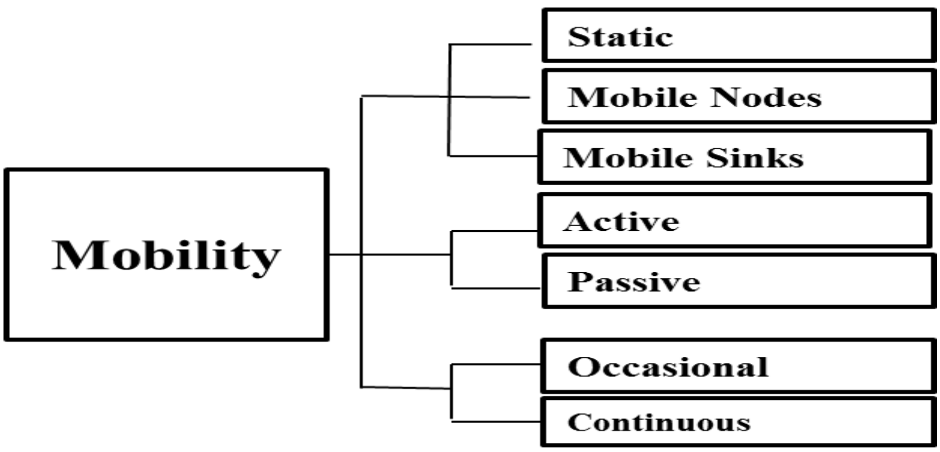

Figure 1. 4: Taxonomy of Mobility [12**]**

- Some applications use mobile nodes attached to mobile entities (e.g., robots or animals) or able to move autonomously.
- Some applications exploit mobile sinks. The nodes may be indifferently static or mobile: the key aspect is that data collection is performed opportunistically when the sink moves in proximity of the sensors.

1.6.1.2. Active vs. passive mobility

Mobility may be either active (i.e., automotive) or passive (e.g., attached to a moving object not under the control of the sensor node). In other words, Mobility may be active or passive. In active mobility the sensors are intelligent enough to find their path and move while in passive mobility the sensors may move by human or environmental assistance.

1.6.1.3. The degree of mobility

Mobility may apply to all nodes within a network or only to subsets of nodes. The degree of mobility may also vary from occasional movement with long periods of immobility in between, to constant travel [**9**].

1.6.2. Mobility Types

The mobile entities in WSN could be acted as a mobile BS, mobile sensor, mobile relay, actuators, or mobile clusterhead, according to the desirable role. These mobile units can be introduced naturally or placed artificially. The mobility pattern of each mobile entity is typically determined based on specific application and the WSN size. S. Misra et al. [**14**] categorize mobile units into three types:

1.6.2.1. Controllable mobility

They follow some predefined trajectories such as mobile robots that the network planner can program them to meet the requirements. An example of this is a TagBot , which is designed by the Carnegie Mellon University. TagBot is an advanced robot that can communicate with sensor motes like MicaZ or Telos. It can move both forward and backward, and turn in any direction by a controlling program, which is resided in an Intel’s board. Robomotes are small in size. Robomote can be equipped with a solar panel for recharging the batter y and move controllably as designed by a programmer.

1.6.2.2. Uncontrollable and unpredictable mobility

They move in a random fashion such that the next movement cannot be predicted. For example, the movement of an animal or human, which carries a sensor, is generally considered in this category. For example, if the sensor is mounted on an elephant in Africa in finding its group behavior, the mobility of sensor is random as the elephant moves.

1.6.2.3. Uncontrollable but predictable mobility

They are like bus or train that move according to a predefined schedule. Therefore, the movement of the sensor carried in the bus or train is usually not random and follows a predetermined path However, they cannot be controlled by the sensor itself. For example, the movement path of a bus to collect sensor data may not be the best routine for WSN performance.

1.7. Sensor Coverage Problem

An important topic researched in literature is the sensor coverage problem. This problem addresses an important question: how well do the sensors observe the physical space? As pointed out in [**10**], the coverage concept is a measure of the quality of service (QoS) of the sensing function and is subject to a wide range of interpretations due to a large variety of sensors and applications. The goal is to have each location in the physical space of interest within the sensing range of at least one sensor [**10**].

Network coverage refers to the coverage relation between field-wide points and network-wide sensors. The main functionalities for network coverage is the coverage characteristics of sensor networks, which include the coverage degree and coverage ratio of the space points, as well as the coverage relation between space points and sensor nodes. Network coverage can be executed before or after network deployment and can also be implemented by centralized or distributed algorithms [**3**]. Coverage can be measured in different ways depending on the application [**4**].

1.7.1. Motivations and Objectives

The network coverage refers to the coverage relation between field-wide points and network-wide sensors. Sometimes, we may regard network coverage as a collective measure of the quality of service provided by a network of sensor nodes at different geographical locations. The sensor unit of a sensor node, which is to perform the sensing task and to produce sensing data, is controllable to be active or inactive. An active sensor node consumes energy to generate sensing data, and an inactive (sleep) sensor node does not generate any sensing data. We use network coverage control to refer to the network-wide control of individual nodes' sensor unit [**3**].

The fundamental motivation of network coverage control, which is also its ultimate objective, can be boiled down to the energy efficiency. A sensor node, which is normally designed for some specific sensing, is in general with small size, low weight, and limited non-rechargeable battery energy.

The study of the network coverage control can help reducing network setup costs. For small-scale networks where sensor nodes can be manually placed at the desired locations, the quality of the sensing task can be guaranteed and the network cost can be minimized. For large-scale network where sensor nodes are normally randomly scattered within the sensor field, network coverage control usually refers to determining the minimum number (cost) of sensor nodes to provide the required coverage requirements. The sensor unit of a sensor node, which is to perform the sensing task and to produce sensing data, is controllable to be active or inactive (sleep). An active sensor node consumes energy to generate sensing data, and an inactive sensor node does not generate any sensing data. Other consequent benefits from the reduced network traffic also include the reduced power consumption for data transmission, the reduced transmission collisions, and the reduced data delivery delay. Therefore, in order to conserve energy and prolong network lifetime, we need to control which sensor to be active (to cover some space points) and for how long [**3**].

For sensor networks containing mobile nodes, mobile nodes change network coverage characteristics via moving to the desired locations. Although network coverage can be greatly improved, the product cost and the moving cost of mobile nodes need to be minimized.

As a short summary, the motivations and objectives for network coverage control can be summarized as to reduce network setup and moving costs, conserve node energy consumption, and prolong network lifetime while guaranteeing the specified coverage requirements.

1.7.2. Coverage concepts

The solutions to coverage and connectivity issues in WSNs involve a lot of basic theories and assumptions. The basic knowledge of coverage concepts is essential. In this section, we describe sensor node properties, Network model, sensing models, centralized/distributed algorithms, and the evaluation indicators of coverage quality.

1.7.2.1. Coverage and communication areas of a sensor node

A sensor node can monitor an area called coverage area; this area is usually called Sensing Range and represented as a disk of radius SR. A node is able to detect any event is occurring in its coverage area.

On the other hand, the vision of a sensor depends on the reception' radius of its communication module CR. A node cannot communicate with a second unless this latter is in its communication range, it means if the Euclidean distance between the two nodes is smaller or equal to CR.

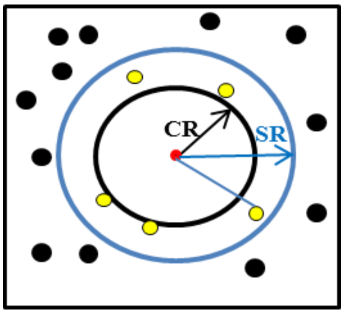

Figure 1. 5: Coverage area of a sensor node [15**]**

We say that a network is connected, whether a given node can communicate with any other node either directly or via other nodes.

Figure 5.1 illustrates an example of coverage and communication area of a sensor. Often we consider the communication range CR larger than the coverage range SR. The coverage of the area of interest consists of the union of all the coverage area of network nodes [**15**].

1.7.2.2. Relationship between coverage and connectivity

In addition to coverage it is important for a sensor network to maintain connectivity which determines the effective transmission of data [**16**]. Connectivity can be defined as the ability of the sensor nodes to reach the data sink. If there is no available route from a sensor node to the data sink then the data collected by that node cannot be processed. Each node has a communication range which defines the area in which another node can be located in order to receive data. This is separate from the sensing range which defines the area a node can observe. The two ranges may be equal but are often different [**4**].

Hence, how to combine consideration of coverage and connectivity maintenance in a single activity scheduling is essential. The target of research on relationship between coverage and connectivity is to select the least number of active nodes, while preserve coverage and maintain connectivity [**16**]. To the best of our knowledge, Zhang [**17**] and Wang et al. [**18**] are the earliest papers discussing how to integrate activity scheduling in both domains: communication and sensing. Both of them independently proved the same conclusion: if a convex region is completely covered by a set of nodes, the communication graph consisting of these nodes is connected when $\mathrm{CR} \geq 2SR$.

In other words, under the condition that $\mathrm{CR} \geq 2\mathrm{SR}$, a sensor network only needs to be configured to guarantee coverage in order to satisfy both coverage and connectivity.

1.7.2.3. Sensor network model

Many models for sensor network have their origin in classic areas of theoretical computer science and applied mathematics. Since the topology of a sensor network can be regarded as a graph, the distributed algorithms community uses models from graph theory, representing nodes by vertices and wireless links by edges [**19**].

A sensor network is described by a graph *G=(V,E)*, where *V(vertices)* is the set of sensor nodes, and *E(edges)* where $E \subseteq V^2$ describes the adjacency relation between nodes. That is, for two nodes u, v∈V, (u, v)∈E if v is adjacent to u. In an undirected graph, it holds that if (u, v)∈E, then also (v, u)∈E; that is, edges can be represented by sets{u, v}∈E rather than tuples [**19**].

a)Edges of a graph

Let $V \subseteq R^2$ be a set of nodes in the two dimensional Euclidean plane X. Given a communication range CR, dist(u, v) giving the Euclidean distance between u and v . The Euclidean graph G= (V,E), is called unit disk graph if any two nodes are adjacent if and only if their Euclidean distance is at most CR [**19**]. The model of the unit disk defines the set E of edges:

$$E = \{(u, v) \in V^2 \mid u \neq v \wedge dist\,(u, v) \leq Rc\} \quad (1.1)$$

Suppose that two sensors s_1 and s_2 are located in side X. Two nodes s_1 and s_2 are connected if they can communicate with each other. It means that the edge (s_1, s_2) ∈E if and only if the Euclidean distance between $s_1$and s_2 $(s1\,, s2) \leq CR$. Or if and only if a network path consisting of consecutive edges in E exists between node s1 and s2 [**16**].

Similarly, the network is said to be k-edge connected if there are at least k mutually edge-disjoint paths between every pair of nodes, or equivalently, there is no set of (k − 1) edges whose removal will result in a disconnected or trivial graph. If a network is k-node connected, it is also k-edge connected, but not necessarily vice versa [**20**].

b)Sensor' Neighbors Set

Given a set of nodes distributed in space, we need to specify which nodes can receive a transmission of a node. Throughout, if a node u is within a node v's transmission range, we say that u is adjacent to v, or, equivalently, that u is a neighbor of v. In the absence of interference, this relation is typically symmetric (or undirected); that is, if a node u can hear a node v, also v can hear u [**19**].

The neighbor set of node i is defined as

$$N(u) = \{v \in V \mid v \neq u \wedge (u,v) \in E\} \quad (1.2)$$

The two nodes u and v are nodes neighbors if there is an edge between the two nodes, i.e. the node u and v are in the communication range of each other, and we said they are communication neighbors.

The communication area, of a node u, is modeled by a circle of radius CR. centered at u, inside which the sensor u can transmit and receive messages from its neighbors.

c)Covered and k-Covered Area in sensor network

Given a wireless sensor network consisting of a set of n homogeneous sensors, $= s_1, s_2, \dots, s_n$, in a 2D area X.

Each sensor s_i ,(i=1,…, n), is located at coordinate (x_i ,y_i) inside X and has a sensing range of R_S which is usually called sensing radius .

Any point p in X is said to be covered by s_i if it is within the sensing range of s_i,

$$p \text{ is covered} \Leftrightarrow \exists s_i \in S, dist\,(p, s_i) \leq SR$$

Similarity, the area X is covered, if every point in X is covered by at least one sensor in S, we say that area X is covered by S.

And any point p in X is said to be k-covered if it is within at least k sensors' sensing ranges.

$$p \text{ is } K\_covered \Leftrightarrow \exists \text{ a set } V = \{v1, v2 \dots, vk\} \subset S \;, \forall\, v_i \in V\; dist\,(p, v_i) \leq SR$$

1.7.2.4. Sensing models

WSN nodes generally have widely different theoretical and physical characteristics. Hence, numerous models mentioned in [**16**] of varying complexity can be constructed based on application needs and working environment. Interestingly, most sensing device models share two facets in common:

(1) Sensing ability diminishes as distance increases.

(2) Due to diminishing effects of noise bursts in measurements, sensing ability can improve as the allotted sensing time (exposure) increases.

Assume sensor S_i is deployed at point (x_i,y_i). For any point p at (x,y), we denote the Euclidean distance between S_i and P as dist(S_i,p),i.e. $dist(S_i,p) = \sqrt{(x_i - x)^2 + (y_i - y)^2}$

Define the general sensibility S(S_i,p) of Si at an arbitrary point P as :

$$S(S_i, P) = \frac{\lambda}{[dist(S_i,P]^k} \quad (1.3)$$

Where dist(S_i,p) is the Euclidean distance between the sensor Si and the point P, and positive constants λ and k are sensor technology-dependent parameters. The less dist(Si ,p) is, the stronger sensing ability will be. Obviously, the denominator cannot be zero in (1.3).

Hence, a common denominator can modify for $[dist(S_i\ ,p) + \delta]^k$ so that (1.3) is meaningful, where δ is larger than 0, but infinite close to zero.

a)The binary disc sensing model

The simplest model is the binary disc model, according to which a node is capable of sensing only from points that lie within its sensing range and not from any point beyond it. Thus, in this model the sensing range for each node is confined within a circular disk of radius Rs, and is commonly referred to as the sensing radius [**16**].

$$C_{xy}(S_i) = \begin{cases} 1 & if\ dist(S_i, P) < SR \\ 0 & otherwise \end{cases} \quad (1.4)$$

b)The probabilistic sensing model

The probabilistic sensing model is a more actual perception, which can be taken as an extension of the binary disc sensing model

$$C_{xy}(S_i) = \begin{cases} 0 & if\ r + r_e\ \leq dist(S_i\ , P) \\ e^{-\lambda\alpha^{\beta}} & if\ r - r_e\ \leq dist(S_i\ , P) < r + r_e \\ 1 & if\ r - r_e\ \geq dist(S_i\ , P) \end{cases} \quad (1.5)$$

Where $r_e(r_e < r)$ is a measure of the uncertainty in sensor detection , $\alpha = dist(S_i\ , P) - (r - r_e)$, and λ and β bare parameters that measure detection probability when a target is at distance greater than r_e but within a distance from the sensor. This model reflects the behavior of range sensing devices such as infrared and ultrasound sensors [**16**].

1.7.2.5. Centralized/distributed algorithms

Once sensors are deployed, an algorithm is run to determine whether sufficient coverage exists in the area.

- Traditionally, a centralized algorithm requires each sensor node to forward all its observations to the fusion center, which results in large energy in communication.
- A distributed algorithm, on the other hand, is run on nodes throughout the network, and allows each sensor node to decide its own working mode by the neighbors' information each has gathered.

Compared to centralized algorithms, distributed algorithms reduce communication energy and detection accuracy while increase the processing energy. Overall, distributed algorithms are more suitable for large-scale networks [**16**].

1.7.3. Performance Metric

Performance metrics are used to compare different coverage control algorithms. For example, if two activity scheduling algorithms can achieve the same level of coverage degree and coverage ratio, the one selects a fewer number of nodes is often considered as the better one. Furthermore, with different problem settings, the performance metrics can be different for different coverage problems [**3**].

1.8. Design Issues of Network Coverage

In what follows, we classify the following design issues for coverage problems: namely, coverage type, deployment method, sensor heterogeneity, activity scheduling, coverage degree, coverage ratio, network connectivity, and performance metric.

1.8.1. Coverage Types

The first step in deploying a wireless sensor network is determining what it is exactly that you are attempting to monitor. Typically you would monitor an entire area, watch a set of targets, or look for a breach among a barrier [**4**]. Coverage type refers to the subject to be covered by a sensor network.

Cardei et al. [**10**] argue that, according to the subject to be covered, coverage in sensor networks can be classified into three types as illustrated in Figure 1. 6:

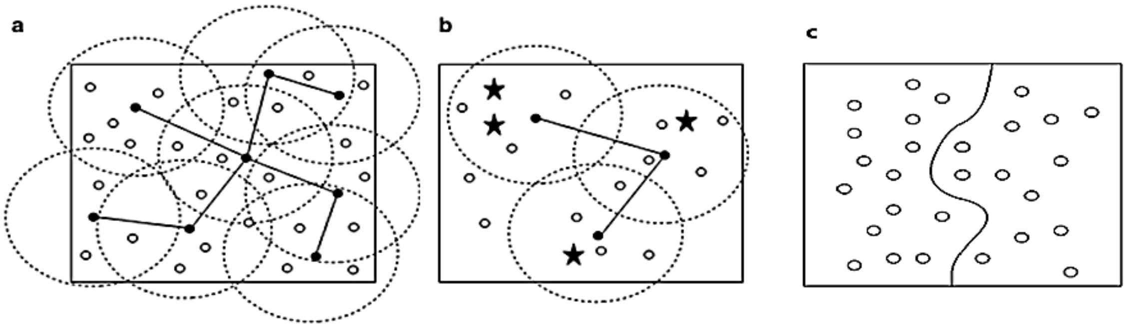

Figure 1. 6: Coverage types. (a) Area coverage. (b) Point coverage. (c)Barrier coverage

a) *Area Coverage (Blanket Coverage):* In the area coverage, the main objective of the sensor network is to cover (monitor) a given area of interest. An area is covered by a WSN if every point in the area is located within communication range of an active sensor which is part of the network. Figure 1. 6. (a) shows an example of a square area which is covered by a WSN [**10**]. In Area coverage problem, on the other hand, equally treats every point in the sensor field and addresses the problem of how to efficiently cover the whole sensor field [**3**].

b) *Point Coverage (Sweep Coverage):* In the point coverage problems, targets are often modeled as a set of discrete points within the sensor field, and it concerns how to cover these targets [**3**]. In the point coverage, the main objective of the sensor network is to cover a set of points. These points may represent targets that have to be monitored continuously by the sensors in the network. Yet in some other scenarios, these points can approximate an area or a region. Figure 1. 6. (b) shows an example of point coverage [**10**].

c) *Barrier Coverage:* Barrier coverage is different from point coverage and area coverage in that the subjects to be covered are not known before node deployment. Instead, it concerns with constructing a barrier for intrusion detection or finding a penetration path across the sensor field with some desired property [**3**]. For example, the maximal breach path problem presented in [**21**] concerns with the barrier coverage, which tries to find a path such that each point on the path has the maximal Euclidean distance to its closest sensor. In barrier coverage, people are interested in each crossing path, which is defined as a path that crosses the complete width of the belt-region-like field [**22**]. Figure 1. 6.(c) shows an example of barrier coverage.

1.8.2. Sensor Deployment Method: Deterministic versus Random

Deployment method concerns with how a sensor network is constructed. In general, a sensor network can be constructed by deterministically placing sensor nodes at desired locations or by randomly scattering sensor nodes into the sensor field.

a)Deterministic Sensor Deployment

In deterministic sensor deployment, the common objective is to place the least number of sensor nodes (the minimum network setup cost) to achieve the application coverage requirement [**3**]. A deterministic sensor placement may be feasible in friendly and accessible environments and when the network size is relatively small [**10**].

b)Random Sensor Deployment

Random sensor distribution is generally considered in remote or inhospitable areas, for military applications, and for applications involving a large population of sensors [**10**]. There are two commonly used random deployment models.

- *Uniform Sensor Deployment*

The random uniform deployment of N sensor nodes such that each node has equal likelihood of falling at any location in the sensor field, in-dependently of the other nodes.

- *Random Sensor Deployment*

In the random deployment, an interesting question is: what is the minimum number of sensor nodes that need to be scattered so that complete coverage can be achieved. This is the critical sensor density problem.

1.8.3. Sensor Type: homogeneous versus heterogeneous

Some applications assume that all sensor nodes in the network have the same characteristics (e.g., sensing range, communication range, types of sensors on the sensor board), while other applications assume that the sensors are heterogeneous [**10**].

Sensor nodes are heterogeneous refers to that sensor nodes have different sensing, processing, or communication capabilities. For example, some sensor nodes are resource rich

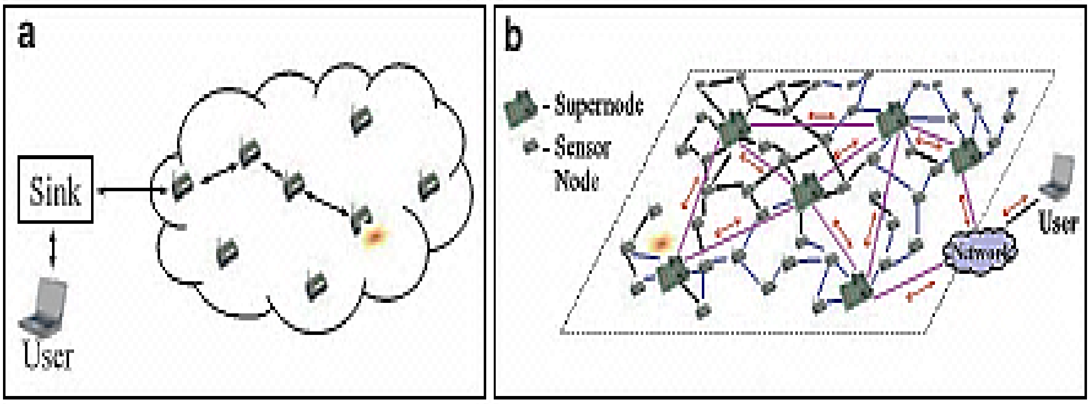

Figure 1. 7: Network types. (a) Simple WSN. (b) Heterogeneous WSN

nodes with more power supply or are equipped with better sensing, processing, and communication units. In the context of sensing disk coverage model, some sensors may cover a disk with larger radius than other sensors'. A heterogeneous sensor network may also consist of both stationary and mobile sensor nodes [**3**].

1.8.4. Network type: simple or heterogeneous.

Some problems consider a simple (or standard) architecture consisting of sensor nodes that report data to a sink, see Figure 1. 7(a). Other works consider a heterogeneous architecture. One architecture which has been recently explored contains two types of wireless devices, as presented in Figure 1. 7 (b). Lower layer is formed by sensor nodes with size and weight restrictions, low cost, limited battery power, short transmission range, low data rate , and low duty cycle. Main tasks performed by a sensor node are sensing, data processing, and data transmission/relaying. Upper layer consists of resource-rich supernodes overlaid over the sensor network, as illustrated in Figure 1. 7 (b). Supernodes can have two radio transceivers, one for communication with sensor nodes and the other for communication with other supernodes. Supernodes have more power reserves and better processing and storage capabilities than sensor nodes. Wireless communication links between supernodes have considerably longer range and higher data rates, allowing supernode network to bridge remote regions of the interest area [**10**].

One or more supernodes are sinks, that is, they relay data between the heterogeneous WSN and users. Supernodes are more expensive, and, therefore, fewer are used than sensor nodes. One of the main tasks performed by a supernode is to transmit/relay data from sensor nodes to/from the sinks [**10**].

1.8.5. Activity Scheduling

Activity scheduling is to schedule the activation and deactivation of nodes' sensor units. In a randomly deployed sensor network, the scattered sensor nodes may be more than the optimum. If the area covered by one sensor can also be covered by other sensors, such a sensor can be as redundant and can be temporarily transited into the energy saving sleep state.

Hence the objective of activity scheduling is to decide which sensors to be in which states and for how long time, so that application coverage requirement can be guaranteed and network lifetime can be prolonged [**3**].

1.8.6. Coverage Degree

Coverage degree describes how a point is covered. For example, in the sensing disk coverage model, coverage degree refers to how many sensors cover a point. A point is called covered if it is within k distinct sensors' coverage disks. Using more than one sensor to cover a point can improve coverage robust-ness. If a point is covered by k sensors, then it can tolerate up to k−1 failed sensors. Coverage degree is considered as one of the application coverage requirements to be observed by coverage control algorithms [**3**].

1.8.7. Coverage Ratio

Coverage ratio measures how much area of a sensor field or how many targets satisfy the application requirement of coverage degree. For example, if eight out of ten targets are covered, then the coverage ratio is 80%. We sometimes use complete coverage to refer to 100% coverage ratio, that is, every point within sensor field (or every target in the target set) achieves the required coverage degree.

Similarly, we use partial coverage to state the situation that not all points in the sensor field (or not all targets in the target set) can be covered with the required coverage degree. Complete area coverage is a very strict requirement, which normally requires a large amount of sensors to be deployed into the sensor field. Instead of asking for 100% coverage ratio, we may allow that some points or targets are not covered and trade-off the coverage ratio with less active sensors. Coverage ratio is often regarded as one of the application coverage requirements to be observed by coverage control algorithms [**3**].

1.8.8. Event type: simple or composite

Some works assume only one sensing measure (e.g., acoustic sensor), while others address composite events. For example, a fire event is better detected if temperature, humidity, and smoke sensor measurements are combined. This will form a composite event. In such a situation, there are coverage requirements for each of the atomic events forming the composite event [**10**].

1.8.9. Sensor mobility: static versus mobile

Majority of studies assume that the sensors are stationary, while some works consider mobile sensors. In mobile sensor networks, sensors can self-propel via springs, wheels, or they can be attached to transporters, such as robots and vehicles [**10**].

A mobile sensor node is equipped with a locomotive unit and can move around after deployment. In general, a mobile node is more expensive than its stationary compeer. As mobile nodes can move to desired locations, it is not unexpected that using mobile nodes can improve sensor network performance. Such performance improvements are often at the cost of expensive mobile nodes and more energy consumed for moving [**3**] .

1.8.10. Sink mobility: static or mobile.

Some works assume that the sinks are stationary, while others assume they are mobile. Recent advances in the field of robotics make it possible to integrate robots as sinks (or gateways) in WSNs.

Adding mobile devices to WSNs infrastructure has attracted increased attention recently. Much of the work has been conducted on data gathering applications, where the mobile sinks move randomly, using predetermined paths, or autonomously. The moving strategies where the sinks take the moving decisions autonomously can better adapt to various network conditions [**10**]. In some cases, a sink can also be a mobile node, and it moves around to collect sensing data [**3**].

1.9. Evaluation metrics of coverage algorithms

How to evaluate the performance of coverage and its algorithm is very important for the network's usability and effective-ness. The main factors are considered as in [**16**] as follows:

1.9.1. QoS of coverage

The Qos of coverage decides the completion of network tasks, reflects the network's sensing ability to the physical world, and is the basis standard of algorithm evaluating.

1.9.2. Number of active nodes

In the case of meeting the coverage requirements, the fewer number of active nodes are, the larger effective coverage area will be. In other words, reducing active nodes do well in the performance of energy consumption.

1.9.3. Associating with the node location or not

Coverage control algorithms associated with a node location depend on external infra-structure (such as GPS) or some position mechanisms, relatively cost high and need to consume large amounts of energy. Meanwhile, there are still some accuracy issues on position. Therefore, coverage control algorithms, not involving position information, have a greater advantage.

1.9.4. Energy efficiency

Coverage control algorithms not only require lowest energy consumption in a single monitoring task, but also maintain energy balance of the network in a series of monitoring tasks. Energy-efficiency mechanisms have also direct impact on WSN lifetime.

1.9.5. Communications overhead

Data transmission is the main source of a sensor node energy consumption. Coverage control algorithms with low cost in the process of communication have a greater advantage.

1.9.6. Network scalability

Coverage control algorithm should be able to adapt to both the scale of different WSNs and the network topology dynamically changed

1.9.7. Network Lifetime

Some other performance metrics include the coverage lifetime which is the maximum time to ensure application coverage, the coverage intensity which is defined as the average time ratio between covered and uncovered period for a point, the movement cost which is used to compare movement strategies, and so on [**3**].

1.9.8. Network Connectivity

Another important requirement is WSN connectivity. Active sensors must form a connected topology to allow sensor data to be relayed to the user.

Sometimes, k-connectivity is enforced, where each sensor node has k distinct paths to a sink. This improves reliability: in case some paths or sensors become unavailable, data can be relayed on alternate paths. Network connectivity, though independently controlled by radio transceiver, is also used as a performance metric for activity scheduling [**10**].

1.10. A Taxonomy for Area Coverage Problems

Bang W. [**3**] present a taxonomy for network coverage problem, by classifying coverage problems into three categories based on the coverage type, namely, point coverage problems, area coverage problems, and barrier coverage problems.

In this dissertation, we focus mainly on blanket coverage which is depicted in Figure 1. 8, where the objective is to deploy nodes in strategic ways, such that an optimal area coverage is achieved according to the needs of the underlying applications. The coverage problem basically requires placing a minimum number of nodes in an environment, such that every point in the sensing field is optimally covered [**20**].

In the area coverage problem, the subject to be covered is the whole sensor field. Complete area coverage requires that every space point within the sensor field should be covered, while partial area coverage allows that some space points need not to be covered.

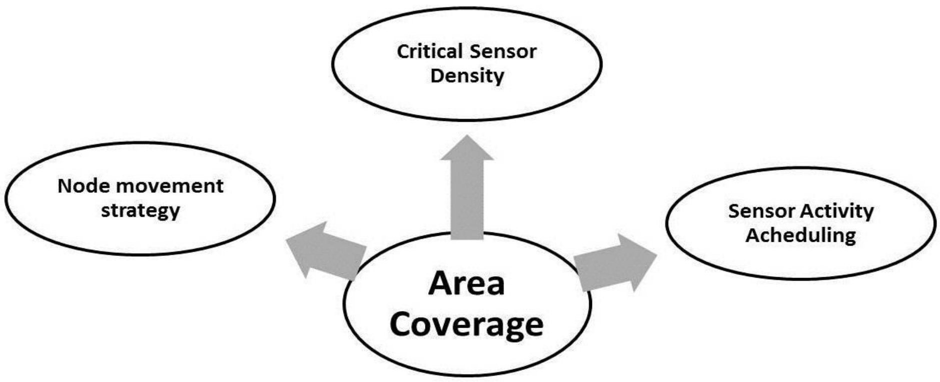

Figure 1. 8:A taxonomy for Area coverage problems in sensor networks [3]

1.10.1. The critical sensor density (CSD) problem

The CSD problem for coverage configuration before network deployment, where the objective is to find the least number of sensor nodes per unit area to provide complete coverage for the whole sensor field. In deterministic node placements, the objectives are to determine not only the least number of nodes to be placed but also their locations. In random node deployments, the objective is to determine the critical sensor density:

Whenever the number of sensor nodes to be scattered is not less than this critical density times the area of the sensor field, all points of the sensor field can be covered almost surely in every random deployment.

1.10.2. The sensor activity scheduling problem

This problem has been intensively studied in the literature. Based on different assumptions and objectives, many problem variants have been formulated and different algorithms have been proposed.

Activity scheduling (see Figure 1. 9) is to schedule the activation and deactivation of nodes' sensor units. In a randomly deployed sensor network, the scattered sensor nodes may be more than the optimum. If the area covered by one sensor can also be covered by other sensors, such a sensor can be as redundant and can be temporarily transited into the energy saving sleep state.

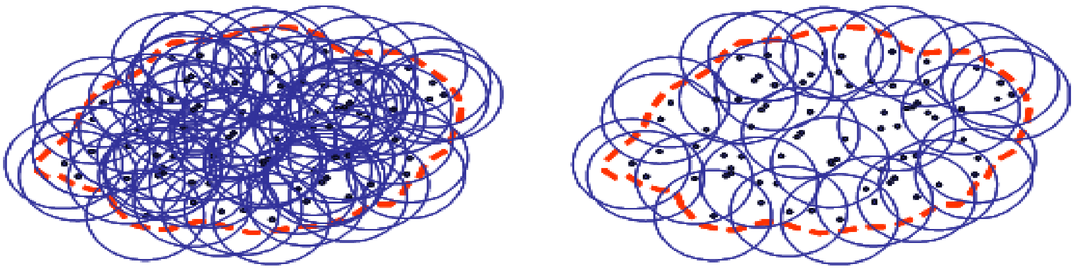

Figure 1. 9: Sensor Activity Scheduling. (a) a deployed sensor network.
(b) Only part of the nodes is necessary for covering the entire area [23]

Hence the objective of activity scheduling is to identify coverage redundant sensors and schedule sensors' activity, to decide which sensors to be in which states and for how long time, so that application coverage requirement can be guaranteed and network lifetime can be prolonged. In the context of coverage problems, network lifetime is often defined as the period from the network setup time to the time that the deployed network cannot provide adequate coverage (e.g., the coverage ratio less than a predefined threshold).

1.10.3. The node movement strategy problem

The node movement strategy for sensor networks containing mobile nodes, where the objective is to leverage mobile nodes to control network coverage. Mobile nodes change network coverage characteristics via moving to the desired locations. Although network coverage can be greatly improved, the product cost and the moving cost of mobile nodes need to be minimized. The design of node movement strategy should balance between network coverage and movement cost [**3**].

This second category consists of coverage schemes that exploit mobility to relocate nodes to optimal locations to maximize coverage. Keeping this in mind, some of the deployment strategies take advantage of mobility to relocate nodes to sparsely covered regions after an initial random deployment to improve coverage [**20**].

1.10.3.1. Functionality and Mobility approaches

In a mobile WSN, there is at least one mobile entity and the remaining sensors are static. According to the design objective, the mobile entities are able to communicate with its neighboring sensors if required. In addition, if there are multiple mobile entities, they are capable of forming a local network like a MANET, which is self-configurable, adaptive to the changing environment, robust, and scalable [**14**].

The mobile entities of interest in WSN can be sinks, relays, or sensors [**8**]. In addition, a mobile entity could be a clusterhead in the network, depending on the deployment strategy, network architecture, and application [**14**].

In the following, we investigate some approaches that improve the network lifetime, coverage, and connectivity by using mobile entities.

a)Mobile Sink Approach

The mobile sink approach [**25**,**26**,**27**] is a common solution exploiting sink nodes' mobility in order to get closer in the vicinity of the reporting sensors. It also achieves higher degree of load balancing among sensor network nodes and can offer extensive improvements of the network lifetime. The mobile sink can follow random, predictable, or controlled mobility. Random movement of the sink node can be equivalent with the scenario where people wearing sinks randomly move and collect information from sensors deployed in a certain area (market for example). Predictable mobility can be the movement of a bus or train, while robots can be enabled to achieve specific tasks with controllable movements [**8**].

The basic role of the BS is to collect the data generated from various sensors. Additionally the mobile BS mounted on the mobile unit can effectively enhance the lifetime by periodically or continuously changing its locations according to a predefined strategy [**14**].

The mobile sink approach has several disadvantages. All nodes must know the position of the sink in order to route the information to it. Also, most scenarios have the sink acting as a gateway to a backbone network and it is difficult to engineer a system whereby a mobile sink is always connected to the backbone network.

Mobile sinks perform advantageous regarding network lifetime improvement compared to other mobile entities. In such a way, moving the sink nodes demonstrates better performances than the mobile relay approach, but relays are beneficial in application scenarios where a mobile sink is not feasible (for example, in hostile terrains) [**8**].

b)Mobile Relay Approach

Another approach to enhance the connectivity and the network lifetime in WSNs is the mobile relay approach as in [**28**,**29**,**30**]. It can lead to elongating lifetime of bottleneck sensor nodes [**8**]..Mobile relaying nodes are special mobile sensors designed for releasing the relaying load of some sensors in the network. The mobile relaying node is able to move in a way to serve as a substitute relaying node for the sensors. For example, they could be Mobile Ubiquitous LAN Extensions (MULEs) [**29**] or Message Ferry [**28**] which are mobile devices specifically designed to roam around to collect data from nearby sensors and deliver them to the BS [**14**].

Mobile relay can follow a random, predictable, or deterministic mobility pattern. Randomly moving Data MULEs are introduced as forwarding agents to help gather the sensing data, thereby saving energy due to single-hop transmissions (i.e., from a sensor to a mule that is passing by) . Controllably moving Message Ferry is a mobility-assisted approach, which utilizes a set of special mobile nodes called message ferries to provide communication service for nodes in the deployment area. These two approaches are introduced for sparse sensor networks, while deterministic mobility is suitable for dense sensor networks where the relay follows a deterministic path for achieving network lifetime improvement.

Mobile relay approach has serious latency drawbacks leading to a decrease in the achievable throughput. Therefore, this kind of solutions is mostly suitable where mobile sinks cannot be implemented and are robust, flexible, and easier to design [**8**].

c)Mobile Sensor Approach

The basic functionality of a sensor is to sense the environment and transmit its data to the BS periodically. On the other hand, the mobile sensor can be used for increasing the efficiency of data relaying and controlling the network topology. Mobile sensors can be distributed in the network domain with minimum human intervention like other stationary sensors. In a SWSN, the coverage and connectivity are fixed once the deployment stage is performed. On the other hand, the mobile sensor could be used to form an ideal topology which could improve the coverage and connectivity, or release the relaying load for some bottleneck nodes [**14**].

The movements of sinks and relays mainly improve network lifetime and provide more efficient energy utilization. The mobile sensors, through network topology adjustment, can improve different network performances (i.e., coverage, sensing etc.).In order to best fulfill its designated surveillance tasks, a sensor network must maximally or fully cover the observed region where the events occur, without internal sensing holes. This task can be accomplished by moving the sensor nodes toward desirable positions and is particularly relevant when an incremental deployment of sensors is not possible. Due to the limited power availability at each sensor, energy consumption will be the primary issue in the design of any protocol for mobile sensors.

Since sensor movements and, to a minor extent, message transfers, are energy-consuming activities, the protocols should minimize movements and message exchanges, while following a satisfactory coverage [**8**].

d)Mobile cluster heads approach

In [**31**] Mobile cluster heads approach is introduced. A mobile clusterhead can be one of the mobile sensors through an election process or a special node placed manually. They form clusters in the network and forward the information collected within their own cluster to the BS.

Unlike stationary clusterheads, mobile clusterheads can increase the energy efficiency and intelligently form the cluster topology adaptively according to the environment or changes in the network mission.

1.11. Conclusion

Recent technological advances have led to the emergence of distributed wireless sensor and actor networks (WSANs) which are capable of observing the physical world, processing the data, making decisions.

In this chapter we provide some backgrounds and introduction about sensors, sensor nodes, and sensor networks. Then we present coverage problem formulation, their objectives, coverage taxonomy, the main types of coverage and their design choices.

Coverage is one of the most fundamental issues in WSNs, which has a great impact on QoS of WSNs. Many algorithms, strategies and mechanisms have been proposed by researchers around the world to solve this problem. We a brief introduction to the basic knowledge of coverage concepts is given in this chapter. Then, we describe the coverage issues from three aspects: The critical sensor density (CSD), sleep scheduling mechanism and coverage schemes that exploit mobility.

The next chapter describes the main and approaches solutions proposed in the litterateurs to the area coverage problem in static and mobile sensor networks.

Chapter 2

Related Work

2.1. Introduction

Many researchers have realized that the network performance of a WSN is restricted by the nature of stationary and are envisioning the design of Mobile WSN to improve the WSN performance by using mobile entities. The Mobile WSN technology can be practical and efficient due to recent advance in robotic technology or realizing the potential usage of naturally moving objects such as vehicles, animals, and even human. Network coverage, on the other hand, can be regarded as a collective measure of the quality of service provided by a network of sensor nodes at different geographical locations. The coverage problem basically requires placing a minimum number of nodes in an environment, such that every point in the sensing field is optimally covered [**14**].

In the area coverage problem, the subject to be covered is the whole sensor field, in which every single point of the network should be within the sensing range of at least one sensor node. In order to reduce data volume and prolong network lifetime, it is much necessary to control which sensors to sense and for how longs. To achieve such a goal, Area coverage has been studied in static, mobile, and hybrid networks [**32**].

Several strategies that are similar in spirit in terms of ensuring coverage of a sensing field, while reducing redundancy and increasing overall network life time.

In this chapter we are interested in particular in area coverage algorithms. We present different area coverage solutions based on node activity scheduling strategy and node movement strategy. We will present in more or less detailed some of these existing solutions in different network type.

2.2. Area coverage solutions Classification

The coverage of an entire field refers to area coverage, in which every single point of the network should be within the sensing range of at least one sensor node. Area coverage has been studied in static, mobile, and hybrid networks. Different types of networks are named according to the motion ability of involved sensor nodes.

Recent research issues about static coverage and dynamic coverage are described here as well as the solution to optimize coverage area of interest [**32**].

Several centralized and distributed approaches have already been proposed in literature. In centralized solutions, the information about topological changes in dynamic networks must be propagated throughout the network, to maintain the information needed for each node to make decision [**15**].

A number of distributed and localized protocols relax this full information propagation but use instead a wave type of computation and communication (with neighbors), memorization at nodes, unbounded delays or have other problems. Localized solutions have significantly lower communication overhead since no global view of the network is required. Localized protocols are needed for dynamic networks, whose topology changes due to mobility, changes in activity status, or changes due to failures or adding more nodes. In localized solutions, topological changes simply imply some modifications in the neighbor-hood of a node [**33**].

In this chapter we introduce existing centralized and distributed approaches of area coverage problem have been studied in literature. Therefore, in what follows, we will a more or less detailed presentation of some of these approaches based on node activity scheduling strategy and node movement strategy

2.3. Solutions based on node activity scheduling strategy

In wireless sensor networks, all nodes share common sensing tasks, which imply that not all sensors are required to perform the sensing tasks during the whole system lifetime. Making some nodes sleep does not affect the overall system function as long as there are enough working nodes to assure it. Therefore, if we initially deploy a large number of sensors and schedule them to work alternatively [**15**], A significant amount of energy savings can be achieved and system lifetime can be prolonged correspondingly, i.e., redundancy is exploited to increase system lifetime. Such techniques; known as node activity scheduling schemes.

2.3.1. CDSC (Centralized Dominating Set for Coverage)

Pazand et al. [**34**] have presented a graph theoretical approach based on minimum dominating set without using location information. This algorithm denoted CDSC (Centralized Dominating Set for Coverage) extracts a collection of dominating sets, after a neighbor's discovery procedure. After that, the base station designates nodes having high degree to be active. CDSC guarantees a high level of field coverage, with the aim to extend the overall network lifetime.

1) Protocol description

A centralized node scheduling scheme have been proposed, to solve the coverage problem in WSNs, based on minimum dominating set (MDS). This scheme has three phases. The first phase involves constructing the graph corresponding to the network. The next step is dedicated to the selection of different sets of nodes dominant sensors. The last phase exploits the dominant sets previously determined to schedule the activity of sensors.

a) Phase 1: construct the network graph

Initially, each node sends a Hello message and built its neighbor table. These tables are sent to the base station in order to construct the graph corresponding to the network and to represent it as adjacency matrix, to initiate the next phase.

a) Phase 2: Determine the dominant sets of minimum cardinality

As the base station chooses the node p having the highest number of neighbour's. This node p becomes the first member of the first MDS (M1). Then, all neighbours of pare excluded from the next selection. The other members of M1 are determined by the same process which continues until processing all nodes.

Then, two different algorithms have been proposed based on the relationship between communication rays (CR) and surveillance (SR) to determine the dominant sets of minimum cardinality.

a) Phase 3: schedule the dominant sets of maximum cardinality

The third phase is the simplest part of the protocol. Once the MDSs calculated in the previous phase, they should be sequenced to ensure network coverage. At each period, After determining M1, all of its members are omitted from the calculation of the next sets. Subsequently, the above process is repeated to obtain members of others MDSs; {M_2, M_3,.., M_k}. Finally, the computed MDSs are scheduled to ensure coverage.

2) Discussion and critics

- ☑ The protocol uses a base station that may be overridden in the case of networks with high density because of the problem of scalability.
- ☑ Any problems found in a centralized system posed by this algorithm.
- ☑ The protocol use a neighborhood discovery phase generally has a high communication cost of increasing the probability of message loss (because of collisions for example). Therefore, the protocol performance degrades dramatically

2.3.2. Distributed Coverage Preserving based on Dominating Set (DCovPDS)

In [**35**], a distributed algorithm, denoted DCovPDS (Distributed Coverage preserving based on Dominating Set), was presented. This algorithm based on the use of the concept of minimum connected dominating set (MCDS) from graph theory. The objectives are to ensure a balance in energy consumption among sensors (network longevity) and to have a low cost inherent in communications.

1) Protocol description

DCovPDS is a decentralized coverage preserving protocol that uses a concept from graph theory; minimum dominating sets, in order to construct cover sets. It divides the network lifetime in successive activity rounds of equal length, each consisting of a decision phase and a sensing phase. In each round, a small number of active nodes are selected to provide coverage, which reduces energy consumption and therefore extends the network lifetime. *DCovPDS* is operated by two phases;

a) Decision phase: All sensors are initially in listening state during a timeout. When the timeout expires, and if the node does not receive any activity message, it will decide to be *active*. Otherwise, it will switch to *passive* state.

b) Sensing phase: active nodes form a Minimum Dominating Set (MDS) of the network, providing coverage of the area of interest.

This protocol is based on a timeout scheme. When a round starts, every node i selects a timeout and evaluates its coverage once its timeout expires. The timeout of a node i, in a given activity round P, is calculated according to its residual energy percentage REP_i and to its state in precedent round (P-1). A timeout inversely proportional to the remaining energy level would allow the weaker nodes to decide later, thus maximizing their chances to preserve their energy by becoming passive.

$$Wtime_i = \frac{A_i}{REP_i} + N_i \qquad (2.1)$$

Where;

REP_i : Residual energy percentage

N_i : Random number

$Wtime_i$: Waiting time or timeout

A_i : Number of activity runds where the sensor was active

The author assume, that any two neighboring nodes would select different random numbers, so that two nodes never attempt to simultaneously decide, which leads to non-covered portions of the interest area.

The determination of a minimum dominating set is a NP-hard problem [19]. Therefore, heuristic solutions have been proposed to solve it. *DCovPDS* suggest two different heuristics (using the same principle) according to the relationship between communication range CR and sensing rang SR.

- ***Case SR=CR:*** If a node i receives at least one message of domination by other nodes before the expiry waiting time $Wtime_i$, so we put the node i in passive state (dominated), otherwise we put the node i active (dominant) and it sends a domination message.

- ***Case SR<CR***: in order to avoid sensing holes, a passive node must have at least m dominating neighbors (m active nodes) and not only one. If a node i receives m (given integer depending on the network density) domination messages by other nodes before the expiry waiting time Tattentei, so we put the node i in passive state (dominated), otherwise we put the node i active (dominant) and it sends a message of domination.

2) Discussion and critics

- ☑ In the end, active nodes perform surveillance (coverage) of the area of interest during the capture phase. DCovPDS treats the coverage problem in WSN effectively with fewer messages (to more than 2 posts / node), and without neighbor discovery phase.
- ☑ However, at the beginning of each period, DCovPDS needs a synchronization process for the nodes start at the same time in the decision phase.

2.3.3. Connected Cover Set based on IDentity of node (CCSID)

The authors of [**36**] presented an centralized algorithm based on the use of the concept of minimum connected dominating set (MCDS) from graph theory. This algorithm will be denoted CCSID (Connected Cover Set based on IDentity of nodes). CCSID suggest determining cover sets in WSN. More precisely, a connected cover set based on identity of node (CCSID) that uses a graph theory concept (MCDS), in order to build coverage sets. The main contributions of this approach are: (1) high coverage ratio, (2) small number of active nodes, (3) connectivity guaranteed.

1) Protocol description

CCSID is proposed as a basis for a solution to the problem of cover set construction. It is based on analytical approach and uses the concept of minimum connected dominating set from graph theory. The WSN is modeled by a graph G (S,E) . CCSID is operated by two phases; The duration of the first phase determines a minimum dominating set MDS as a cover set for assuring the coverage in WSN. And in the second phase, it makes this MDS connected for obtaining a MCDS, for assuring more area coverage and the connectivity in WSN.

a) Phase 1: Construction of MDS.

The first phase determines a minimum dominating set (MDS) as a cover set that ensures the coverage in the WSN. The determination of a minimum dominating set is an NP hard problem. Therefore, heuristic solutions have been proposed to solve it. CCSID obtain the MDS by determining the Maximal Independent Set (MIS) having the minimum cardinality.

b) Phase 2: Construction of MCDS.

In this phase, CCSID assures the connectivity of MDS, it makes disconnected elements of MDS connected by adding the minimum number of node, for obtaining a MCDS, guaranteeing both coverage and connectivity.

2) Discussion and critics

- ☑ CCSID is a centralized solution to maintain coverage and to assure the connectivity in Wireless Sensor Networks based on MCDS.
- ☑ The load on the sensors near the base station is larger than the other (no balancing in the network).
- ☑ Each MCDS provides network connectivity and does not take measures the overall coverage of the monitored zone.
- ☑ The more we approach the base station, the rate of coverage increases. However, the rate decreases as you move away from it.

2.3.4. A Border Coverage Protocol for Wireless Sensor Networks (BCP)

In [**37**], a distributed algorithm, denoted BCP (Border Coverage Protocol) to maintain border coverage in WSNs. was presented. The BCP uses a concept from graph theory: minimum dominating sets and the concept of Voronoi Polygons, in order to cover both the border and the area of interest. BCP performances, in terms of coverage of area of interest and border, are preserved in the case of MAC and physical layer impacts.

1) Protocol description

Border Coverage Protocol is mainly based on a simple method to detect the boundaries nodes by using the principle of Voronoi polygon while running an improved version of DCovPDS protocol. The BCP uses a concept from graph theory: minimum dominating sets and the concept of Voronoi Polygons, in order to cover both the border and the area of interest. This protocol runs in three phases. In the first phase, it detects boundary nodes. Transfer node and internal nodes are detected in the second phase. The last phase is for maintenance operation, *in order to* guarantee much border coverage in WSN.

a)Phase 1: Detection of boundary nodes

Initially, all nodes are considered as boundary nodes, so the Voronoi polygons associated with each node are endless. First, the protocol *DCovPDS* is running and each active node (dominant) sends an activity message (*Msg_Activity, D*).This message is received by one-hop neighboring nodes and the receiver node detects the angle of transmission among one of its four sides, then, the side will be considered as finite. If Voronoi polygon is finite, then, the node is not a boundary node (internal).Else, the Voronoi polygon is infinite and the node is a boundary node. This phase stops the execution when the number of period BCP protocol will run.

b) Phase 2: Detection of transfer and internal nodes

At the beginning of this phase, all the boundary nodes were detected in the previous phase. Each boundary node will send domination and transfer message (Msg_Activity, T) to all its one-hop neighbors. All nodes receiving this message switch to the passive state and are considered as transfer nodes. All transfer nodes of AI are detected at the end of this phase. So, there are two types of nodes here (transfer and internal). In addition, nodes receiving an Msg_Activity message from active boundary nodes are considered passive and internal nodes receiving an Msg_Activity message from active transfer nodes are considered passive as well. This is done by choosing the waiting time of boundary nodes smaller than the one of transfer nodes, and the waiting time of transfer nodes smaller than the one of internal nodes. This process will continue running until all nodes in the boundary area will be out of service.

c) Phase 3: Maintenance

As time goes on, some sensor nodes along the boundaries may exhaust their energy and may die. In this case, each boundary node sends a domination and removal message (Msg_Activity, F) to all its one-hop neighbors. Each receiver of this message, checks if there is a possible node that will replace the dying one according to its table of transmitter nodes. If it is the case, then it will remove the dying node from its table, and will check again if the selected node is a border node or not, to continue monitoring the node area. One continues running the last two phases until all sensors are out of service.

- **Timeout computation**

In the first phase, the nodes use this function for calculating waiting time.

$$Wtime_i = \frac{A_i}{REP_i} + N_i \qquad (2.2)$$

A_i: is the number of periods of activity where the node *i* was active. N_i: a random number of node *i* is calculated as following

$$N_i = \begin{cases} \left]0\ ,\ \frac{RPE_i}{2}\right] \textit{If the node i is passive} \\ \left]\frac{RPE_i}{2},\ 1\right] \textit{If the node i is active} \end{cases} \qquad (2.3)$$

In the second phase the calculation function of the waiting time takes into account the difference between the internal, transfer and boundary nodes. Indeed, the waiting time of the boundary nodes is smaller than those of transfer and internal nodes. So, there are three functions, one for each kind of node. The boundary nodes use formula (4) to compute their waiting time. The second function (6) is the same that is used by the BCP protocol for boundary nodes and we add the value $(E_{initial}/E_{threshold})^2$.

$$Wtime_i = \left(\frac{E_{initial}}{E_{threshold}}\right)^2 + \frac{A_i}{REP_i} + N_i \qquad (2.4)$$

Expression (7) adds a multiplicative factor to the first term of expression (6):

$$Wtime_i = 2 * \left(\frac{E_{initial}}{E_{threshold}}\right)^2 + \frac{A_i}{REP_i} + N_i \qquad (2.5)$$

The waiting time of a passive node is shorter compared to active node, because it adds a low value (A_i=0), so it is more likely for passive nodes to be active.

- ***Case SR=CR:*** The proposed protocol determines dominating sets of boundary, transfer and internal nodes, in a distributed manner.
- ***Case SR>CR:*** In this case, applying the same principle as the case (CR = SR), a great number of portions of coverage holes in the area of interest, which significantly

degrades the coverage. To overcome this problem, a generalization of solution is used based on the concept of m-dominating sets.

2) Discussion and critics

- ☑ A distributed algorithm based on MDS, it is an improved version of DCovPDS protocol, it preserves both border and area coverage USING the concept of Voronoi Polygon.
- ☑ MDSs constructed provide the coverage of area, preserve energy under an arbitrary ratio of sensing range and transmission radii
- ☑ BCP treats the coverage problem in WSN effectively with low communication overhead, without neighbor discovery phase, and without location information.
- ☑ BCP Need more active nodes to overcome the holes appearing in the AI. And covering sensors may not be connected.

2.3.5. Optimal Geographical Density Control (OGDC)

OGDC is One protocol that attempts to combine coverage and connectivity is the Zhang and Hou [**38**] have proposed the optimal geographical density control (OGDC) protocol. This protocol is decentralized and localized, it tries to minimize the overlap of sensing areas of all sensor nodes for the case when Rc >= 2Rs, where Rc is the sensor node communication range and Rs is its sensing range.

1) Protocol description

An important, but intuitive result was proved by Zhang and Hou [**38**], which states that if the communication range Rc is at least twice the sensing range Rs, a complete coverage of a convex area implies connectivity of the working nodes. If the communication range set up is too large, radio communication may be subject to excessive interference. Therefore, if the communication range can be adjusted, a good approach to assure connectivity is to set transmission range as twice the sensing range. Two nodes are neighbors if they have overlapping sensing areas. By intuition as shown in Figure 2. 1:Two sensors, u and v, with overlapping sensing areas can directly communicate with each other if Rc>=2Rs, when Rc >= 2Rs , two neighbors are within their communication ranges.

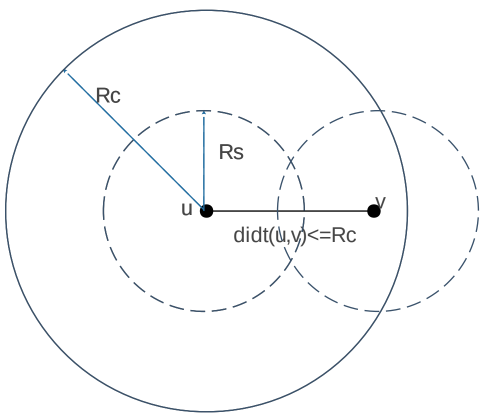

Figure 2. 1:Two sensors, u and v, with overlapping sensing areas can directly communicate with each other if Rc>=2Rs

The algorithm runs in rounds, and at the beginning of each round, a sensor with sufficient power will be randomly chosen to start the process of node selection a set of one or more starting nodes are selected as working nodes. After a backoff time, a starting node broadcasts a power-on message and changes its state to ON. The power-on message contains (1) the position of the sender and (2) the direction along which a working node should be located. The direction indicated by the power-on message of a starting node is randomly distributed. Having starting nodes randomly selected at the beginning of each round ensures uniform power consumption across the network. Also, the back-off mechanism avoids packet collisions. The nodes in OGDC can be in any of three states: ON, OFF, or UNDECIDED. They quantify time into rounds which are comprised of a node selection phase and a steady state phase.

a) Node selection phase

At the beginning of each round, all nodes are UNDECIDED and carry out the operation of selecting working nodes by changing either to ON or OFF state until the beginning of the next round. This decision is based on the power-on messages received. Every node keeps a list with neighbor information. When a node receives a power-on message, it checks whether its neighbors cover its sensing area, and if so, it will change to OFF state. A node decides to change into the ON state if it is the closest node to the optimal location of an ideal working node selected to cover the crossing points of the coverage areas of two working neighbors. By the end of this phase, all the nodes change their states to either “ON” or “OFF” OFF for the steady state phase.

b) Steady state phase

In the steady state phase, all nodes keep their states fixed until the beginning of the next round. The length of each round is so chosen that it is much larger than that of the node selection phase but much smaller than the average sensor lifetime.

2) Discussion and critics

- ☑ Zhang and Hou [**38**] is one of the earliest papers discussing how to integrate activity scheduling in both domains: communication and sensing. He has proved that, under the condition that Rc >= 2Rs, a sensor network only needs to be configured to guarantee coverage in order to satisfy both coverage and connectivity.
- ☑ OGDC is a fully distributed algorithm, while the location information of each sensor node needs to be known in advance.
- ☑ Simulation results show that OGDC does well in reducing number of working nodes. OGDC shows good results in terms of percentage of coverage, number of working nodes, and system lifetime. But it does not take into consideration the edge effect of the sensing field, OGDC cannot preserve the original sensing coverage completely

☑ OGDC protocol is fully localized so it should scale well. However is useful primarily for dense sensor networks. If there is not a high degree of redundancy then the overhead of implementing would not be worth the benefits.

2.3.6. Coverage through Hilbert Mobile Beacon

The choice of Hilbert trajectory is motivated by the fact that is sufficient to locate all nodes in the network and to perform coverage while ensuring network connectivity at the same time. J.M. Bahi et al. in [**5**] tackle the problem of localization and coverage in randomly deployed high density wireless sensor networks using of space-filling curves by organizing the sensor nodes into disjoint sets (rounds) which are activated successively. The main idea is to use a single mobile beacon node to help static sensor nodes to locate themselves and to find a coverage scheduling between them.

1) Protocol description

This approach defines a Hilbert space filling curve as a trajectory for the mobile beacon. The use of a mobile beacon is suggested to divide the region of interest into unit squares following the same method used in the Hilbert space filling curve. A proper choice of the order of the Hilbert curve (i.e. the region subdivisions) is studied to guarantee the localization of all nodes as well as the total area of coverage.

➢ **Hilbert Trajectory Order**

In order to localize nodes in the sensor network. A theoretical study, that allows all nodes to be localized, led to the following condition on the Hilbert curve order *m*:

$$m \geq \left\lceil log_2\left(\frac{\max(Height, Width)}{\sqrt{\frac{2}{5}}R_c}\right)\right\rceil \qquad (2.6)$$

Where Rc stands for the communication range of the mobile beacon

On other hand, as mentioned in Zhang and Hou [**38**], if the communication range "CR " of nodes is at least twice the sensing range "SR", a complete coverage of a convex area implies connectivity of the working nodes.

For this coverage purpose, if every node in a US is covering the entire US, only one node per US is required to be active during a round to ensure a whole coverage of the initial covered area, where at the same time connectivity is maintained.

More formally, a node is covering a US iff the order of Hilbert trajectory as follows:

$$\mathrm{m} \geq \left\lceil \log_2\left(\frac{2\sqrt{2}\max(\mathrm{Height,Width})}{\mathrm{R_s}}\right)\right\rceil \qquad (2.7)$$

Equations (2.6) and (2.7) define two conditions on the order of Hilbert trajectory in order to be able to locate nodes and perform coverage respectively. Note that the communication range is usually much longer than the sensing range. If only condition (2.7) is considered, the both localization and coverage are performed at the same time.

- **Mobile Hilbert beacon Communication**

When the mobile beacon traverses the region of interest, it sends packets containing the h-keys. Instead of sending two coordinates X and Y it sends only one, which saves energy on the mobile beacon. In this approach, in order to perform localisation, the mobile beacon sends packets at every h-keys position. By doing so, the correspondence between the Hilbert coordinates and the grid coordinates is maintained. The localisation algorithm as well as the coverage algorithm needs this information to work.

- **Increment coverage**

After finding the Hilbert trajectory order, the mobile beacon traverses the monitored area, US after US progressively, and sends localization packets to nodes. Once nodes, within a US, have localized themselves, they send their positions to the mobile node which performs local grouping into disjoint sets. It sends then back the activation schedule to nodes, and keeps their position in a local table for future use[1]. For this reason, it is called ***incremental coverage***. Generally, all sensor nodes which are part of the active set are in the active state, whereas all other nodes are into a low energy, sleep state. These sets can be activated in turn by rounds. A round is the life-time of an activated set.

Therefore, power aware strategies should be designed to prolong network lifetime. In this work three different scheduling methods are presented according to the choice of an active node per US:

- *Random scheduling*, the first is to-tally random and it is based on activating a node per US. It is the simplest approach of incremental coverage
- *Global scheduling*, consists on finding the best schedule based on connectivity and max coverage constraints, using branch && bound strategy. In this technique, the objective function is to find an optimal solution which guarantees the highest number of rounds while ensuring the fullest possible coverage. The main metric is considered to maximizing the number of rounds.
- *Max distance scheduling,* an efficient heuristic method based on *max-distance* are proposed, evaluated and Compared to others scheduling methods, which done best results. This heuristic reduces the search space by considering a less complicated metric. It is based on the maximum distance between two sets of coplanar points for Round construction. This is based on the definition of the surface of intersection between two discs, which is based on the distance between their centers. We know that when the distance between the two centers increases the area of overlap decreases. Obviously, increasing the distance between two nodes, we decrease the overlap surface and on the other hand we increase the area covered by these nodes.
- *Improved random scheduling,* It is an improved version of the random approach. The main difference is that this new version includes knowledge about previous schedules. The choice of an active node for a US remains random.

2) Discussion and critics

- ☑ This approach defines a Hilbert space filling curve as a trajectory for the mobile beacon. The choice of Hilbert trajectory is motivated by the fact that is sufficient to locate all nodes in the network and to perform coverage while ensuring network connectivity at the same time.
- ☑ Use mobile beacon for both localization and coverage. Proposed solution organizing the deployed nodes into a maximum number of disjoint active sets to reduce energy consumption.
- ☑ a significant reduction of the consumed energy. Also, finding a scheduling (coverage) between nodes will dramatically simplify and reduce the initialization phase of the network, i.e. neighboring discovery and routing initialization.
- ☑ The random scheduling is totally random and it is based on activating a node per US, this simple scheduling with low complexity (O(1)) gave better results. However, it does not select the optimal schedule neither it takes advantage from previous schedules in previous visited USs.
- ☑ The global scheduling approach selects the best schedule, its complexity (O(n!)) remains a serious drawback to be considered in our mobile beacon
- ☑ An efficient heuristic based on max-distance is proposed. It reduces dramatically the search space, while considering a less complicated metric , the latter derives the best schedule, with regards to the max distance metric. This leads to a low complexity of O(n log(n)).
- ☑ The improved version of the random approach, it takes advantage from previous schedules in previous visited USs, and try to select the optimal or near optimal schedule with acceptable complexity.
- ☑ The results clearly highlighted the effectiveness of schedules derived with max-distance approach. Their quality is better, in particular in high density networks.

2.4. Solutions based on nodes movement strategy

The mobile entities of interest in WSN can be sinks, relays, or sensors. In our work we interest about sensor mobility in mobile WSN and their impact on coverage and network topology. The design of node movement strategy should balance between network coverage and movement cost. It is often assumed that given enough remaining energy, a mobile node can move to a desired location without any limitation in the movement distance. In some cases, however, mobile nodes can only move within a limited distance [**3**].

The node movement strategy problem for sensor networks containing mobile nodes, where the objective is to leverage mobile nodes to control network coverage. Mobile nodes change network coverage characteristics via moving to the desired locations. Although network coverage can be greatly improved, the product cost and the moving cost of mobile nodes need to be minimized. Some other objectives include optimizing movement cost and reducing message overhead. The existing movement schemes grouped into three categories, namely, coverage pattern based movement, virtual force based movement, and grid quorum based movement [**3**].

In the group of virtual force-based node movement strategies [**39**,**40**,**41**,**42**,**43**], mobile nodes are likened as electromagnetic particles. Similar to the electrostatic force between two electromagnetic particles in the physical world, two mobile nodes as two particles expel each other if their distance is too close or attract each other if their distance is far apart.

2.4.1. Virtual Force Algorithm (VFA)

A similar idea with a different approach based on cluster-based distributed sensor networks is presented in [**43**]. A virtual Force Algorithm (**VFA**) is proposed as a sensor deployment strategy to enhance the coverage after an initial random placement of sensors, where the sensor locations are determined based on a virtual force assumed to exist among sensors and between the sensors and obstacles in the field. For a given number of sensors, the VFA algorithm attempts to maximize the sensor field coverage.

1) Protocol Description

The cluster head is responsible for executing the VFA algorithm and managing the one-time movement of sensors to the desired locations. The VFA algorithm is designed to be executed on the cluster head, which is expected to have more computational capabilities than sensor nodes. The cluster head uses the VFA algorithm to find appropriate sensor node locations based on the coverage requirements. The new locations are then sent to the sensor nodes, which perform a one-time movement to the designated positions. No movements are performed during the execution of the VFA algorithm.

a) The VFA Algorithm

A judicious combination of attractive and repulsive forces is used to determine the new sensor locations that improve the coverage. Once the effective sensor positions are identified, a one-time movement with energy consideration incorporated is carried out, that is, the sensors are redeployed, to these positions.

Each sensor locally calculates the sum of the vectors (representing relative positions) of neighboring nodes acting it and moves according to the movement vector thus computed.

In the so-called virtual force based mobile sensor deployment algorithm (VFA), there is a powerful cluster head (CH) which executes the algorithm. The CH communicates with all the sensors, collects sensor position information, and calculates forces and desired position for each sensor.

In VFA, the distance between two adjacent nodes, when all nodes are evenly distributed, is defined as a *threshold* to distinguish attractive or repulsive force between the nodes.

The total force on a node will be the sum of all the forces given by other sensors together with obstacles and preferential coverage in the area (Figure 2. 2, Equation 2.12). In terms of forces, each sensor Si node will be exerted by three types of forces:

- Repulsive (Negative) forces due to the obstacles (Frep),
- Attractive (positive) forces from the areas with pure coverage (Fattr),
- And forces (either attractive or repulsive) from the other sensor nodes Sj depending on their distance and orientation (Fsensor: is expressed by polar coordinate notation (r, θ), where r is the magnitude, and θ is the orientation angle.)

Therefore, the total force F can be expressed as:

$$\mathbf{F_i} = \mathbf{F_{attr_i}} + \mathbf{F_{rep_i}} + \sum_{\mathbf{j=1,j\neq i}} \left(\mathbf{F_{sensor_{ij}}}\right) \qquad (2.12)$$

$$\vec{F}_{ij} = \begin{cases} \left(w_A\left(d_{ij} - d_{th}\right), \theta_{ij}\right), & if\, d_{ij} > d_{th} \\ 0, & if\ d_{ij} = d_{th} \\ \left(w_R\left(\frac{1}{d_{ij}}\right), \theta_{ij} + \pi\right), & otherwise \end{cases} \qquad (2.13)$$

where w_A/w_R: measure attractive/repulsive force ,

And d_{th} : a threshold distance, note that d_{th} is a predetermined parameter that is supplied by the user, who can choose an appropriate value of d_{th} to achieve a desired coverage level over the sensor field.

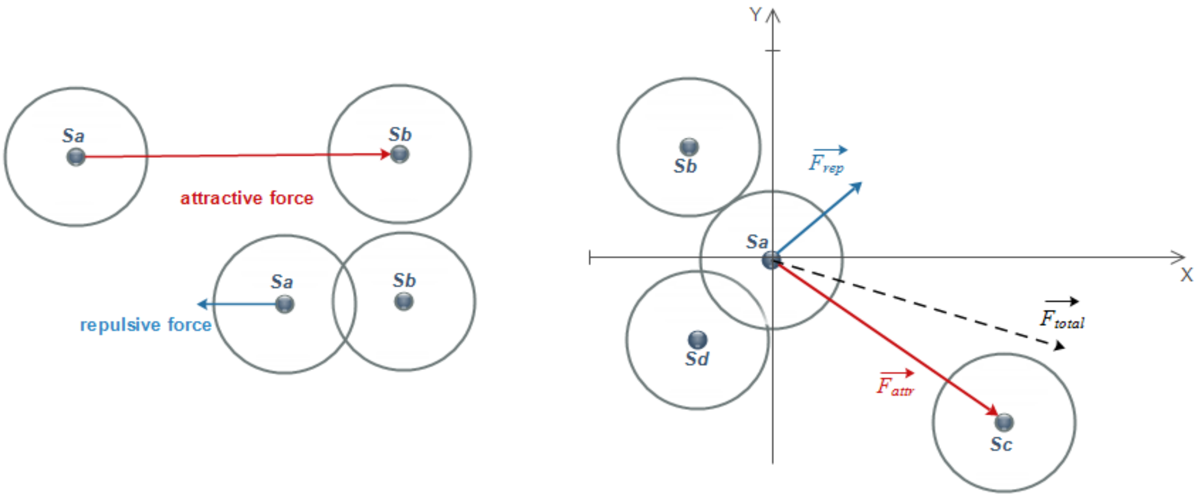

Figure 2. 2:Attractive and repulsive virtual forces

b) Target Localization algorithm

When a sensor detects a target, it sends an event notification to the cluster head. In order to conserve power and bandwidth, the message from the sensor to the cluster head is kept very small; in fact, the presence or absence of a target can be encoded in just one bit.

Detailed information such as detection strength level, imagery, and time series data are stored in the local memory and provided to the cluster head upon subsequent queries. Based on the information received from the sensors within the cluster, the cluster head executes a probabilistic localization algorithm to determine candidate target locations, and it then queries the sensor(s) in the vicinity of the target.

2) Discussion and critics

- ☑ Simulation results shown that the VFA algorithm improves the sensor field coverage considerably compared to random sensor placement. The proposed energy-conserving target localization method also show that considerable energy is saved in localizing a target.
- ☑ The efficiency of the VFA algorithm depends on the values of the force parameters w_A and w_R. The algorithm converged more rapidly for our case studies if $w_R \gg w_A$.
- ☑ The sensor placement strategy is centralized at the cluster level since every cluster head makes redeployment decisions for the nodes in its cluster. Nevertheless, the clusters make deployment decisions independently; hence there is a considerable degree of decentralization in the overall sensor deployment for the network.
- ☑ The VFA algorithm however is also applicable for alternative location indicators, distance measures, and models of preferential areas and obstacles. Hence, the VFA algorithm can be easily extended to heterogeneous sensors.
- ☑ The proposed target localization algorithm can also be used for a deterministic sensor placement based on the pre-computation of sensor locations.

In the group of grid based node movement strategies [**44**,**45**,**46**,**47**,**48**] . The mobility-assisted network redeployment problem is viewed as a load balancing problem in traditional parallel processing systems. The sensor field is partitioned/ into many small grid cells, and the number of nodes in each cell is considered as the load of the cell. Unlike the coverage pattern-based and the virtual force-based movement schemes, the grid based movement schemes need not to specify the exact target location for a moving node. Instead, a target cell is specified for a moving node, and this node can move to any location within the target cell.

The existing literature offers approaches where the sensor area is divided into two dimensional evenly partitioned grid structures [**44**,**45**,**46**,**47**], or in hexagonal grids [**48**,**49**]. In this section we interested in approaches in 2D grids.

2.4.2. Scan-based Movement-Assisted sensoR deploymenT method (SMART)

In [**47**] Yang et al. present an algorithm called *SMART* (Scan-based Movement-Assisted sensoR deploymenT method) is introduced, which partitions the region of interest in a 2D mesh through clustering. The algorithm is distributed and scan based, nodes are treated as a load and the objective is to achieve a load balanced state in each cell.

1) Protocol description

Yang et al. [**47**], apply a scan technique called SMART to identify overloaded and underloaded cells and to direct the movement of those moving nodes from overloaded cells. The scan process will be executed twice–once for all rows and once for all columns to achieve a balanced state. Each cluster corresponds to a square region and has a CH that is in charge and which communicates with adjacent CHs. A hybrid approach is used for load balancing, where the 2D mesh is partitioned into 1D array by row and by column. Two scans are used in sequence: One for all the rows, followed by the other for all the columns

a) The scan procedure.

Within each row and column, the scan operation is used to calculate the average load and then to determine the amount of overload and underload in clusters. Load is shifted from overloaded clusters to underloaded clusters to achieve a balanced state.

The scan procedure for a row is as follows. It is first initiated from one end of the row to another and then from the other end back to the initial end (second scan). After the first scan, the average number of nodes for each cell in a balanced state can be obtained. After the second scan, each cell can identify whether it is *overloaded* or *underloaded* or *neutral*.

Let w_i denote the load of a cell *i*, and let *vi* denote the prefix sum of the first *i* cells,

i.e., $v_i = \sum_{j=1}^{i} w_i$.

After the first scan from left to right, the right end of the row computes the *average load* of a cell as $\bar{w} = v_n/n$.

In the second scan from right to left, each cell compares its load w_i with w. Specifically, if $w_i = w$, then cell *i* is in a ***neutral*** **state**; if $w_i > w$, then cell *i* is *overloaded* and in a ***give*** **state**; if $w_i < w$, then cell *i* is *underloaded* and in **a *take* state**.

A node in a ***give state*** cell needs also to determine the number of sensors to be sent to each direction. Let $w_i^{\rightarrow}$ denote the number of sensors to be sent from a give state cell *i* to the right direction, and $\overleftarrow{w_i}$ to the left direction. They are computed as follow

$$w_i^{\rightarrow} = \min\{w_i - \bar{w}, max\{v_i - \bar{v}_i, 0\}\}, \qquad (2.15)$$

$$\overleftarrow{w_i} = (\mathrm{w_i} - \bar{\mathrm{w}}) - w_i^{\rightarrow} \qquad (2.16)$$

where $\bar{v_i} = i\,\bar{w}$ is the prefix sum in the balanced state.

b) Dispatch algorithm

After determining $w_i^{\rightarrow}$ and $\overleftarrow{w_i}$, a simple sender-initiated node dispatch algorithm then is executed to dispatch mobile nodes from overloaded cells to underloaded cells as follows:

1. For each cell *i* in the give state, the cell head sends $w_i^{\rightarrow}$ sensors to its right neighbor and $\overleftarrow{w_i}$ sensors to its left neighbor.

2. For each cell j in the take state, when the cell head senses several bypassing sensors, it intercepts as many sensors as possible to fill its deficiency, $\overline{w} - w_j$, and let the superfluous sensors move along their respective original direction.

This approach is simple in that a cell in the take state needs not to distinguish how many sensors should be taken from its right side and how many taken from its left side. The scan algorithm can be illustrated by an simple example of a 2D scan is presented in Figure 2. 3 (a) is the initial unbalanced state. After the first row scan from left to right, the average load of the third row is $\overline{w} = 4$, and only the cell (1,1) is in the ***give state***. Furthermore, after the second row scan from right to left, it computes $w_1^{\rightarrow} = 3$. After applying the sender-initiated node dispatch algorithm, all rows achieve the balanced state, as shown in Figure 2. 3 (b). The same procedure is then applied to all columns to achieve balanced columns, as shown in Figure 2. 3 (c).

The 2D scan discussed previously works only when there is no hole. Otherwise, certain rows and columns may not be connected. In areas with holes, a pre-processing is performed for planting "seeds" in holes at each 1D scan. These seeds will serve as CHs in the holes.

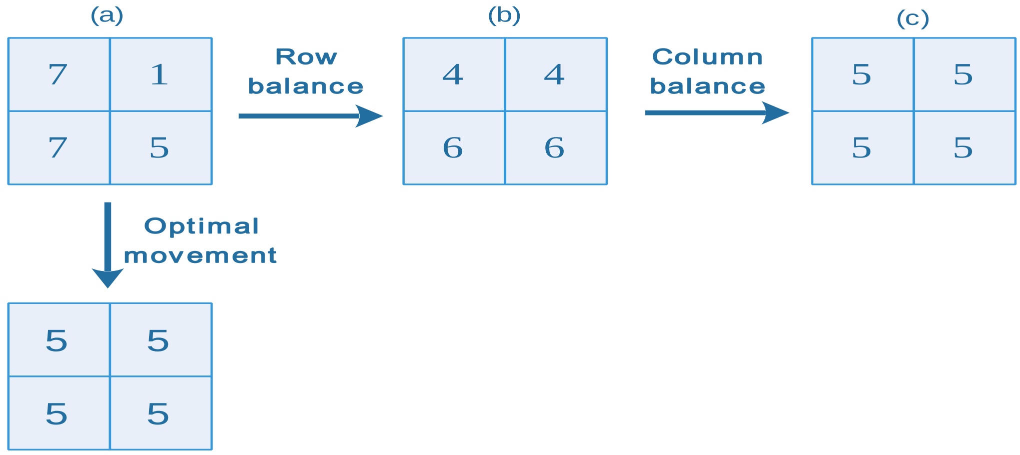

Figure 2. 3: 2D SMART algorithm

2) Discussion and critics

- ☑ SMART can be effective when used in relatively dense sensor networks as a complement for the existing sensor deployment methods.
- ☑ SMART achieves a more balanced state than other sensor deployment methods in unevenly deployed sensor networks.
- ☑ SMART needs a few rounds, which are bounded by eight, for load balancing. In a sparse network, SMART may need more rounds to achieve a balanced degree,
- ☑ When the number of deployed nodes is less than $4n^2$, with n rows/n columns, the performance of SMART is reduced, since more rounds are needed, and a balanced final state cannot be achieved.
- ☑ This approach requires the network to be dense enough so that load balancing can be proceeded in the entire sensory field, what may generate huge message overhead.

2.4.3. The flip based sensor mobility solution

In [**45**,**44**] Chellappan et al. argue that a kind of mobile nodes can only flip a limited distance to a new location and can only flip once. The initial deployment of such flip-based sensor nodes may not provide optimal coverage of the sensor field. Therefore, the objective is to design an optimal flip plan such that after nodes flip, the area coverage can be maximized, and the total number of flips is minimized. However, finding such an optimal flip plan is not trivial.

1) Protocol description

The simple example taken from [**44**] illustrates that the flip plan B obtained optimizes both coverage and the number of flips. As shown in Figure 2. 4, the flip plan A cannot cover all cells; however, it only requires five flips. The flip plan B as shown in Figure 2. 5 can cover all cells, but it requires more flips.

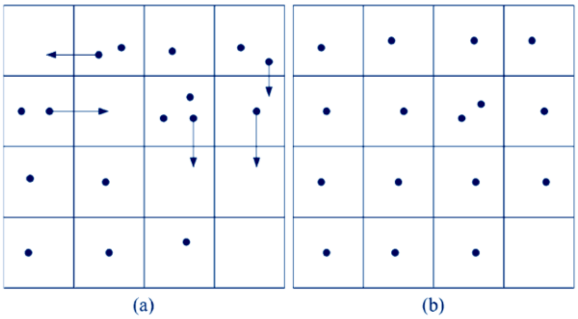

Figure 2. 4: Illustration of flip plan A: (a) Initial node deployment and flip plan; (b) The resulting deployment, the right bottom cell not covered. The total number of flips is 5

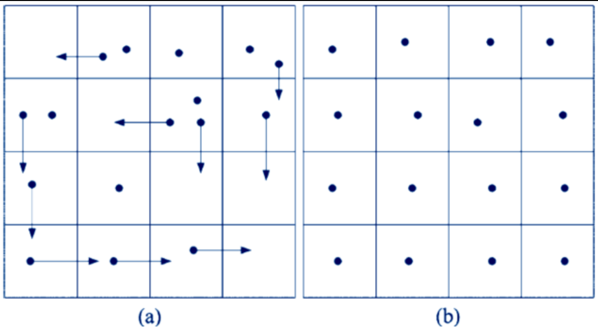

Figure 2. 5: Illustration of flip plan B: (a) Initial node deployment and flip plan; (b) The resulting deployment, all cells covered. The total number of flips is 10

Chellappan et al. translate the problem of finding an optimal flip plan into a minimum-cost maximum-flow problem. A cell is called a source if it has two or more nodes; a cell is a forwarder if it has only 1 node; and a cell is a sink if it has no node. The maximum flow problem is to maximize flows from multiple sources to multiple sinks in a graph without violating capacity constraint of any graph edge. Each flow in the graph denotes a flip path from a source to a sink, and the maximum flow value denotes the maximum number of empty cells that can be filled. To minimize the number of total flips, each flip is also associated with a cost. Then the minimum-cost maximum-flow problem is to find paths that minimize the overall cost while still maximizing the flows. Many existing algorithms can be used to solve a minimum-cost maximum-flow problem in a graph.

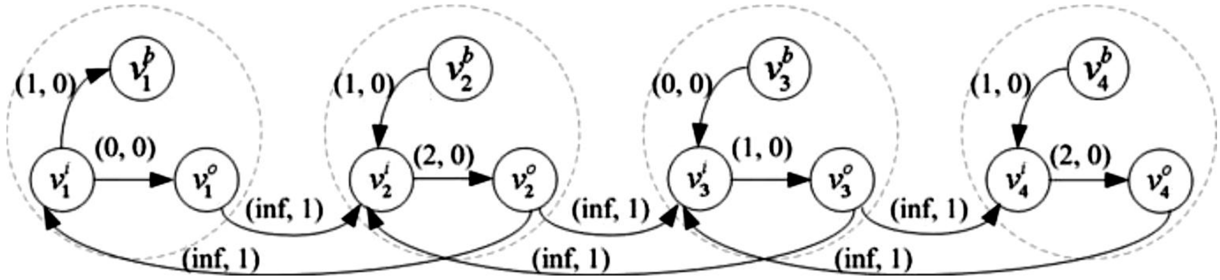

Figure 2. 6: An illustration of a virtual graph for the four cells in the top row of Figure 2.11. The (capacity, cost) is shown for each edge

a) Virtual graph Construction

Chellappan et al. propose to construct a virtual graph as follows. Each cell is represented by three vertices: The *base* vertex (v^b) tracks the number of nodes in a cell, the *in* vertex (v^i) tracks the numbers of nodes that have flipped into this cell, and the *out* vertex (v^o) tracks the number of nodes that have flipped out of this cell.

Edges are then added in order to encourage nodes to flip from source cells to sink cells. Suppose that there are *n* nodes in a cell at the initial network deployment. For the three vertices in one cell, an edge $\langle v^i, v^o \rangle$ is added with capacity *n* from vertex v^i to , v^o, and an edge $\langle v^0, v^i \rangle$with infinite capacity *inf* is added. If *n* > 0, an edge $\langle v^b, v^i \rangle$with capacity *n* − 1 is added, and if *n* = 0, an edge $\langle v^i, v^b \rangle$with capacity 1 is added. For vertices in different cells, if a node can flip from a cell *i* to cell *j* , then an edge $\langle v_i^0, v_j^i \rangle$ with infinite capacity *inf* is added. The cost of an edge needs to represent the number of flips. Therefore, a cost of 1 is added for the edges connecting two cells (e.g., $\langle v_i^0, v_j^i \rangle$) , and a cost of 0 is added for all other edges. Figure 2. 6 illustrates a virtual graph constructed for the four cells in the top row in Figure 2. 4.

b) Determining the minimum cost maximum flow plan

After constructing the virtual graph, determining the minimum cost maximum flow plan in a graph is a two step process:

1) The first step is to determine the maximum flow value from all base vertices of source cells (sources) to base vertices of sink cells (sinks) in the graph.
2) Then, the second step is to determine the minimum-cost flow plan achieving such maximum flow value.

Many existing algorithms can be used to find minimum-cost maximum-flow paths. For example, the Edmonds–Karp algorithm can be used to find the maximum flow value, and the successive approximation algorithm can be used to find minimum-cost flow plan. Each flow path specifies a flip path. For example, a flow path $\langle v_i^b, v_i^i, v_i^0, v_k^i, v_k^0, v_l^i, v_l^0, \dots, v_m^0, v_j^i, v_j^b \rangle$ denotes the flip sequence $\langle i, k, l, \dots, m, j \rangle$where one node should flip from cell *i* to cell *k*, one node from *k* to *i*, etc.

Once the base-station determines the flip plan, it will forward instructions to the sensors (that need to flip). For each sensor, the base-station can forward instructions on the reverse direction of the original path of communication between the sensor and the base-station. The forwarded packet contains the destination of the sensor and the intended region the sensor needs to flip to. Since sensors know the regions they reside in, they can determine the direction of the intended region (i.e., left, right, top, or bottom region). All sensors assumed that are equipped with *steering mechanisms* that allow sensors to orient themselves in an appropriate direction prior to their flip.

2) Discussion and critics

- ☑ Chellappan et al. translate the problem of finding an optimal flip plan to optimize coverage into a minimum-cost maximum-flow problem.
- ☑ The solution use the flip-based sensor mobility model, where sensors can flip (or hop) only once to a new location and the flip distance is bounded. The mobile nodes can only move within a limited distance flip (hop)
- ☑ Increasing flip distance achieves better coverage and reduces the number of flips required per unit increase in coverage. However, such improvements are constrained by initial deployment distributions of sensors due to the limitations on sensor mobility.

2.5. Comparative study of Area Coverage Protocol

Table 2. 1: Comparative Study of Area Coverage Protocols

	Criteria / Protocol	Network Type	Protocol type: *Distributed*	Protocol type: *Localized*	Protocol type: *Centralized*	Location information	Neighbor discovery phase	Key Goals	Advantages and disadvantages
Solutions based Node scheduling strategy	CDSC [**34**]	Static			✓		✓	maintaining connected area coverage	☺ MCDSs constructed guarantee both connectivity and coverage of the monitored area. ☺ location independent protocol ☹ Centralized Construction of MCDSs become expensive when the network size increases ☹ High communication cost
	DCovPDS [**35**]	Static	✓	✓				Energy Efficient Coverage	☺ distributed and localized algorithm based on MDS, ☺ MDSs constructed provide the coverage of area, preserve energy under an arbitrary ratio of sensing range and transmission radii ☺ location independent protocol , low communication overhead ☹ Covering sensors may not be connected.
	CCSID [**36**]	Static			✓		✓	maintaining connected area coverage	☺ MCDSs constructed guarantee both connectivity and coverage of the monitored area. ☺ location independent protocol ☹ Centralized Construction of MCDSs become expensive when the network size increases
	BCP [**37**]	Static	✓					Energy Efficient Border Coverage	☺ distributed algorithm based on MDS, and the concept of Voronoi Polygon ☺ an improved version of DCovPDS protocol that preserves both border and area coverage ☺ MDSs constructed provide the coverage of area, preserve energy under an arbitrary ratio of sensing range and transmission radii ☺ location independent protocol , low communication overhead ☹ Covering sensors may not be connected. ☹ Need more active nodes to overcome the holes appearing in the AI.
	OGDC [**38**]	Static	✓			✓		Energy-Efficient Connected Coverage	☺ distributed algorithm for synchronous WSN ☺ Maximize coverage and minimize the overlap of sensing areas of all sensor nodes, residual energy considered ☹ location aware protocol, fixed coverage radius, when Rc >= 2Rs. ☹ Work only for dense networks in order to be worth the benefits of overhead ☹ cannot preserve the original sensing coverage completely

		Protocol with no neighbor discovery [**33**]	Static		✓		✓		maintaining connected area coverage	☺ Localized algorithm for synchronous WSN ☺ maintain connected area coverage under an arbitrary ratio of sensing range and transmission radii ☺ work for sparse and very dense WSN, very small communication overhead. ☹ location aware protocol
		Coverage through Hilbert Mobile Beacon [**5**]	Static	✓			✓		Energy-Efficient Connected Coverage	☺ Use a single mobile beacon node to help static sensor nodes to locate themselves ☺ Localization of all nodes as well as the total area coverage is garneted in same task. ☺ work for sparse and very dense WSN. ☹ location aware protocol, fixed coverage radius, when Rc >= 2Rs. ☹ Computation expensive
Solutions based Node movement strategy	Virtual force	VFA [**43**]	Mobile			✓	✓		Increasing coverage after random sensor deployment	☺ Cluster based algorithm, combine the ideas of potential field and disk packing ☺ Only local knowledge, no need of centralized control and localization, flexibility, negligible computation time ☺ VFA can be easily extended to heterogeneous sensors. ☹ Possible coverage holes near the obstacles, computation expensive ☹ VFA sets a maximum moving distance of each node
	Grid structure	SMART [**47**]	Mobile	✓		✓	✓		Grid Load balancing , coverage and connectivity	☺ Distributed and scan-clustering based protocol, improved convergence rate ☺ Filling holes is based on seed-planting process (preprocessing) ☹ Work for dense networks, message overhead
		Flip based sensor mobility solution [**44**]	Mobile	✓			✓		coverage with a small number of flips with maintaining connectivity	☺ Resolve min-cost max-flow problem to provide an optimal movement plan for sensors to achieve better coverage with a small number of flips ☺ Flip based sensor mobility model can achieve better coverage while increased flip-distances ☹ In practice, it is quite likely that the mobility of sensors is limited

2.6. Conclusion

Recent research issues about static coverage and dynamic coverage are described as well as the solution to optimize area coverage.

Several studies have focused on developing solutions to ensure coverage in static and mobile sensor networks, and delivering results that are greatly beneficial to the network performances.

In this chapter, we have classified them on two big categories. The first is node activity scheduling strategy is a very efficient solution of energy conservation to coverage problem in WSN. The second was about node movement strategy for mobile WSN, where the objective is to leverage mobile nodes to control network coverage. We have detailed some solutions for each category.

In the next chapter we propose a localized solution based on node scheduling scheme for treating coverage problem in WSN. Then we apply sensor mobility strategy in the network to see their impact on coverage and network topology

Chapter 3

The Proposed Solution

3.1. Introduction

We use network coverage control to refer to the network-wide control of individual nodes' sensor unit. The sensor unit of a sensor node, which is to perform the sensing task and to produce sensing data, is controllable to be active or inactive; a significant energy saved and system lifetime can be prolonged correspondingly. Therefore, if we initially deploy a large number of sensors and schedule them to work alternatively.

For sensor networks containing mobile nodes, the node movement strategy has been used where the objective is to leverage mobile nodes to control network coverage. Mobile nodes change network coverage characteristics via moving to the desired locations.

In this chapter, we design a novel distributed algorithm based on both Node scheduling and node movement strategies to treat the area coverage problem in sensor networks in presence of mobile nodes without obstacles, called CBNMS (Coverage Based Node Movement Strategy).

Our algorithm uses a square grid-based movement technique, it divides the target area into some grids USs, and these grids are obtained following the Hilbert trajectory with a specific order; to ensuring the coverage and connectivity requirement. Our proposed solution is localized. No neighbor discovery and Sensor's locations information is needed.

3.2. Area coverage problem and Motivations

The network coverage refers to the coverage relation between field-wide points and network-wide sensors. Sometimes, we may regard network coverage as a collective measure of the quality of service provided by a network of sensor nodes at different geographical locations. The redundancy is exploited to increase system lifetime Therefore, if we initially deploy a large number of sensors and schedule them to work alternatively. A significant amount of energy savings can be achieved and system lifetime can be prolonged correspondingly [**15**].

For sensor networks containing mobile nodes, mobile nodes change network topology and network coverage characteristics via moving to desired geographical locations, where the primary objective is to maximize the coverage of these mobile nodes [**3**].

As a short summary, the motivations and objectives for network coverage can be summarized as to reduce communication and moving costs, conserve node energy consumption, and prolong network lifetime while guaranteeing the specified coverage requirements.

3.3. Grid based movement strategy

Grid based movement strategy is an easy method for facilitation of data aggregation and routing, where the cost of organizing sensors into grids is low. The sensor field is partitioned/ into many small grid cells and the number of nodes in each cell is considered as the load of the cell. A target cell is specified for a moving node, and this node can move to any location within the target cell.

The objective of node movement is to achieve a load balanced state, mobile nodes in over-loaded cells should move to underloaded cells. Furthermore, nodes should move in an efficient way so that the total moving distance, the total moving energy consumption, and/or the moving delay can be minimized. An optimal movement with minimizing the total moving distance can be obtained, however, normally with a centralized approach [**3**].

The size of a grid cell should be determined according to the coverage and connectivity requirement. To ensure complete coverage of a grid cell by any sensor within a grid cell, the side length of the grid cell should be $l \leq R_S/\sqrt{2}$. To ensure that any node within a cell can directly communicate with any other node in its four adjacent cells, the transmission range of a node should be at least the diagonal of the rectangle constructed from two adjacent cells, and hence $l \leq R_c/\sqrt{5}$. Therefore, the cell size can be chosen as$l = \min\{\frac{R_S}{\sqrt{2}}, \frac{R_C}{\sqrt{5}}\}$.

3.4. Defining Hilbert curve trajectory

Hilbert curves pass through every point in ad-dimensional space once and only once in some particular order. The idea is to graphically express a mapping between one-dimensional values and coordinates of points. We consider in this work a two-dimensional space.

3.4.1. Hilbert space-filling curve

Much of the popularity of space-filling curves is due to the geometric construction given by the great German mathematician David Hilbert in 1891 [**50**]. His basic idea was that if the unit interval should fill the whole of S, then one-fourth of I will fill a corresponding sub square of S of area 1/4 with continuity in neighboring squares. Next, I and S can be replaced by an interval of length 1/4 and a sub-square of area 1/4, respectively, and the process can be repeated. Thus for each n>1, I and S are subdivided into 4^n closed subintervals and 4^n closed sub-squares, respectively. The first four stages are shown in Figure 3.1. At each stage, the centers of the sub-squares are joined by consecutive straight lines in the manner indicated in same figure. This procedure defines a sequence of continuous functions from I into S. Since the lengths of the sides of the sub-squares tend to zero, this sequence converges uniformly to a limit function which is therefore continuous.

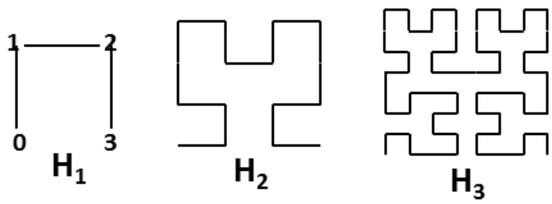

Figure 3. 1: Hilbert curves of order 1, 2, and 3 [51**]**

Hilbert curve is a continuous fractal space-filling curve, it is a one-dimensional curve pass through every point in a d dimensional space once and only once in some particular order. One of the most desired properties of such linear mappings is clustering; intuitively, this property describes the capability of the mapping to represent points that are “close” in the multidimensional space “closely” in the linear space as well. Our choice of using Hilbert curve is due to its superior clustering properties. Studies in the literature have shown that under most circumstances, the linear mapping based on Hilbert space-filling curve outperforms the others in preserving locality. [**52**]

3.4.2. Hilbert Curve Ordering

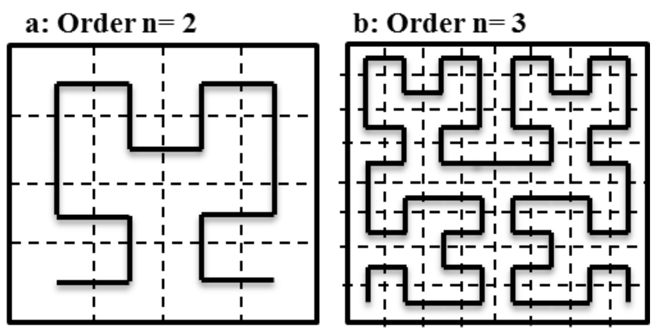

Figure 3. 2: Different orders for the Hilbert space filling curve

As shown in Figure 3.2, the basic curve is said to be of order 1. To derive a curve of order i, each vertex of the basic curve is replaced by the curve of order i-1, which may be appropriately rotated and/or reflected to fit the new curve .Hilbert curve, is generally described as the limit of a series of curves. The path of Hilbert curve imposes a linear order, which can be calculated from the first end of the curve and following the path to the other end [**5**].

3.4.3. Hilbert-keys or h-keys

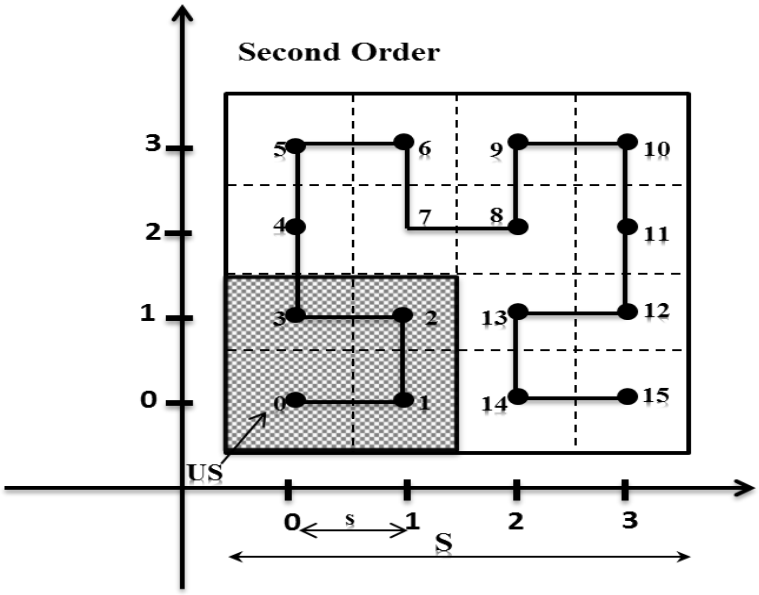

Figure 3. 3: Hilbert curves of order 2 with Hilbert keys [from 0 to 15] [15**]**

They are defined as points order in the linear ordering going from 0 to 4^n-1, where n is the order of the Hilbert curve. The basic Hilbert curve for a 2×2 grid, denoted H1, is shown in first order in Figure 3.1. The procedure to derive higher orders of the Hilbert curve is to rotate and reflect the curve at vertex 0 and at vertex 3.

Figure 3.3 also shows the Hilbert curve of order 2 with Hilbert keys from 0 to 15. We define the scale of the grids "s" as the distance between two consecutive Hilbert-keys. The grid Side Size S=s× 2^n is the length of the grid side [**51**].

3.4.4. Unit Square (US)

A square grid is a grid formed by tiling the plane regularly with squares [**51**]. A unit square is a sub-square-section of the grid where it encloses the basic Hilbert curve with four consecutive h-keys. The first is always divisible by 4. Hilbert curve of order n divides the two dimensions space into 4^{n-1} US. Figure 3. 3Figure 3.3 shows an example of a grid that encloses a 2 order Hilbert curve with four unit squares and each US encloses the basic Hilbert curve with four consecutive h-keys as follow: ({0, 1, 2, 3}, {4, 5, 6, 7}, {8, 9, 10, 11}, {12, 13, 14, 15}). A US is represented by the shaded area. [**5**]

Table 3. 1: Hilbert Curve Notations

Symbol	**Description**
N	The order of the Hilbert curve
s (Scale)	The distance between two h-keys (the scale)

$S= s \times 2^n$	The grid side size
US	The unit square , The number of US equal to 4^{n-1}
Hilbertkeys(h-Keys)	Hilbert keys from 0 to 4^n-1

In summary, a Hilbert- curve of order m divides the two dimensions space into 4^{n-1} US and connecting the centers of the US using 4^n line segments. In the rest of this dissertation; the Hilbert Curve notations as illustrated in Table 3. 1, where: m, s and S denote the order of the Hilbert curve, the scale of the grid and the grid side size ($S= s \times 2^n$), respectively. The path of the Hilbert curve imposes a linear ordering, which may be calculated by starting at one end of the curve and following the path to the other end.

3.4.4.1. Calculation of the average radius of gravity' center GC in each grid US

We call the average radius [**50**] in each grid cell US:"$\mathrm{AvgDist}_{GC}$", the average of the distances between the Gravity center GC and the h-keys vertices of US. It has been calculated (Figure 3.4) by applying Pythagoras' theorem as follow:

$$\mathrm{AvgDist}_{GC} = \sum_{j=4*i}^{4*i+3} dist(CGi, h-keyj) \div 4 = \sqrt{2}\, s\,/2 \qquad (3.1)$$

We assume that for each US an order of activation for its nodes has been assigned during location phase. The order of activation is ordered from the nearest node to GC of US to the farthest one to achieve an optimal precision and accuracy of node's location.

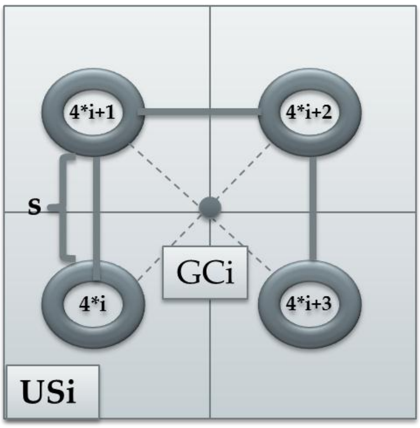

Figure 3.4: Center of gravity "GC" average distance with h-keys

3.4.4.2. Hilbert Trajectory Order Choice

In the group of grid based node movement strategies, the size of a grid cell should be determined according to the coverage and connectivity requirement. As mentioned in Bang et al. [**3**], To ensure complete coverage of a grid cell by any sensor within a grid cell, where at the same time connectivity with any other node in its four adjacent cells is maintained, the side length *"l"* of the grid cell should be $\leq R_s/\sqrt{2}$; and as well $\leq R_c/\sqrt{5}$ respectively.

More formally, a node is covering a grid cell US by any sensor within a grid cell iff:

$$\mathbf{l \leq R_s/\sqrt{2}} \qquad (3.2)$$

From Equation1, we deduce the value of s as follows:

$s = \frac{l}{2} \leq \frac{R_s}{2*\sqrt{2}}$ (3.3)

On the other hand, we have:

$S = s * 2^n = \max(Height, Width)$ (3.4)

From the two equations above, we deduce the order of Hilbert trajectory *n* as follows:

$n \geq \left\lceil log_2\left(\frac{2\sqrt{2}\max(Height,Width)}{R_s}\right)\right\rceil$ (3.5)

Equation 3.5 is the same equation derived from J.M. Bahi et al. in [**5**] , which done a proper choice of the order of the Hilbert to guarantee a whole coverage of the initial covered area, where at the same time connectivity is maintained.

3.5. System Model

The system consists of a finite set of N sensors. All nodes are uniformly distributed on a target area. This grid (see Figure 3. 5) is obtained following the Hilbert trajectory with a specific order n=3. The Hilbert trajectory divides the target area into square grids USs, each one has a Rendezvous Node RN and a set of sensor nodes, nodes are located within the same US are redundant nodes, as shown in gray color.

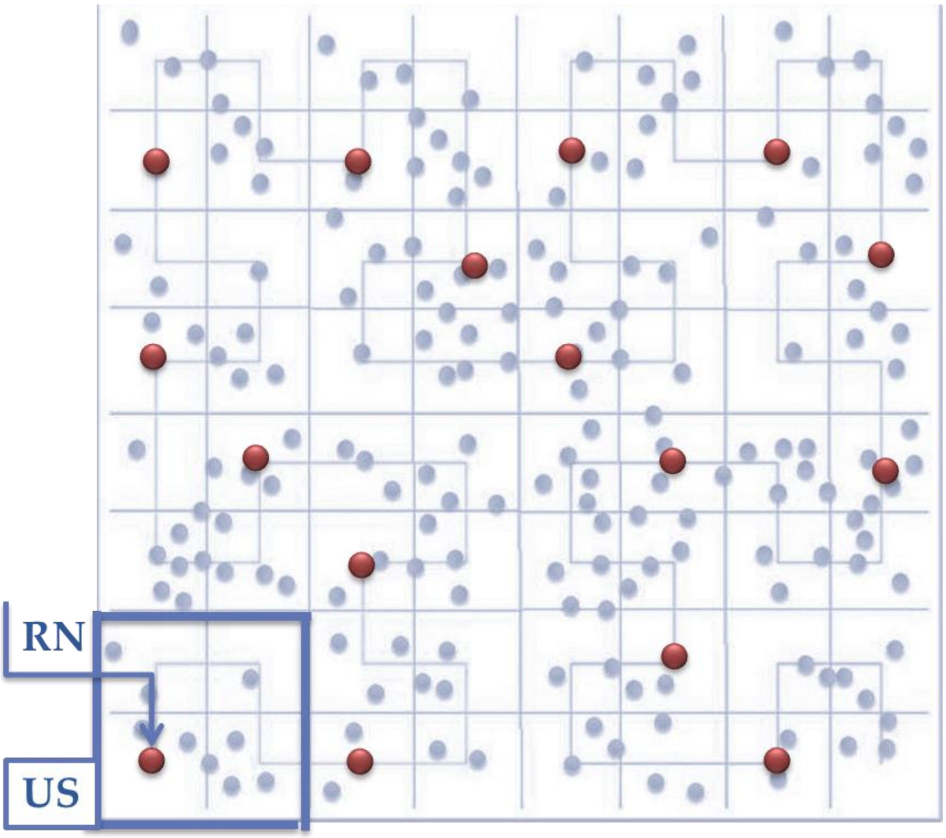

Figure 3. 5: Nodes are uniformly distributed on a square grid with 16 US

3.5.1. The neighbors of each Region US

The whole network is defined as a square grid and each cell of the square represents a region US in the WSN. The neighbors of each US are defined according to Neumann neighborhood scheme [**53**], in which every region US has less than four neighbors, while corner's US has two neighbors.

The target field in Figure 3. 6 is partitioned into 16 grids US according to Hilbert curve of order n=3.We can observe that the von Neumann neighbors of each Region US with pair number US has less than four neighbors with impair US number, and vice versa. The Von Neumann neighborhood shown includes the central US of number "8" and the four cells {7, 11, 9, and 13} that are horizontally and vertically adjacent to it.

5	6	9	10
4	7	8	11
3	2	13	12
0	1	14	15

Figure 3. 6: A region US and its four Von Neumann neighbors

We suppose that each grid header RN has array, which contains Von Neumann neighborhood of its US, with their load.

3.5.2. Rendezvous Node RN

Upper layer of WSN consists of resource-rich supernodes overlaid over the sensor network. Supernodes can have two radio transceivers, one for communication with sensor nodes and the other for communication with other supernodes. Wireless communication links between supernodes have considerably longer range and higher data rates, allowing supernode network to bridge remote regions of the interest area. Supernodes are more expensive, and, therefore, fewer are used than sensor nodes.

Each region US has exactly one super-node called Rendezvous "RN". Which are placed in the place of h-key divisible by 4; during location phase as a grid header RNs have more power reserves and better processing and storage capabilities than sensor nodes.

3.5.3. Rendezvous' nodes communication

The Rendezvous node RN is able to directly communicate with Sensors of its Region.

The Rendezvous node RN is able to directly communicate with its Rendezvous nodes neighborhood as shown in Figure 3. 6 which are located within its radio range.

3.5.4. Sensor Nodes

Each sensor node in the WSNs comprises sensing unit, transceiver, processing unit, and power supply (usually battery). We supposed that all Network sensors are mobile, where sensor nodes can be equipped with various locomotion devices that enable a sensor node to move around.

3.5.5. Sensors communication

Each sensor node is able to directly communicate wirelessly with any other node located within its radio, which is modeled by a disc diameter CR. Each node can directly communicate with only its Rendezvous node RN located within its Square grid US, where its geographic position is known by all nodes located in its US. We suppose that each node has information's about its region US (Us_{Num} : the number identifier of its region . RN_{num}: h-key value of its grid header RN. $RN_{position}$: The position of its RN), these informations are common for nodes located in same US.

3.5.6. Movement of Sensor Node

It is often assumed that a mobile node can move to a desired location without any limitation in the movement distance. In some cases, however, mobile nodes can only move within a limited distance as in [**44**]

In our case we use a fixed distance mobility model; we assume that sensors are equipped with steering mechanisms (similar to the one in [**44**]) that allow sensors to orient themselves in an appropriate direction prior to their hop. Nodes in sub-squares can freely move to other sub-squares and traveling a distance of 2*Scale; the sensor node cannot move up to once in the network in order to maximize area coverage and reduce their coverage overlaps and movement cost.

Mobile sensors can turn via RN orders with uniform speed horizontally or vertically, on Top or up down direction in a straight line for fixed amount of time. Nodes have the capability of locomotion such that the sensor moves from high dense area to low dense area.

3.5.7. System Assumptions

We assume that the target area is on a 2D plane. We assume that all the sensor nodes are homogenous with same sensing and communication range. Both ranges are assumed as disk based model. Nodes which are located within the same US (“redundant nodes”) then they are called as neighbors. Nodes are all “equal” with respect to their capabilities. Every node has capabilities for sensing, communication, processing and mobility.

Initially all node has own identity, the fact that no two nodes have the same identity, this represents both the only difference between two nodes in the network. We assume that No previous knowledge about nodes positions, all sensor nodes are not equipped with GPS system. Each node is able to directly communicate with only its grid header RN located within its Square unit. Since sensors know the regions they reside in, they move following their RNs direction to an appropriate region (i.e., left, right, top, or bottom region) with uniform speed in a straight line for fixed amount of time.

We assume that the monitored area is divided into grid of squares “US” according to Hilbert trajectory, the condition (3.5) of the previous section must be imposed on the Hilbert curve order to ensure coverage of the entire area. Only one node per US is required to be active during a round to ensure a whole coverage of the covered area, where at the same time connectivity is maintained.

We assume that, in each grid "US" of Hilbert curve, the distance vector ($Dist\_V_{id}$) between each node and h-keys is known during location phase, these location information's are stored in RN of the region as location information of nodes. All rendezvous nodes know their von Neumann neighborhood of theirs regions, with their loads. All Rendezvous nodes are equipped with GPS system.

3.6. Protocol Description

Given a target sensing field with a uniform initial sensor distribution,. Initially the target area is divided into m × m equal square grids USs using Hilbert space-filling trajectory of specific order n; where $m= 4^{n-1}$, sensor nodes are uniformly distributed on grids, almost an equal number of distributed sensors in each grid. Only one node per US is required to be active during a round (e.g., one node lifetime) to ensure a whole coverage of the monitored area, where at the same time connectivity is maintained.

We use a scheduling activity to schedule the activation and deactivation of nodes' sensor units, based on the selection of an appropriate subset of sensor nodes that must remain active. Moreover; our algorithm uses a square grid-based movement strategy in order to leverage mobile nodes to enhance network coverage.

Our solution defines two distributed algorithms. The first algorithm is executed in each RN of US; to determine a local order of activation of nodes in its region and store them in a specific Queue called *QueueUS*. The order of activation based of candidate nodes to be active is the nearest to grid center to achieve optimal precision of network coverage. Then, the RN ensures that each region US has exactly one sensor node to guarantee the coverage of the entire area.

Besides, each sensor node executes their proper distributed algorithm to balance between active, passive or move task while topology changes via mobility of nodes. The main idea of our algorithm is that, at each time step, every candidate active sensor node still active while other neighbor nodes remain in Non-active state for saving energy otherwise in order to reduce their coverage overlaps; it can leave its region to low dense area neighbor and becomes in Moving state. The candidate sensor node still decreases its energy by one power unit during one time unit until Threshold value of energy, where the node becomes in close to death state. Then it sends a notification message of out of energy; end of round, to its RN to activate other node on the *QueueUS* where a new round will be started.

In order to maintain the network connectivity and to prevent the hole in the monitored region, the threshold value is chosen as the energy required for an activity period to surveillance the region whereas a new sensor located in the same US activated in the other side.

The mobility in our approach is active, which means that sensors are autonomic and intelligent enough to find their path and move. The only assistance used is the RN of US, which done the direction of node to its lowest dense neighbors (i.e., left, right, top, or bottom region)

Our mobility model is a fixed distance mobility, all nodes moved to their appropriate region via a translation geometry method ,for a fixed amount of time and for a fixed distance unit until traveling a distance of 2*scale; a distance between two sub-squares without exceed border.

3.6.1. The state transition diagram of CBNMS

In our scheduling scheme, the operation is divided into rounds as shown in Figure 3.7. Each round begins with a scheduling phase, followed by a sensing phase.

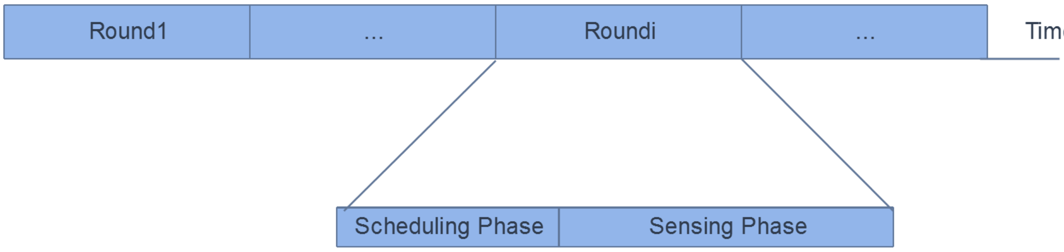

Figure 3.7: Network Lifetime

In the scheduling phase, nodes are activated or deactivated according to Broadcast Message sent from their RNs which contain candidate nodes' id .Only candidate node per US is required to be active during a round. Candidate nodes perform sensing tasks during the sensing phase. While other nodes neither turn off their communication unit and sensing unit to save energy nor switch to motion state. To minimize the energy consumed in the scheduling phase, the sensing phase should be long compared to the scheduling phase.

Figure 3. 8 shows a state transition diagram for CBNMS. At any time, a node can be in one of four different states: Running, Waiting, Moving and Close to death states.

- *Running:* A node is active and can communicate with other nodes and *sense* a field.
- *Listening:* Initially all nodes are in the listening state .A node is neither communicating with other nodes nor sensing a field. But a node *listens* to messages.
- *Waiting:* A node saves energy but a node is neither communicating with other nodes nor sensing a field. But a node *listens* to messages after a waiting timeout,
- *Moving:* A node is in motion until reach its new region to switch to the Waiting state.
- *Close to death:* A node is alive and can communicate with other nodes and sense a field, but it has not more energy to sense the field, it will turn off.

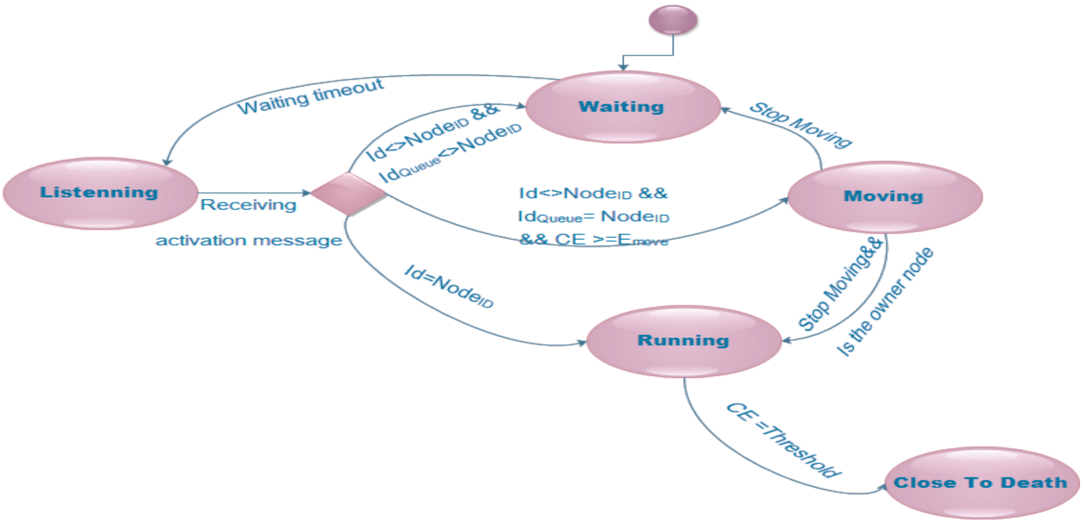

Figure 3. 8: State transition diagram of a sensor node in CBNMS protocol

Therefore, Running, Moving and Waiting states imply that the sensor node is alive. Activation message sent from each RN indicate a starting of a new round; the round is terminated when the active set are changed.

3.6.2. Detailed description of CBNMS protocol

If after uniform deployment, the monitored area is devised as shown in Figure 3.9 into subzones USs, Each Us has a grid header RN which store all node's location information and its RN' neighbors location information. Nodes are scattered and ordered into an appropriate queue *QueueUS.*

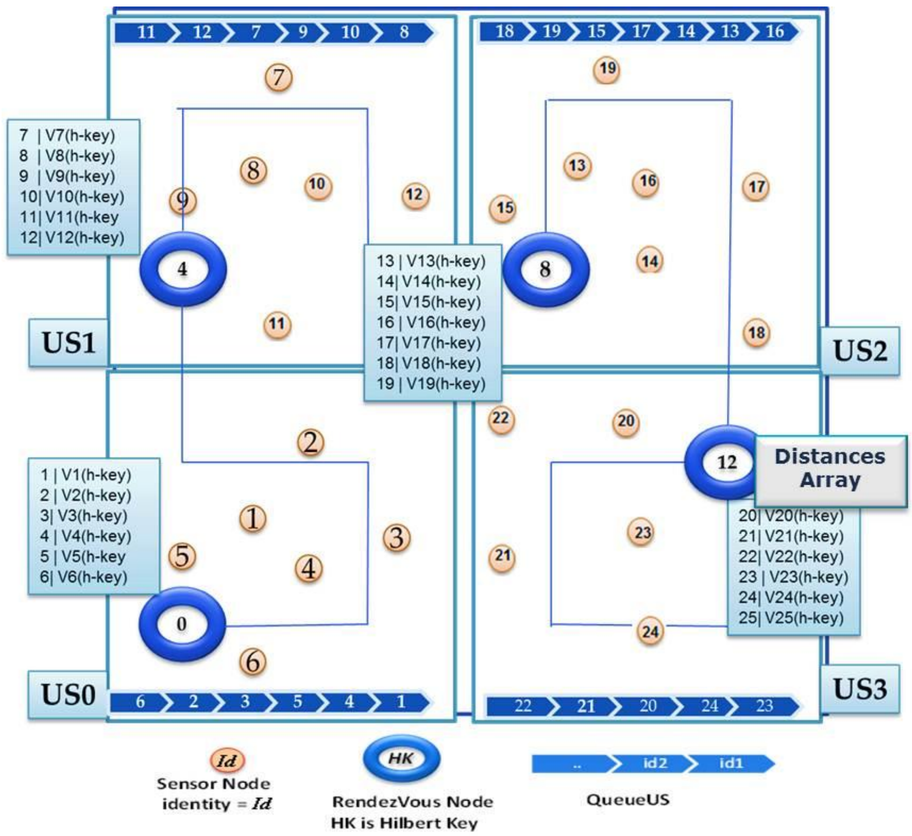

Figure 3.9: A monitored Area example; Hilbert order n=2

The algorithm we proposed uses different variables and functions discussed in the following two sections.

- ***Algorithm Variables:***

1) *Each sensor i characterized by Variables:*

- Id : Node identity, it's unique. no two nodes have the same identity in the network
- E_{init} : Initial energy, assumed to be the same for all nodes.
- CE_i: Current energy of sensor i, initially it is equal toE_{init}.
- $E_{Threshold}$: Threashold energy value; assumed to be the same for all nodes. It means sensor has no more energy to sense the field, it will turn off.
- E_{UM} : Constant Value, equal to be the energy cost for a sensor to move one unit of distance.
- E_{Move} : Equal to be the energy cost for a sensor to move from a region Us to another. $E_{Move} = E_{UM} * dist.$
- $localT_i$: Local timer initiate to 0, node i increases its timer by one after each active period when it loses one power unit.
- $TimeU_i$: The duration for sensor to lose one power unit.(Time constant assumed to be the same for all nodes)
- $waiting_{time}$: The duration of waiting sensor until switch to idle state, in order to receiving activation message from its RN, where $waiting_{time} = CE_i * TimeU_i$.
- Us_{Num} , RN_{num} and $RN_{position}$: Each sensor i has information's about its region (Us_{Num} : the number identifier of its region. RN_{num}: h-key value of its grid header RN. $RN_{position}$: The position of its RN). These informations are common for all neighbors' nodes.
- $Counter_{Migration}$: Energy migration counter of each sensor node move initiate to 0 and increment by 1 after each movement of one distance unit.
- $Moving\ Direction$: Moving direction values are (Left: 0, Right: 1, Bottom: 2 or Top: 3), these values are selected according to the underloaded region among its four region neighbors. Mobile nodes should move to underloaded cells.
- $dist_i$: A fixed moving distance equal to 2*scale, it is the same for all nodes.
- $QueueTrace_i$: The history information's of its old region before moving to another one.

2) *Each Rendezvous' node RN characterized by variables:*

- $Neighbors_{RN}[2,4]$: Array of two dimensions, it contains Von Neumann neighborhood address of each US, with their load (the number of nodes in each region).
- $QueueUs_{RN}$: A queue contains all region' nodes ordered with a specific order. where the first element is the candidate node to be active in the next round
- $QueueHist_{RN}$: A queue contains all informations of nodes which are moved, this informations concern identity of nodes and their distance vector.

➤ ***Summary of the algorithm messages***

There are three types of messages sent, messages sent from sensors to their RNs, messages sent from RNs to their sensor nodes and messages sent between RNs neighbors:

- *SendActivate(id,RN_{num},id_{queue}) :* Broadcast Message sent from RN of each Us to all nodes located in same Region Us. It's an active announcement to node of identity id and non-active/move announcement to other nodes (redundant nodes). According to *id* value the receiving sensor node (i.e, identity: $Node_{ID}$) changes its state as follow:
  - ☑ *Id equal to $Node_{ID}$:* The node is running in all cases .When the node is listening; it will switch to running state. Otherwise when the node is moving, it stays in motion. Wheras the node is running has never happened.
  - ☑ *Id different than $Node_{ID}$:* It means that one of their nodes neighbors which located in the same US becomes in active state in the next round. The node will stay in moving state. Or else the node will stay in waiting state; otherwise it will switch to Moving state if condition (3.6) is satisfied.

    $CE \geq E_{move} \textbf{ and } Node_{ID} = id_{queue}$ (3.6) where,

    id_{queue} is the last element in QueueUS, which is closest to boundary of its US.
- *OutOfEnergy($Node_{ID}$):* A notification message of out of energy ; sent from each sensor to its RN when $CE \geq E_{threshold}$ in order to activate other node on the *QueueUS* and to prevent the hole in the monitored region. The sensor switch to close to death state, it has no more energy to sense the field, it will turn off.
- *ReplyMsg($Node_{ID}$):* A notification message sent from each sensor node to its RN which is ready to receive messages. Upon $waiting_{time}$ each waiting node switch to idle state.
- *SignalMove($Node_{ID}$):* A moving announcement sent from nodes want moving to its corresponding rendezvous node RN. When the RN receives the message, it makes an update procedure by making update to all its tables and queues and stores all nodes' information's in a History Queue.
- *OkToMove($Node_{ID}$, Direction, NewUs,RN_{num}):* A moving confirmation message is sent in response to *SignalMove* message . It contains the moving direction and the New Us information's (i.e.; US location address, RN_{num} value).
- *StopMoving($Node_{ID}$, Direction):* Stop moving announcement sent from node of identity id to its new region ,which indicate that the required distance is traveled by a sensor. It's a Notification message of the arrival of new sensor node. The new RN makes an update to all its tables and queues according to the coming node id.

3.6.3. The CBNMS protocol phases

In *CBNMS*, only a subset of sensors are in working state to ensure desired sensing coverage, and other nodes are put into low-powered sleep state or move toward under-load region. Scheduling sensors to work alternatively, implicate a significant amount of energy savings can be achieved and system lifetime can be prolonged.

Network lifetime is divided into rounds, and each round has a scheduling phase. In order to starting scheduling phase a preliminary phase is precede it. This phase assigns an order of activation for its nodes locally by theirs RNs via a specific strategy; explained later; and sort those in a queue named *QueueUS*.

1) Activation Order assignment

For each square unit US, the RN attributes an order of activation of all US' nodes, then it stores them in a specific queue "*QueueUs*"; this order is determined using distance vectors of nodes $\mathbf{Dist\_V}_{id}\mathbf{[hkeyj]}$ stored in each RN during location phase. The order is shown in Figure 3. 10 and sorted as follow:

For Each RN do

1. Average out the distances of each node id located in US with its h-keys position:

$$\mathrm{AvgDist_{id}} = \sum_{j=4*i}^{4*i+3} \mathrm{Dist\_V_{id}[hkeyj]} \div 4 \qquad (3.7)$$

2. Calculate the difference's values $\mathrm{Diff_{id}}$ between the average radius of nodes $\mathrm{AvgDist_{id}}$ and average radius of GC $\mathrm{AvgDist_{GC}}$ then ; $\mathrm{Diff_{id}} = |(\mathrm{AvgDist_{id}} - \mathrm{AvgDist_{GC}})|$ (3.8)

3. Order these nodes in a special queue' called (*QueueUS*) with an increasing order of $\mathrm{Diff_{id}}$.

Finally *QueueUS* has the activation order of nodes from the nearest one to GC to the farthest one, to achieve optimal coverage precision.

All steps illustrated above run in each RNs of grid square US. This phase is considered as a preliminary phase which acts before the scheduling phase illustrate next.

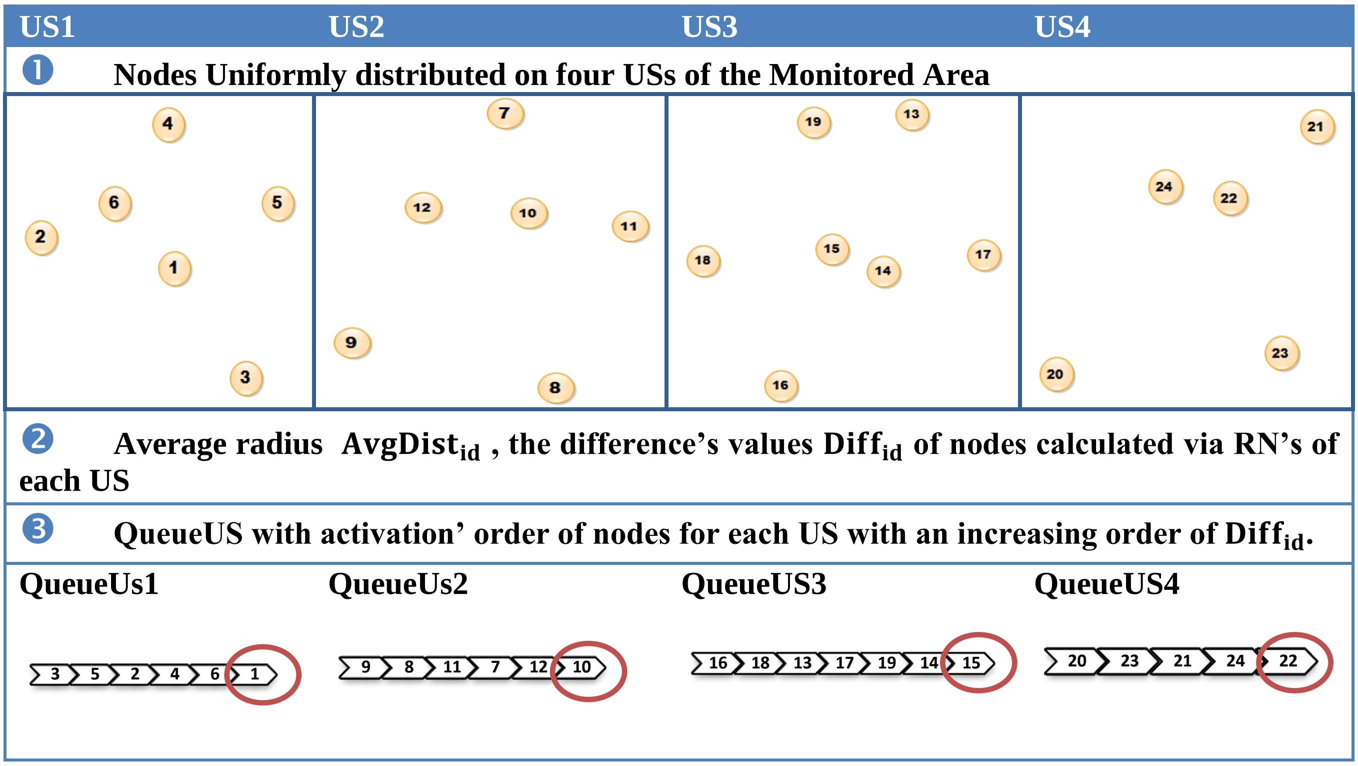

Figure 3. 10: A snapshot of a sensor network with activation Order Assignment steps

2) The Node scheduling phase

In our scheduling scheme, the operation is divided into rounds. Each round begins with a scheduling phase, followed by a sensing phase. All sensor nodes which are part of the active set are in the active state, whereas all other nodes are into a low energy, sleep state. These sets can be activated in turn by rounds.

Our scheduling method use the *QueueUS* constructed in the initialization phase in order to choice of an active node per US. Nodes are activated or deactivated according to the broadcast message *SendActivate(id,RN,*id_{queue}*)* sent from their RNs which contains the identity of candidate nodes 'to be active, which is the first element in *QueueUS*. Only one candidate node per US is required to be active during a round. Candidate nodes perform sensing tasks during the sensing phase. While other nodes neither turn off their communication unit and sensing unit to save energy nor switch to motion state.

Every alive sensor nodes is part of the active set which form a current round. This current round is terminated when its active set is changed. This active set is changed if one of their alive sensor nodes switches to other state; upon receiving an activation message from their RNs, upon stop moving when it reaches its new region, or when a node has no more energy to sense the field and will be in Close to death state.

a) Rendezvous Node algorithm process

Each region US contains a grid header RN and k-covered redundant sensor nodes that can cover same region US. In the beginning each RN of a square unit US and following the *QueueUS* order, it sends an activation message *sendActivate(Id,* RN_{num}*,*Id_{queue}*)* to all their nodes (broadcast message to all grid sensors). *Id* represents the identity of the candidate node to be active during the next round which is the first element in *QueueUS*. RN_{num} represents the identifier of grid header RN, only nodes owns the same Un_{um} receive this message; it means nodes located in the same US. Id_{queue} represents the node concerned for moving in the next round.

Then upon receiving of the activation message by sensors , a sensor which has the same identity *Id* received becomes active ,while other nodes still moving or saving energy , or decide to move to low dense US if they have same identity as id_{queue} . This message is for active announcement to node id and non-active/move announcement to other nodes (redundant nodes).

When the active sensor node reaching a *Threshold* value of energy, it means that it has no more energy to monitor its region, the node becomes in *close to death* state, then it sends an out of energy notification message *;*end of round ,to its RN in order to activate other node. The RN remove the first element of the *QueueUS* as it will turn off, then it repeats the same process again by sending the activation message to all its nodes to activate other candidate node, which is on the head of *QueueUS* ; where a new round will starts. The pseudo code of each RN is presented in *Algorithm 1*

***Algorithm 1**: Activity scheduling algorithm for each RN*

QueueUs() ; *//*a queue* contains a local activation schedule of nodes located in same US
RegionLoad **init** to QueueUs.size(); Priority ={0,1}**init** to 1 for regions of pair Us_{Num} , and **init** to 0 otherwise; Replynodes **init** to 0;

BEGIN

While(QueueUS<>null)**Do**

Id= *QueueUS* .***firstElement***() ; //Take the first element id of QueueUS ;

Id_{queue}= *Id* ; // no nodes leaving US next round

If (*Priority==1*) **then**

If (*under-loaded region neighbor* exist) **then**

Id_{queue}= QueueUS .***lastElement***(); //candidate node to move of identity Id_{queue}

If (ReplyNodes< RegionLoad) **then** Wait(); //waiting for reply of all redundant nodes

// *only nodes located in its region US receive broadcast message ;*//

Send broadcast message *SendActivate*(*Id, RN_{num}*,Id_{queue}) to one hop neighbors.

ReplyNodes=0; //Reset *ReplyNodes to zero*

If(Priority==1) **then** Priority =*0* **else** Priority=1; //Change Joining/Leaving nodes priority

End while;

If (*ReplyMsg**(Id)** message has been received*) **then** ReplyNodes++//sensor node enter in idle state

If (*OutOfEnergy **(Id)*** message has been received) **then**

Begin

QueueUS .***Dequeue*** (); //Remove the first element of QueueUS; the node will be turn off.

RegionLoad--;

Go to 1; //then select the next active node

End If;

END.

b) Sensor Node algorithm process

Each sensor node i has an identifier id. It contains information's about residual energy *CEi*, a local timer *localTi* initiate to 0, and its current state *Statei.* Sensor nodes of the network can be in one of four different states: Active (i.e., the sensor monitors, transmits and wastes energy), Passive (i.e., A node saves energy), Moving (i.e., A node is in motion) and close to dead (i.e., the sensor has litle energy and will be turned off). Therefore, active, moving and passive states imply that the sensor node is alive.

Initially, all network nodes are in *passive* state, they still in waiting state for saving energy until a global time where all sensors switch to *idle* mode, where a sensor is capable of transmitting and receiving messages, but not sensing.

Then, a sensor decides to be: active, non-active or move upon receiving an activation message *sendActivate(Id, RN_{num},Id_{queue})* from its RN. According to values of *Id* and Id_{queue} ; a sensor can decide its state in the next round. A node which has the same identity received becomes active, while other nodes becomes saving energy or decide to move toward low dense US neighbor. The pseudo code of the activation message influence in each sensor i is detailed in *Algorithm 2*.

Algorithm 2: Influence of *SendActivate(id,RN,id_{queue})* on each Node i with identity $Node_{ID}$

$State_i$ ***init*** *to passive; $localT_i$* ***init*** *to 0;Boolean isNeverMoved* ***init*** *to true; CE_i* ***init*** *to E_{init}*

BEGIN

If (activation message *SendActivate*(*Id, RN_{num},Id_{queue}*) has been received)**then**

If (RN_i== *RN_{num})* **then** //only nodes located in the same region receive this message

if (*$Node_{ID}$ ==Id* **and** *$State_i$=Passive*) **then**

Begin

$State_i$ = Active; //The node decide to be active next round, pass to *active task*

Go to active task;

End if;

Else if *($Node_{ID}$ ==Id* **and** *$State_i$==Move*) **then** *Statei* remains unchanged

Else if (*$Node_{ID}$* <>Id **and** $State_i$==Passive) **then**

if (*$Node_{ID}$* == Id_{queue} **and** $CE \geq (E_{threshold} + E_{Move}$)**and** *isNeverMoved=true*) **then**

Begin

$State_i$ = Move ; //The node decide to move next round , ***pass to moving task***

Go to Moving task;

Else

Begin

Statei remains unchanged ; //the node stay in waiting state

$waiting_{time}$ =CE_i*$TimeU_i$; //calculate saving energy delay

Go to saving Energy;

End if;

Else if *($Node_{ID}$ <>Id* **and** *$State_i$==Move*) **then** *Statei* doesn't change;

End if;

End If;

Each sensor node performs different tasks which are sensing task, saving energy task and moving task, which are detailed as follow:

a) At each active task (see Algorithm 3), every active sensor node works one unit of time, loses its energy "*CEi*" by one, and the local timer "*localTi"* increases by one active task. This procedure repeated until sensor node runs out of energy. When an active node reaching a *Threshold* value , it means that it has no more energy to monitor the region US, the node becomes in *close to death* state. Then it sends a notification message *OutOfEnergy($Node_{ID}$)* to its RN, in order to activate other node to sense the US field; where a new round will start.

b) Each passive sensor still in waiting state for saving energy until reaching $waiting_{time}$, then it switch to listen mode in order to will be capable receiving and processing messages. It sends *ReplyMsg($Node_{ID}$)* to its RN, that it is ready to receive activation message announcement or a migration decision from its RN.

c) For moving task; we distingue three operations type which explained later.

Algorithm 3: *Pseudo-code for Sensing Task of each Node i*

```
Active Task();
Repeat{
   Sensing the field for one TimeUi.     //consumes 1 time unit ,1energy unit in active mode
   CEi --;                  //Decreases its energy by 1;
   localTi++;               //increase local counter, number of active task of node i
until (CEi< EThreshold )                 // Threashold value is reached, end active task for node i
 Statei = 2                 // Setting node' state to close to death state;
 Send OutOfEnergy (NodeID) message to its RendezVous node RN;     //node i will turned off
```

c) Node Scheduling Scenarios

In this section we present an example of the influence of broadcast message to alternate sensor nodes activity. At the beginning all sensors are passive; each US of the monitored area is k-covered where k is the number of their sensors nodes as shown in Figure 3. 10. The example shows that id's circled in *red* on each *QueueUs* are the first sensor nodes candidate to be active.

The RN of square unit 1 sends *sendActivate(1,0,3)* message to all region' nodes. The node of identity one "1" becomes active while other nodes; node 2, node 4, node 5 and node 6 remains in non-active state for saving energy otherwise node 3 which is so far to the region center; it decides moving according to their order on the *QueueUs*.

The active set of first round represent the nodes closest to center of grid, where these identities are included in the header of *QueueUs* queues.

- ✓ At time t1, RN of US1 sends an activation message *sendActivate(1,0,3)* to all nodes of its region , The node of identity "1" becomes active while other nodes; node 2, node 4, node 5 and node 6 remain passive for saving energy otherwise node 3 which is so far to the region center; it decides to move
- ✓ At time t2, RN of US4 sends an activation message *sendActivate*(22,12,20) to all nodes of its region , The node of identity "22" becomes active while other nodes; node 21, node 23 and node 24 stay in passive state.
- ✓ At time t3, RN of US3 sends an activation message to all nodes of its region, the node of identity "15" switch to active state, node of identity "16" decides moving while other nodes stay passive.

- ✓ At time t4, RN of US3 sends an activation message *sendActivate*(10,4,9) to all nodes of its region , The node of identity “10” becomes active while node "9" switch to moving state and other nodes remain passive.

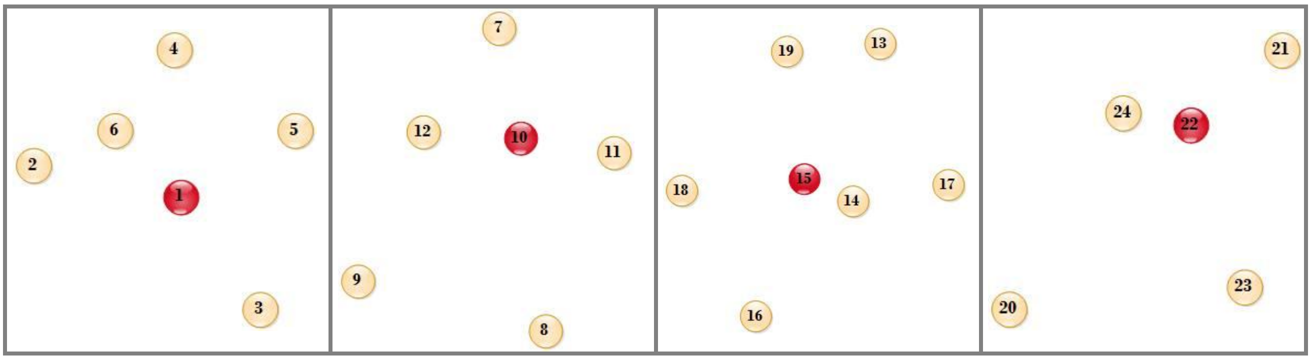

Figure 3. 11: Influence of broadcast message *sendActivate*()on scheduling activity

- ✓ At time t4, the set of active nodes colored in red in Figure 3. 11; {1; 10; 15; 22} represent the first Round; which ensure the both coverage and connectivity of all the monitored area.
- ✓ At time t5, Sensor of identity “1” has no more energy to monitor its region US1, it sends notification message *OutOfEnergy*(*Id*) to its RN; end of round. The RN removes the node of identity 1 from its QueueUs and activates the first element of the queue which becomes the “node of identity 6” by sending *sendActivate*(6, 0, 5). Where a new round are generated. The node 6 will be activated, while node1 becomes in dead-end state, as is illustrated in Figure 3. 12.

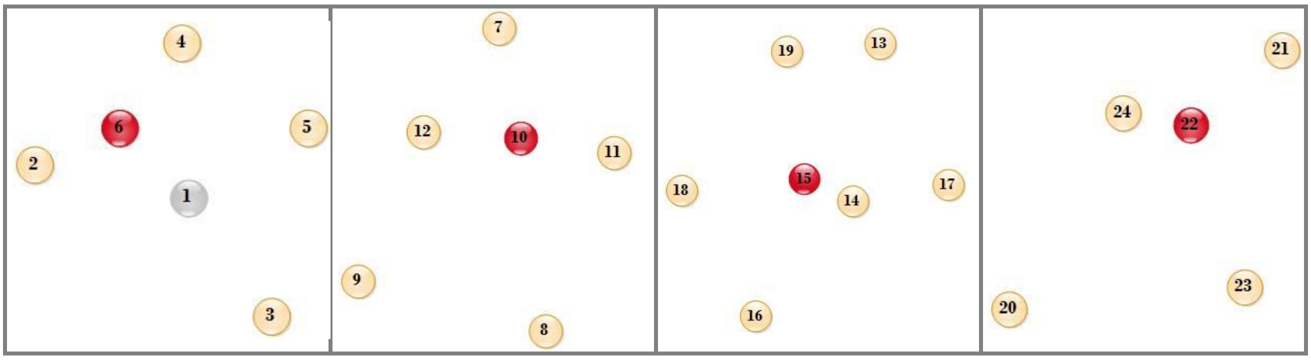

Figure 3. 12: Influence of sending OutOfEnergy(Id)message on scheduling activity, message sent from node will be turned off

In order to see the effect of mobile nodes to maximize coverage , we define a movement strategy based on sensors closest to border of each US, so nodes far to US center is the more eligible to migrate its region to achieve optimal coverage, in order to minimizing movement cost.

3) CBNMS with node movement strategy

When nodes receive an activation message *SendActivate(id,RN,id$_{queue}$)*from their RNs, only nodes that have same identity as the identity "*Id$_{queue}$*" carrier the broadcast message; are concerned to move toward low dense neighbor US. These nodes will move; if they have more energy to move; directly after receiving moving direction from their RNs. RN offers the adequate direction of node to its lowest dense neighbors .Then node will; turn to:(left, right, top or down) via a fixed distance.

According to nodes mobility there is an update process followed in our algorithms; in RN algorithm side (see *Algorithm 4*) and in Sensor algorithm side (see *Algorithm 5*). We distingue three operations type: Operations when nodes leave their region US, operations when nodes join new region US, and periodic update operations; when the region loud in terms of number of nodes have been changed.

a) Operations when nodes leave their US

Nodes closest to boundary of US are the more eligible to migrate their regions. And when receiving a broadcast message from their RNs, this represents movement permission to these nodes.

For minimizing the movement cost the sensor node with maximum energy will move and the sensor node with minimum energy stays in the region. Nodes closest to boundary of US are classing on the queue' QueueUS of their RNs, which means that they having last chance to be active, and more chance to stay saving energy. So these nodes have capabilities for movement and have a sufficient energy for moving.

Our movement method consists of migrating nodes from high density to low density grid; only RNs have knowledge about load charge of their region neighbors, according to $Neighbors_{RN}$array. Leaving update procedure work as follow:

1. When a node i want to move, it sends *SignalMove*($Node_{ID}$) message as a moving announcement to its corresponding rendezvous node RN.
2. When the RN receives the message, it makes update to all its tables and queues ;as follow:
   - ✓ The RN removes the last element of its *QueueUS*, which contains information's about node want moving with identity *Id*. It deletes the node' distance vector from its array of distances, to store them in a History Queue of the US.
   - ✓ The RN discovers the region destination adress *NewUS* of node want moving i and selects the moving direction *Direction (0: to left neighbor, 1: to Right neighbor, 2: to down neighbor, 3: to Top neighbor).* The choice of region destination depends on region neighbor having a less number of nodes among other neighbors.
   - ✓ Then the RN sends moving confirmation message *OkToMove*($Node_{ID}$,*Direction*, *NewUs*, RN_{num}) is sent in response to *SignalMove* message to node i .
3. Upon receiving of *OkToMove*($Node_{ID}$,*Direction*, *NewUs*,R_{Nnum}) message by node want moving i , it starts moving to new region US to the adequate direction, for a fixed amount of time and for a fixed distance d= 2*scale.

b) Operations when nodes join new US

Once the mobile node i joins the new US and stop moving, it sends *StopMoving*($Node_{ID}$) message to its new RN, which indicate that the required distance is traveled by a sensor ;then it changes its state to non-active for saving energy.

The Rendezvous node corresponding to this new US upon receiving *StopMoving* ($Node_{ID}$) message; as a notification message of the arrival of new sensor node; it makes an update and performs the following process:

- ✓ It registers the identity of node" $Node_{ID}$ " of the arrival node i in its region, it store $Node_{ID}$ on the QueueUS, in order to be the candidate active node next round.

Algorithm 4: *Update Movement process in RN algorithm side*

NewUS=getNewRegion(); // Function discover the address of lowest load region according neighbors array Direction=getNewRegioDirection(); // Function discover the direction of lowest load region

If (*SignalMove*($Node_{ID}$)) message has been received) **then**

Begin

QueueUS .***Dequeue*** (); //Remove the last element of *QueueUS*; the node will leave the region.

Send *OkToMove*($Node_{ID}$,*Direction, NewUs, RNnum*) to one hop neighbors,// respond to SignalMove message

RegionLoad- -;

End If;

If (*StopMoving*($Node_{ID}$) message has been received) **then** //notification message of the arrival of new node

Begin

QueueUS.***Enqueue*** ($Node_{ID}$);//Register the arrival on the QueueUS ,it will be the active node next round.

RegionLoad++;

End If;

c) *Periodic* Update Operations when nodes Leave/Join US

When a node leaves its old region US, the number of nodes in this US is changed and decrease by one. When a node joins a new Region US, the number of nodes is increased by one. When a node will lose its energy and turned off the number of nodes is decrease too. According to this change of the region load in terms of number of nodes of each region US. Each US must inform their neighbors about its new region load.

Periodic update message sent from each RN to all their Neighbors RNs. This update message sent if the number of nodes in region Us is changed (i.e; *The Region Load is changed*). We make update of *Region Load* received in $Neighbors_{RN}$ *array* before each sending of broadcast activation message.

Algorithm 5: Update Movement process in Sensor i' algorithm side

$Counter_{Migration}$ init to 0;

send *SignalMove*($Node_{ID}$) message to its RN // moving announcement to its corresponding RN

If (OkToMove($Node_{ID}$,Direction, NewUs, RNnum) has not been received) **then** Wait();

//The node Start moving to new region US to the adequate direction

Repeat *//Movement Task*

Move *to new location for one distance unit (1Scale) for a fixed amount of time*

$CE_i = CE_i - E_{UM}$ //Decreases its energy by E_{UM} ;

$Counter_{Migration}$ ++; //energy counter migration

until ($Counter_{Migration}$ == 2) ; // the fixed distance of 2 scale is traveled

send StopMoving($Node_{ID}$)message to its RN

Begin

isNeverMove=false; // sensor node move only one once

Statei = 0 //Changes its state to NON-ACTIVE for saving energy

End if

d) Management of Leaving nodes and joining nodes operations

Each region US manages operations when nodes leave their region US and operations when nodes join new region US, the management of these operations in same time is more complicated. In order to avoidance the arriving and leaving nodes in the same time we alternate between regions US for leaving operation.

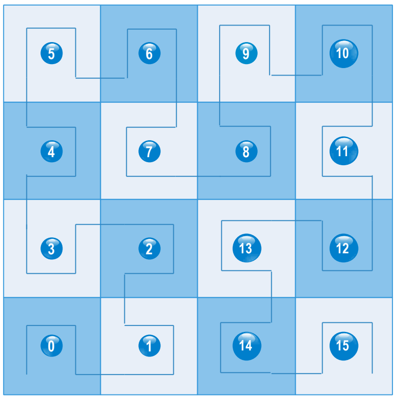

Figure 3. 13:Leave/Join operations management

The neighbors of each US ; as shown in Figure 3. 13 ; are defined according to Neumann neighborhood, in which every rendezvous node with pair number has less than four neighbors with impair number, and vice versa while corner's US has two neighbors.

According to Neumann neighborhood in the beginning we make priority to pair regions to manage leaving operations while other regions neighbors (odd ones) are responsible to receive the arrival nodes. Then upon receiving *OutOfEnergy($Node_{ID}$)* message from their active nodes. The priority will be reversed where pair regions will manage joining operation, while odd ones will manage leaving operations.

e) Movement Strategy Scenario

In this section we present an example of our solution with movement strategy, in order to see the effect of mobile nodes to maximize area coverage.

We consider a WSN where nodes scattered uniformly and densely in area of 20*20 surface units (u^2). Let l the side length of the grid region US. We suppose for all nodes the same Communication Range CR, the same Sensing Range SR; equal to 6 u.

Our objective is to find the minimum Hilbert Trajectory order n (the shortest trajectory) to guarantee a whole coverage of the monitored area, where at the same time connectivity is maintained, and we assign to scale "s" its maximum value.

Therefore, using the previous calculus, as exposed earlier in section 3.3.4.2, we obtain:

Using equation (3.3) , the maximum value of scale "s" is:

$\mathrm{s} = l/2 \leq \mathrm{SR}/(2 * \sqrt{2}) \approx 2.121u$;

And , using equation(3.5) we find the minimum Hilbert Trajectory order n to coverage of the entire area ; as follow:

$\mathrm{n} \geq \lceil 3.24 \rceil \Longrightarrow n = 3$

More formally, in order to maintain network connectivity, the connectivity is maintained within a grid US iff:

$$l \leq \frac{\mathrm{CR}}{\sqrt{5}} \Rightarrow \mathrm{s} = l/2 \leq \frac{\mathrm{CR}}{2*\sqrt{5}} \Longrightarrow CR \geq 2 * s * \sqrt{5} \Longrightarrow CR \geq 9.485$$

Figure 3. 14 (A) and Figure 3. 15(A) shows the initial sensor deployment area, the nodes are uniformly deployed into grids following the Hilbert path corresponding to Hilbert Trajectory order n=3 of the previous calculus.

Each grid header RN stores the distance estimated during localization phase between nodes is physically according to the position of the h-key in the area.

❶ ***First State :*** *Senor network with mobile nodes*

As shown in Figure 3. 14(A), each region US of the monitored area is k-covered where k is the number of their sensor nodes. In our case each region US is varied from 1-covered to 3-covred.

In the first iteration in Figure 3. 14(B), nodes upon receiving broadcast message from their RNs change their state. The active set become nodes colored in red, where their identities are included in the header of broadcast message. While other nodes stay in non-active state for saving energy.

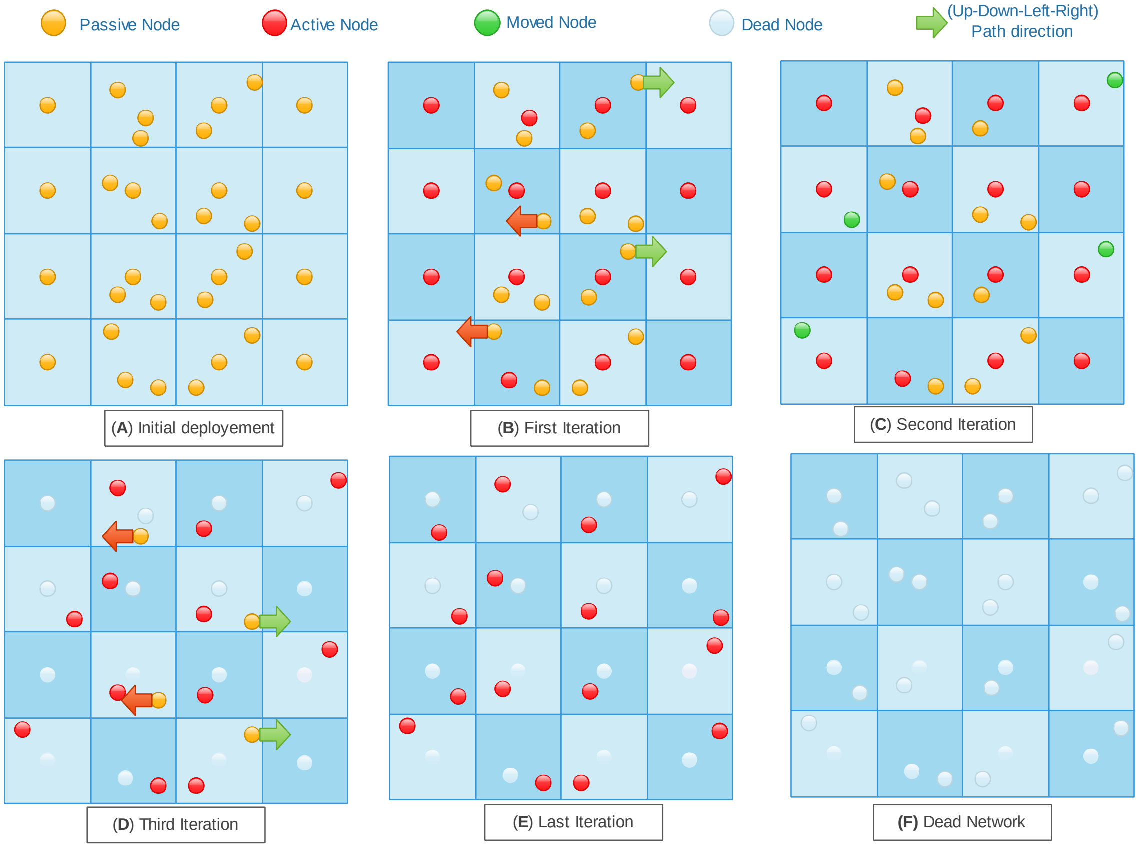

Figure 3. 14: Snapshot of the sensor network with node movement strategy

Otherwise each nodes closest to boundary of pair US (Blue square units) are eligible to migrate to under-load region following their path direction; clarified by fleches; there are four mobile nodes in our case.

In the second iteration in Figure 3. 14(C), all moving nodes colored in green; upon travelling their fixed distance and joining theirs new regions US, they stop moving and switch to waiting state for saving their energy, in order to be active next round.

The active set still sense the field until reaching a threshold value of energy; each active node has no more energy to sense its region US becomes in dead-end state; and send a notification of out of energy message ;that it will be turn off to its RN.

Next, each RNs upon receiving out of energy notification from its old active node. It still wait for all reply messages from their nodes switch to idle state , in order to repeat the activation process by sending broadcast message to their nodes located in same region US.

In the third iteration, as illustrated in Figure 3. 14(D), nodes upon receiving broadcast message from their RNs change their states. All old active nodes turned off which are colored in gray. The active set becomes nodes shown in red including the joined nodes of previous round. While each node closest to boundary of odd regions US (gray square units) is eligible to migrate to under-load region, there are four mobile nodes in this third iteration case, which are indicated by fleches.

Figure 3. 14(D), in last iteration, upon stop moving of new arrival nodes. All their destination regions are free; implicate that they will turn to active state to cover their new regions. All remained nodes form the last active set; nodes colored in red, where the whole coverage of the monitored area is guaranteed with 100 percent.

All active nodes still sense their region until reaching a threshold value, they will turn off, and becomes in dead-end state; and send a notification of out of energy message to its RN. All *QueueUs* are empty because there is no remained alive node.

The network will be dead with no alive sensor node as illustrated in Figure 3. 14(E).

❷ ***Second State :*** *Senor network without mobile nodes*

The example shown in Figure 3. 15, gives the corresponding sensor nodes scheduling of the previous example without mobile nodes.

We see that only first iteration according to Figure 3. 15(B) can cover the whole monitored area, while the second and third iteration of Figure 3. 15(C) and Figure 3. 15(D) respectively provide less coverage ratio. The coverage ratio is degraded where empty regions are generated.

So CBNMS achieves a high coverage ratio in presence of mobile nodes, which maximize the area coverage due nodes mobility.

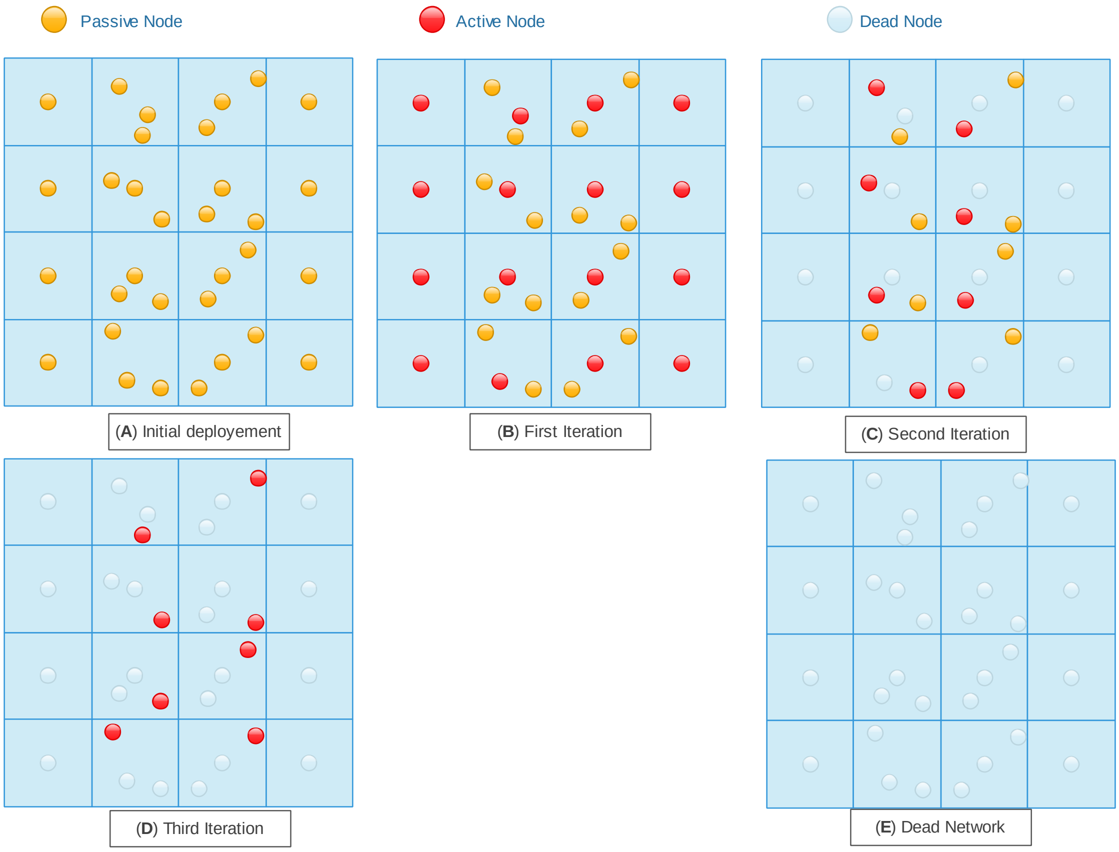

Figure 3. 15: Snapshot of the sensor network without node movement strategy

3.7. Conclusion

In this chapter, we present a novel localized and distributed algorithm “CBNMS” for mobile sensor networks to maximize the network coverage. The monitored area is divided into some grids USs, and these grids are obtained following the Hilbert trajectory with a specific order; in order to ensure the coverage and connectivity requirement.

We use Node scheduling strategy to schedule the activation and deactivation of nodes’ sensor units. For our coverage purpose, only one node per US is required to be active during a round, where the choice of an active node for each square grid based on the centric ones to gravity center. For our coverage purpose, only one node per US is required to be active during a round.

Our approach requires the network to be dense enough so that load balancing can be proceeded in the entire sensory field. We use a square grid-based movement strategy; the movements within a Hilbert sub-squares can freely move to other sub-squares. The node movement method consists of migrate the node from high density to low density grid, only nodes closest to boundary of US decide move to US neighbors.

The next chapter will describe the simulations conducted by our solution by measuring some performances.

Chapter 4

Simulation and performance Analysis

4.1. Introduction

Traditionally, three main techniques for analyzing the performance of wired and wireless networks are the analytic methods, simulations and physical measurements. Performance evaluation through simulation has the advantage that the resulting precision or accuracy of all nodes does not have to be measured but is directly accessible. Thus, much larger systems can be evaluated simulation is developed to illustrate the correctness of our analytical results

In addition, a small number of sensor networks exists that due to various unsolved research problems, making it almost impossible practical measures. It seems that the simulation is the only possible approach for quantitative analysis in sensor networks. A simulation environment that tests protocols with good precision would be very beneficial in terms of time and cost.

We have describe in the previous chapter our distributed protocol called CBNMS (Coverage based Node movement Strategy); that preserves the area coverage in sensor networks, whereas nodes are mobiles and the network topology is changed.

To evaluate the performance of our area coverage protocol, we have simulated theirs functionalities using a simulator developed in JAVA.

In this chapter, we present the simulation results and performance evaluation of our protocol, this by comparing it with different solutions of area coverage problem.

4.2. Simulation

Various network simulation environments exist to test sensor networks algorithms, including OMNET ++, GloMoSim, OPNET, Java-Sim, SensorSim, NS-2 and many others.

To experience the proposed solution, we chose to use our proper simulation tool. We have not chosen to implement our protocols in a popular simulator such as NS-2 or OPNET . We wanted to study aspects of the protocols by limiting the "noise" caused by other parameters whose impact on the results remains difficult to estimate. To analyze the performance of the proposed protocol, we developed a simulator written in Java.

In this section, we evaluate the performance of CBNMS, and then we compare its results with others obtained by other Coverage area of interest algorithms (AI).

4.2.1. Simulator Presentation

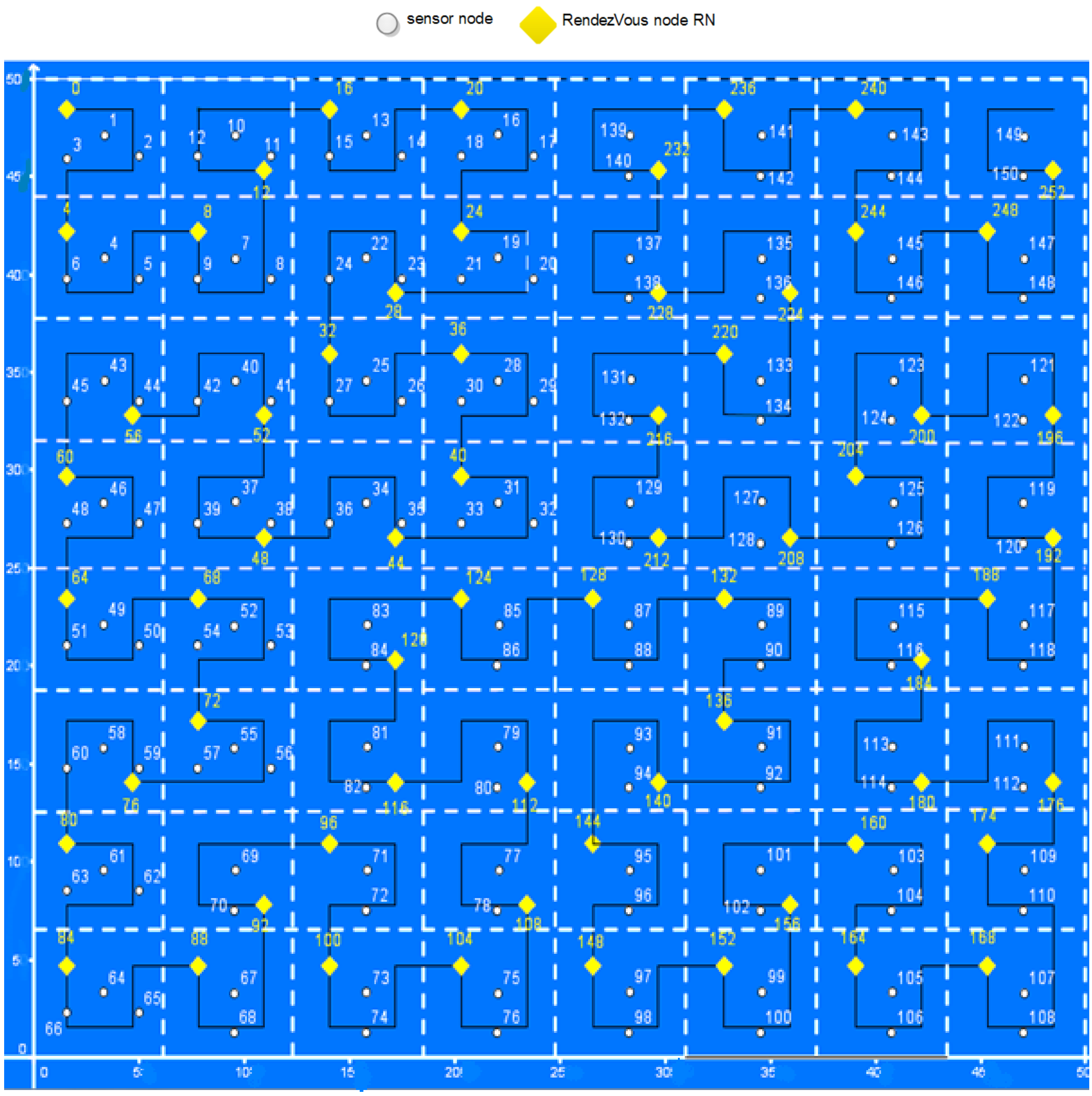

Figure 4. 1:An example of a monitored area with 150 deployed nodes in Hilbert space of order n=4

The programming language should allow both coding algorithms proposed (see Chapter 3), and the support of technologies mentioned. Although several languages suitable for this task, we chose the JAVA language that meets our needs, namely, true object-oriented programming to our modeling, technology support (Network, MAC, physical layer ...) and the possibility to have a mobile application deployed in multiple environments.

Simulations have been developed in Sun Java, a very popular programming environment. In the implementation we have represented each sensor node by a thread "lightweight processes," however; we have scheduled synchronization and communication processes to ensure competition between nodes and rendezvous Nodes.

4.2.2. Simulation' parameters

To evaluate our approach, we conducted multiple series of simulations using the Java programming environment. Our simulations involve scenarios where the number of nodes varies from 100 to 450. The nodes are distributed uniformly over a grid of squares of 2-dimentional plane, in an area of 50x50 square units. The monitored area is divided into grid of squares according to Hilbert trajectory. To increase the confidence of the obtained results, we performed several runs (an average of 100 runs) and took average results.

Energy levels are initially fixed at 100 units. An active node still running until it loses all battery unit, consuming energy by 1 during 1 unit of time. Energy spent for idle nodes is 0.2 units loosed per node. The energy cost of messages is 1 unit for the transmission of one message, and 0.2 units for the reception of one message. Energy levels of sleeping nodes remain unchanged.

For sensor mobility, we use the energy model presented as used in [**54**,**55**], where a constant rate of energy drain will incur during sensor. Let *Emu* equal to be the energy cost for a sensor to move one unit distance, and dist the distance traveled by a sensor. The energy spent by a sensor during its mobility is computed as: *Emove(dist) = Emu × dist.* In our case, mobile nodes can only move within a limited distance where *dist= 2*Scale*, and scale represent the distance between two square unites, Emu equal to energy spent in idle state ,and *Emove* is the energy spent by a sensor during its mobility from a square unit to another.

The first simulations are used for performance parameters to make comparison with Area coverage algorithms in static and mobile network. Table 4.1 summarizes all simulation parameters used for our solution in static and mobile network.

In our simulations, we used a network topology of 500 nodes. We suppose for all nodes the same sensing range of 8m (i.e., when$SR \geq l \times \sqrt{2} = 7.98$, the coverage of grid cell US is guaranteed, and with $SR = 8$ is the minimum value), using the previous calculus, as exposed earlier in chapter 3, it results in a Hilbert Trajectory Order the corresponding Hilbert trajectory is (order 4, scale s= 2.82 m and 64 grid cells), to guarantee a whole coverage of the initial covered area, where at the same time connectivity is maintained. We varied CR from 8 to 30 meters and for each value we found its correspondent coverage rate.

	Parameters	**Values**
Area Coverage Algorithms' in static network	Surface of the area of interest	50m × 50m
	Number Of deployed Nodes	100, 150, 200, 250,300,350,400,450
	Communication range (CR)	8, 12, 16, 20, 30 m
	Sensing range (SR)	8 m
Area Coverage Algorithms' in mobile network	Surface of the area of interest	50m × 50m
	Number Of deployed Nodes	100, 150, 200, 250,300,350,400,450
	Communication range (CR)	8, 12, 16, 20, 30 m
	Sensing range (SR)	8 m
	Mobility degree	10%, 20%, 30%, 40%,50% of deployed nodes
	Moving Distance	2xScale (a fixed traveled distance)

Table 4.1: Simulation Paramaters

Because our solution is grid based approach where the objective is to achieve a load balanced state; each grid US requires one node to be active during a round. Therefore, our simulation started with 64 active nodes when each grid US full by one active node.

Simulations have also been made towards the case when all sensors are active until their energy is exhausted. We carry out the computer simulation to evaluate the performance of Our Algorithm in static and mobile network. The main metrics measured in our simulations are:

- *Network coverage;* the total area covered by deployed nodes.
- *Proportion of active nodes;* the average number of working active set.
- *Average number of messages per node* ;the communication cost of each node.

4.2.3. Coverage Evaluation Method

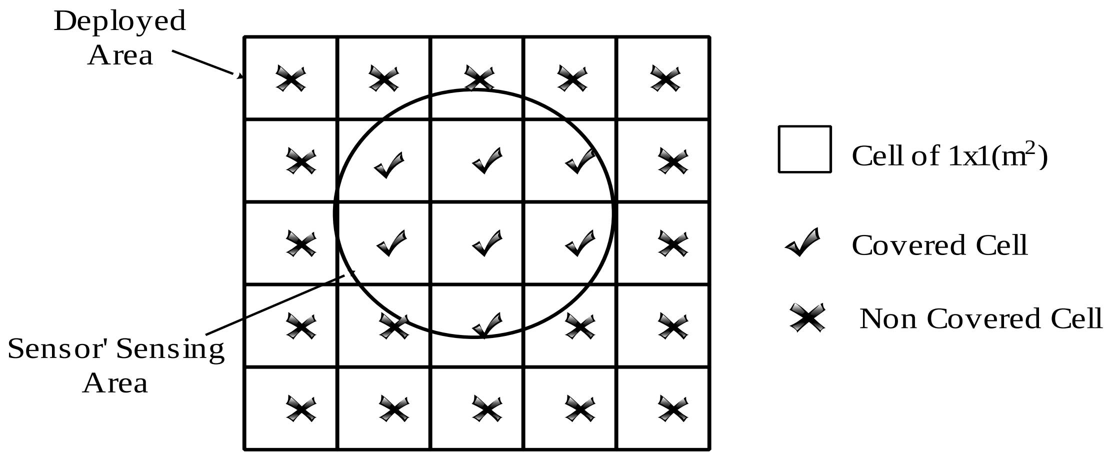

Figure 4. 2: Coverage Evaluation Method of the Monitored Area

To calculate the coverage, we divide the deployed area into 1m×1m unit cells. If a cell center, which is covered by original network, cannot be covered any more after turning off some nodes, we say a sensing hole occurs. The percentage of covered area is then the ratio of the number of cells covered by at least one sensor and the total number of cells (see Figure 4. 2) [**35**].

4.3. Performances evaluation

First, we examine the performance of the proposed protocol (CBNMS) by varying the communication radius and when density is 500 sensor nodes. For this, the sensing radius is 8 m (SR = 8m) and the communication range CR varies between 8 and 30 (m) (CR = 8m To CR = 30m).

Second, we examine the performance (CBNMS) while varying the number of mobile sensor nodes (mobility degree), when the number of deployed nodes is 500 sensor nodes and 800 sensor nodes respectively. The sensing range is fixed to 8 (m) (SR = 8m) and the communication range CR is twice the sensing range, equal to 16 (m).

4.3.1. The coverage ratio provided by CBNMS while varying the communication range CR

Coverage reflects the percentage of active sensors in the network and its value shows the degree to which the network is covered by active nodes. In Figure 4. 3, we illustrate the average area coverage of interest provided by the CBNMS protocol based on the communication radius CR in static and mobile mode.

According to Figure 4. 3, we see that more than the communication range increases, the coverage decreases in mobile and static mode of CBNMS. Because with a large communication radius, more connected set are generated with a small number of active nodes, the probability that a node is in sleep increases. The global coverage versus CR variation is slowly reduced, because our approach is grid based approach, and the choice of Hilbert order is to guarantee a whole coverage and to maintain the network connectivity.

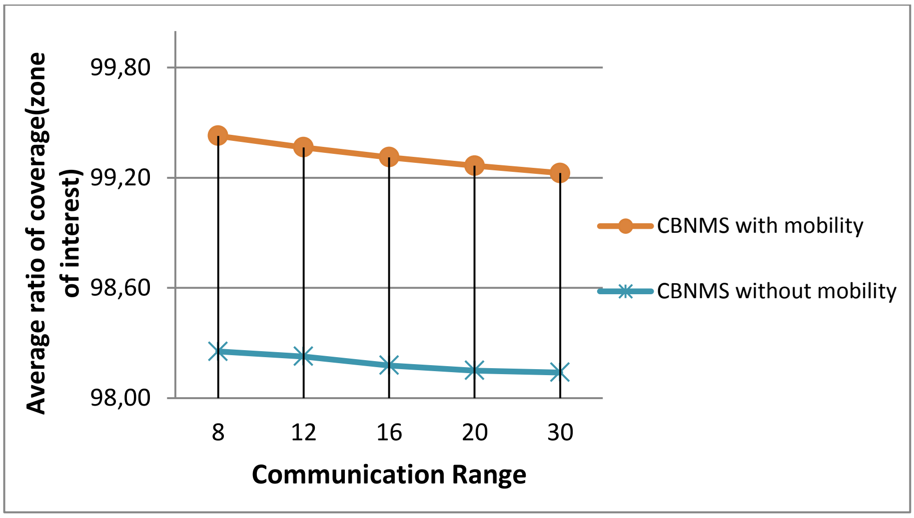

Figure 4. 3: The percentage of coverage while varying the communication range CR

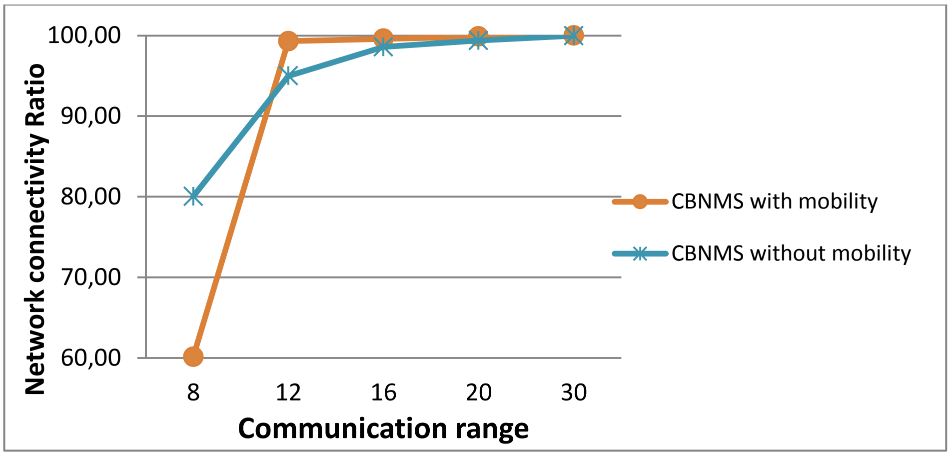

Figure 4. 4: Network connectivity Rate while varying the communication range CR

When attaching the capture at 8m we ensure complete coverage of a grid cell by any sensor within a grid cell, and when we increase the communication range up to 12.61 m (i.e., whenCR $\geq$ l x $\sqrt{5}$, where l is the grid US size see Figure 4. 4) we ensure that any node within a cell can directly communicate with any other node in its four adjacent USs, so more connected set are generated with a small number of active nodes , less area of interest are covered, and many empty regions generate covers holes. We use graph coloration method in order to measure connectivity of each active set.

Due to mobility of nodes, the coverage of monitored area in CBNMS is more extended than in static network with coverage improvement of (1.17%, 1.14%, 1.13%, 1.12%, 1.09% respectively while increase the communication range CR). This is according to links which are created due mobility [**56**], where mobiles and sensors can interact with each other and increase area coverage in one hand. Our movement strategy is loud based strategy where mobile nodes in over-loaded cells should move to under loaded cells. Furthermore, nodes should move in an efficient way to fill the holes and the coverage will be maximized in other hand.

4.3.2. Average ratio of active nodes provided by CBNMS while varying CR

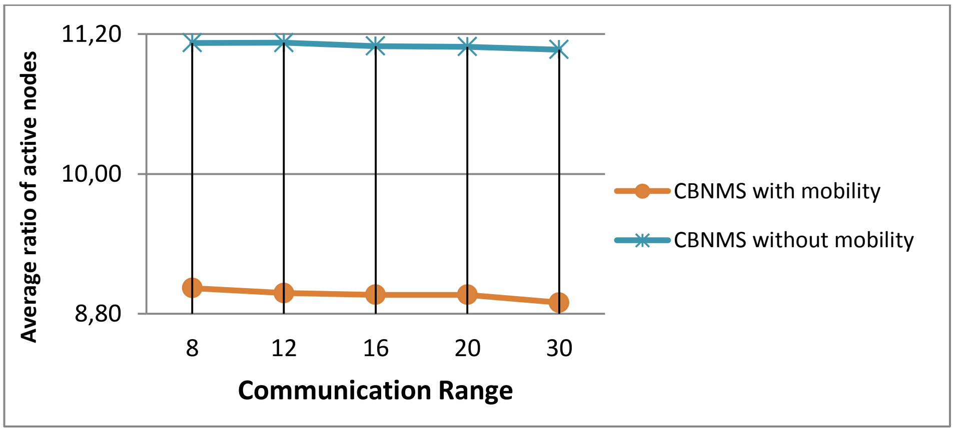

Figure 4. 5:Percentage of Active Nodes while varying the communication range CR

In addition Figure 4. 5 shows the proportion of active nodes of CBNMS with and without mobile nodes. When we increasing the communication range the percentage of active nodes is decreased because more connected set with few number of working sensors are generated, so the number of coverage holes increases and the proportion of active nodes decrease.

CBNMS in presence of mobile nodes generates less active nodes proportion compared to CBNMS without mobile nodes; whereas the communication range values are varied. This difference rate is considerable, in case where (CR=8) the difference rate is 2.10%, this difference rate grow proportionally to the value of communication range until reaching 2.17% where (CR=30). This is due mobility where nodes migrate from less dense regions towards lowest ones ,so network coverage overlaps is reduced and the number of working sensors are decreased.

4.3.4. The coverage ratio provided by CBNMS while varying the degree of mobility

Figure 4. 6 shows that for a fixed degree of mobility, more sensors are deployed, more the coverage rate of the monitored area is important. In addition, we see that more than the mobile nodes increases, the coverage of monitored area in CBNMS are more extended initially. When density is 500 and 800 deployed nodes, the coverage ratio increases gradually from (98.18% and 98.44% respectively) for 0% of mobility degree until (99.45% and 99.53% respectively) for 20% of mobility degree. Then the coverage ratio decreased slightly by (0.13%,0.08% respectively) with addition of 10% of mobile nodes ,this is due to the fact that more empty regions are generated which degraded a little the network coverage performance. With mobility' degree of 40%, CBNMS provides best coverage ratio with (99.47% and 99.63% respectively) where the addition of 10% of mobile nodes improve considerably the coverage ratio, this is due the mobility of nodes make filling to empty regions USs generated.

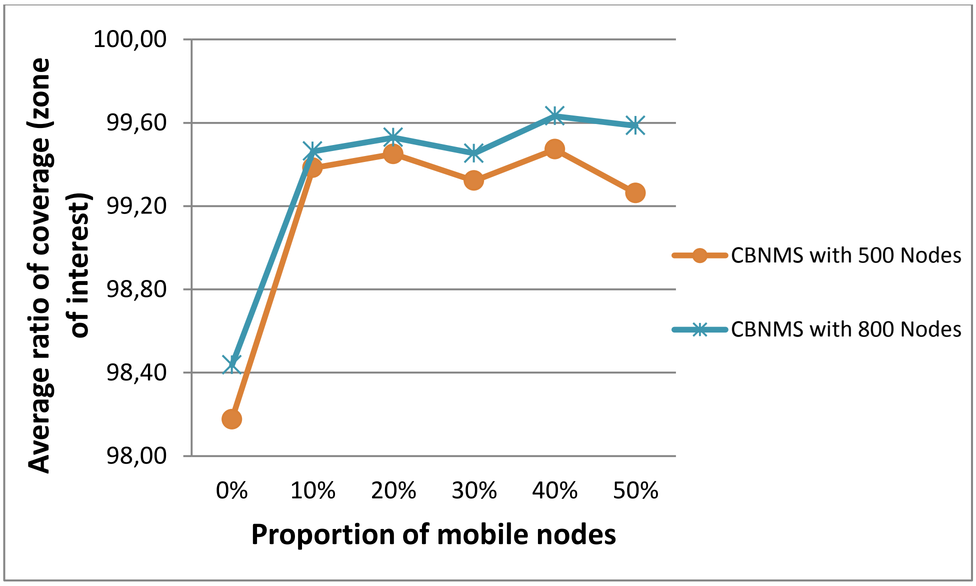

Figure 4. 6:The percentage of coverage while varying the degree of mobility

Finally with mobility degree of 50%, CBNMS coverage ratio is decreased again with (0.21%, 0.05% respectively).

We see that CBNMS in presence of mobile nodes provides an improvement of coverage ratio compared to CBNMS without mobility, whatever the degree of mobility of nodes. While the degree of movement affects slightly an increase or decrease of the coverage ratio of the monitored area.

4.3.4. Average ratio of active nodes provided by CBNMS while varying the degree of mobility

Figure 4. 7 presents the evolution of the percentage of active nodes provides by CBNMS when the number of deployed nodes is 500 and 800 respectively; while varying the mobility' degree in the network for CBNMS. The percentage of active nodes is higher in network without mobility (11.10% and 6.86% respectively); we see that more than the degree of mobility increases; the percentage of active nodes is decreased until reaching(8.86% and 5.22% of active nodes respectively) with 50% of mobility' degree . Because due mobility, network topology is changed and more new active set with few number of working sensors are generated, so proportion of active nodes decrease.

4.3.3. Average number of messages per node

Assuming no collisions, no Neighbor discovery phase and no Node' location information is required. We quantify the amount of sent messages in our protocol. Let N be the number of deployed sensor nodes, in our CBNMS algorithm only one active node is sufficient for survey each square unit US, other nodes are sleeping nodes. Each active node will turned off send one *SendOutOfEnergy*(*Node*$_{ID}$) message to its RN . Sleeping nodes switch to idle state send *ReplyMsg*(*Node*$_{ID}$) to their RN. It means the number of *SendOutOfEnergy*(*Node*$_{ID}$) equal to N the number of deployed sensor nodes, and L be the

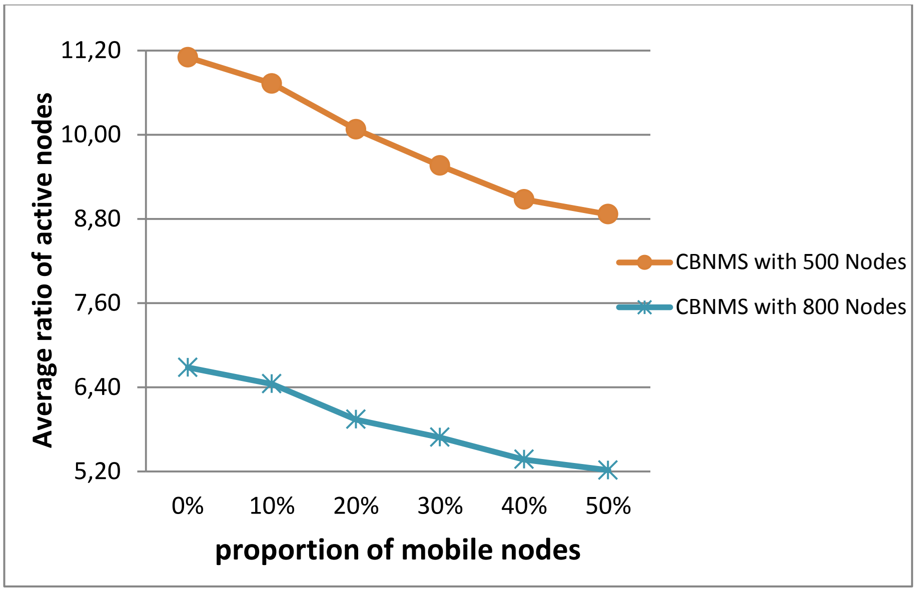

Figure 4. 7: Percentage of Active Nodes while varying the degree of mobility

number of *ReplyMsg*(*Node*$_{ID}$) messages. Thus the total number of messages in our protocol CBNMS is equal to N + L.

In presence of mobile nodes there are two types of messages was sent for each mobile node. *SignalMove*(*Node*$_{ID}$) message was sent to its RN of US when a node want moving to new region *NewUS*, and *StopMoving*(*Node*$_{ID}$) was sent to new RN of its new region *NewUS* upon the termination of moving node. Let MN be the number of mobile nodes, so the number of sent messages required in our algorithm in presence of mobile nodes is equal to N+L+2*MN messages.

No neighbor discovery phase is required in our protocol; in static mode each grid US of the monitored area is k-covered where k is the number of their sensor nodes, it needs exactly one message for active node, while k-1 messages sent from redundant nodes in listen mode. While, two messages per mobile node are added and required in mobile mode

Figure 4.8 shows that static CBNMS increases linearly in when the number of deployed nodes increase; It has a high communication cost resulting from *Reply messages*, because the number of *Reply* messages increases with K number, it requires only 4.35 message per node for a density of 500 nodes, while it requires 5.87 messages per nodes for 800 deployed nodes. Indeed, in presence of mobile nodes the average number of messages is reduced compared to static mode in CBNMS, this average decreases proportionally when the degree of mobility increase. Initially with degree of mobility of 10%, CBNMS generates 4.34 and 5.29 message per nod for density of 500 and 800 deployed nodes respectively. When the degree of mobility achieve 50% the average number of messages is reduced to 3.78 and 4.26 message per node for 500 and 800 deployed nodes respectively. This is due the mobility of nodes which reduce nodes in listen mode where the number of L messages decrease.

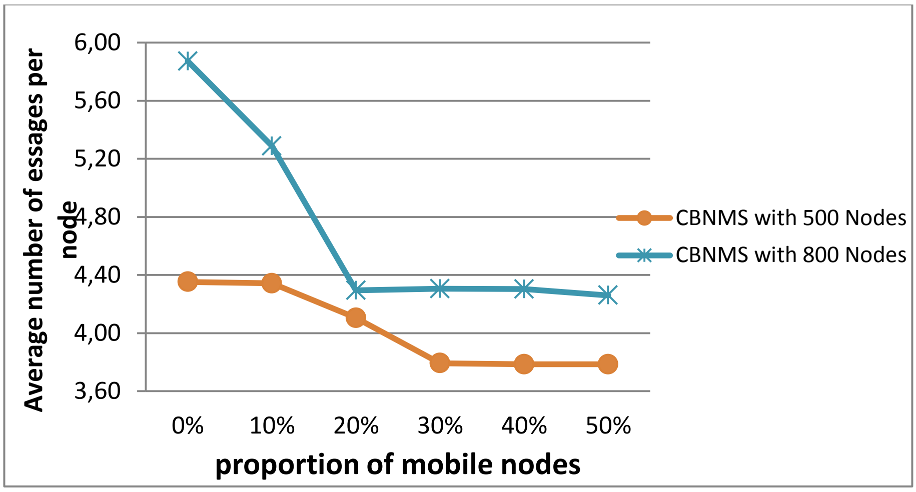

Figure 4.8: Average number of messages per node while varying the degree of mobility

4.4. Comparisons' of performances

In this subsection, we compare the performances of CBNMS with four solutions: the first is the centralized algorithm proposed by Pazand and Datta [**34**]. This algorithm will be noted CDSC (Centralized Dominating Set for Coverage). And the second is distributed algorithm proposed by khanouche [**35**]. This algorithm will be noted DCovPDS (Distibuted Coverage preserving based on Domanting Set). And the third algorithm is another centralized one proposed by Khalil [**36**]. This algorithm will be noted CCSID (Connected Cover Set based on IDentity of node). While the last one proposed by Dabba [**37**] . This algorithme will be noted BCP (Border Coverage Protocol), which preserve both border and area coverage is border coverage.

We will consider here two cases depending the CR/SR ratio. For that we fix the sensing range at 10m, while the communication range is equal to Rs for the first case, and 2×SR for the second (CR>SR).

4.4.1. Percentage of active nodes

In those series of experiments, we varied the deployed nodes density from 100 to 450 nodes. It is important to have as few active nodes as possible.

1) First case CR=SR:

Figure 4.9 shows the average percentage of active nodes while varying densities in case CR=SR. We can observe that, for CR=SR and 100 deployed nodes, CDSC, DCovPDS , BCP, CCSID , CBNMS and CBNMS with mobile nodes have 14.89%, 15.74% ,16.38%, 23.43% ,34.90% and 34.79% of active nodes respectively. Then as the density increase as CBNMS generates less active nodes in its both static and mobile mode, but it is more than other approaches CCSID, BCP, CDSC and DCovPDS (6.09% and 8.52% Vs 5.76%, 4.70% , 4.30% and 4.27% respectively for WSN consist 450 deployed nodes).

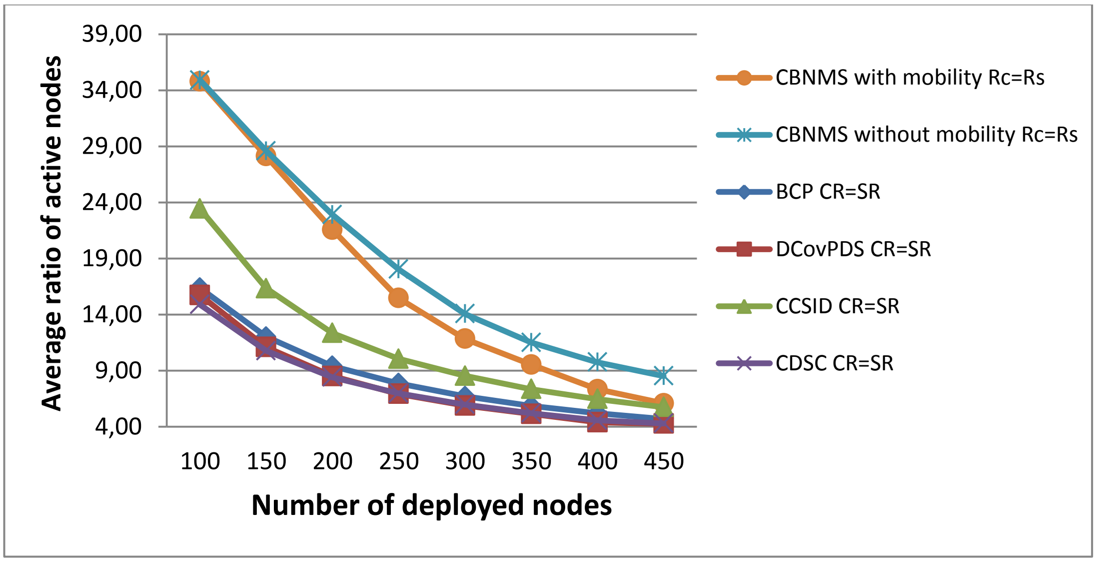

Figure 4.9:Average percentage of active nodes selected by CBNMS, BCP, CCSID, DCovPDS and CDSC in case (CR=SR).

This difference of active nodes proportion is exploited by CBNMS for assuring more connectivity in WSN, each grid US requires one node to be active during a round. Therefore, for each round the average number of nodes to cover the whole target area is 64 nodes. And due mobility, so network coverage overlaps is reduced and the number of working sensors are decreased.

2) Second case Rc>Rs:

We see from Figure 4.10 that for (CR > SR) and the number of deployed nodes equal to 100, CCSID has 13.87% of active nodes, CDSC has 18.81% of active nodes, DCovPDS with 19.44% of active nodes ,followed by BCP with 22.45%, then CBNMS with 37.77%, and 38.44% for CBNMS with mobile nodes. More than the number of deployed nodes increases, CBNMS without mobility generates more active nodes that CCSID, CDSC and DCovPDS (10.53% against 3.55%, 8.16% and 9.70% respectively) for a topology consisting of 450 nodes, and less active nodes that BCP when density reach 350 deployed nodes(13.48% against 14.53%) and(10.53% against 13.52% for density of 450 deployed nodes) . While CBNMS with mobility generates more active nodes that CCSID and CDSC (8.19% against 3.55%, 8.16% respectively), and less active node that DCovPDS when density reach 400 nodes (9.48% against 10.12% for a density of 400 nodes) and (8.19% against 9.70% for a density of 450 nodes), and less active nodes that BCP when deployed nodes is 300 nodes (13.77% against 14.94% active nodes) until reaching(8.19% against 13.52% of active nodes for a density of 450 deployed nodes).

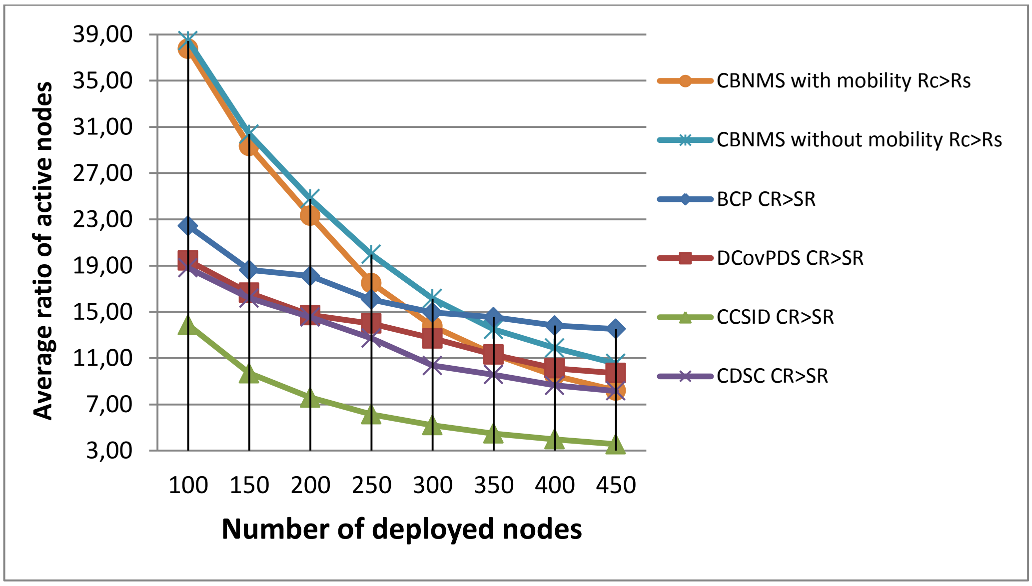

Figure 4.10: Average percentage of active nodes selected by CBNMS, BCP, CCSID, DCovPDS and CDSC in case (CR>SR).

Our protocol requires more active nodes that DCovPDS, CCSID and CDSC because CBNMS protocol is localized and each RN ensure that its region US is covered by one node at once (64 nodes for 64 USs), which generate more active nodes compared to other solutions. The mobility of nodes extends network lifetime and; so more connected set are generated with a small number of active nodes where the active nodes proportion is decreased

Obviously, when the network size is high, this network needs fewer nodes to monitor the considered area. We then say that the percentage of nodes required to ensure coverage is inversely proportional to the network density.

4.4.2. The coverage rate

The coverage rate is defined as the percentage of the area of interest covered by the set of active nodes. This percentage is calculated by using the method described in Section 4.2.3. This rate is the most important metric to evaluate the performance of the algorithms that solve the coverage problem in wireless sensor networks.

1) First case CR=SR:

Figure 4.11 shows the average rate of coverage provided by CBNMS , CCSID, CDSC, BCP , and DCovPDS in the case (CR = SR).

When (CR = SR), CBNMS without mobile nodes protocol provides less coverage ratio compared to other protocols, but according to Figure 4.11 the area coverage maximized and grow proportionally to the number of deployed nodes. Our algorithm can reach 89.10%, 98.89% where (density=100, and density= 450 deployed nodes respectively). This is due to the fact that the average number of nodes in each grid needs one node; so we can surveil the whole target area by at most 64 active nodes. Our approach requires the network to be dense enough to generate more sets of active nodes with 100% of converge rate, in order to maximizing area coverage. With a number of active sets less than 64 nodes empty regions have been occurred; so network coverage performances will be degraded in low density network than in high density network.

While CBNMS protocol in presence of mobile nodes brings a considerable improvement compared to CBNMS without mobile nodes. Due mobility of nodes, the coverage of monitored area in CBNMS is more extended, this is according to links which are created due mobility [**56**] in one hand, and our loud based movement strategy extends network lifetime where more sets of active nodes with 100% are generated on other hand. Initially the coverage ratio increases gradually to the number of deployed nodes, then proportionally to node movement proportion which increase or decrease the coverage ratio of the monitored area from a density of 250 nodes the coverage ratio reduced slightly with 0.03% , and the same reduction for a density of 450 deployed nodes.

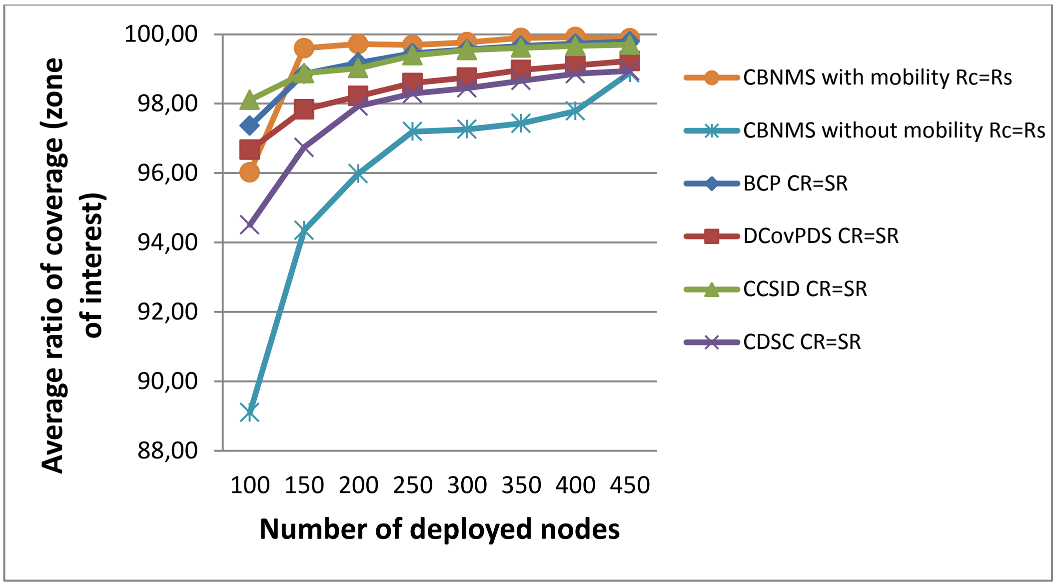

Figure 4.11: Average coverage ratio vs. density achieved by CBNMS , BCP, CCSID, DCovPDS and CDSC in case (CR=SR).

CBNMS provides better coverage ratio than BCP, CDSC, DCovPDS and CCSID protocols, except BCP, DCovPDS and CCSID which provide best coverage ratio initially with density of 100 deployed nodes; this is due to the fact that it uses more active nodes. For 100 deployed nodes, as it is shown in Figure 4.11 that CBNMS provides an improvement of coverage ratio equal to 1.51% compared to CDSC. This improvement continues for a density of 150 deployed nodes (0.72%, 0.74%, 1.77% and 2.86% compared to CCSID, BCP, DCovPDS and CDSC respectively). And this improvement continues for a highest density (0.09%, 0.18%, 0.65% and 0.94% compared to BCP, CCSID, DCovPDS and CDSC respectively for 450 deployed nodes). So there is an improvement by our solution.

1) Second case CR>SR:

Figure 4.12 shows the average rate of coverage provided by CBNMS, CCSID, CDSC BCP, and DCovPDS in the case (CR> SR).

When (CR>SR), CBNMS without mobility is increasing when density increase as in CCSID, CDSC and DCovPDS protocols, with 89.40% for the lowest density and it increases gradually until it exceeds 98.81% for the highest density. CBNMS brings only one improvement of 0.6% compared to CDSC, due to relatively high densities considered of 450 deployed nodes.

In the case of CR>SR, the results of CBNMS protocol in presence of mobile nodes provides better coverage rate than CCSID ,CDSC ,BCP and DCovPDS protocols, except the CCSID of coverage is best initially until that the number of deployed nodes reach 150 nodes (from 150 nodes the CBNMS coverage rate is better than CCSID).

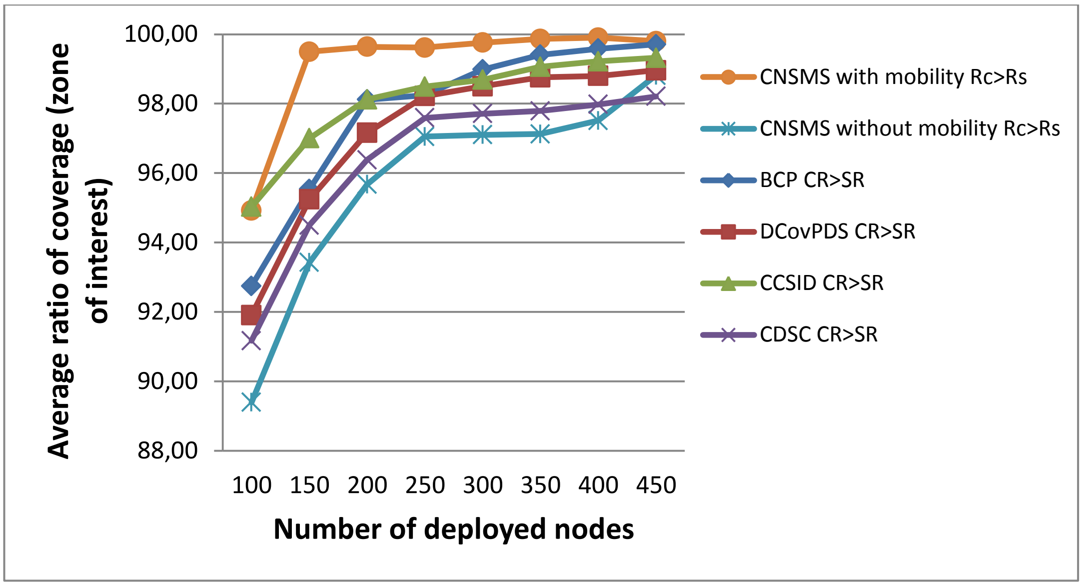

Figure 4.12: Average coverage ratio vs. density achieved by CBNMS , BCP, CCSID, DCovPDS and CDSC in case (CR>SR).

CBNMS provides a better coverage ratio than CDSC and DCovPDS. It is 94.92%, 92.75% , 91.91% and 91.17% for CBNMS, BCP, DCovPDS and CDSC respectively for 100 deployed nodes, so there is an improvement of coverage ratio equal to 2.17%, 3.01% and 3.75% compared to BCP, DCovPDS and CDSC respectively. The coverage ratio for BCP, CCSID, CDSC and DCovPDS grow proportionally to the number of deployed nodes, while the coverage ratio for CBNMS increase proportionally to the number of deployed nodes initially until that the number of deployed nodes reach 200 nodes, after that the coverage ratio of the monitored area increase or decrease according to movement proportion. For 250 deployed nodes the coverage ratio decrease slightly by 0.02%, then the coverage ratio increase when the number of deployed nodes increase until reaching the best coverage ratio of 99.90%, after that the coverage ratio a little reduced by 0.1% for 450 deployed nodes. And this improvement continues for a highest density; the coverage ratio is 99.80%, 99.71%, 99.32%, 98.21% and 98.96% for CBNMS, BCP, CCSID, CDSC and DCovPDS respectively for 450 deployed nodes. So there is a significant improvement by our solution due nodes mobility.

4.5. Conclusion

In this chapter, we design and evaluate a novel distributed algorithm for mobile sensor networks to maximize the network coverage in presence of mobile nodes. Our algorithm uses a square grid-based movement strategy, following Hilbert curve trajectory with a specific order; to ensuring the coverage and connectivity requirement.

We compare the performances of CBNMS with four solutions: the centralized solutions CDSC algorithm and CCSID algorithm; and the distributed algorithms DCovPDS and BCP algorithm. Simulation has been done to validate the effectiveness of the suggested solution. The results show that, CBNMS outperforms the centralized solutions CDSC , CCSID and the distributed solutions DCovPDS, BCP in terms of coverage ratio which is the most important metric.

Simulation results show that CBNMS achieves a high coverage ratio in presence of mobile nodes, which maximize the area coverage in mobile WSN.

Conclusions and future research

Coverage is one of the most fundamental issues in WSNs, which has a great impact on QoS of WSNs. Many algorithms, strategies and mechanisms have been proposed by researchers around the world to solve this problem. We present a brief introduction to the basic knowledge of coverage concepts is given in the beginning.

Then, we describe the coverage issues from two big aspects: sleep scheduling mechanism; which is a very efficient solution of energy conservation to coverage problem in WSN and coverage schemes that exploit mobility using node movement strategy for WSN with mobile nodes. Several studies have focused on developing solutions to ensure coverage in static and mobile sensor networks, and delivering results that are greatly beneficial to the network performances. In different networks types, to further reduce the deployment cost, it is always better for us to deploy the minimum number of sensor nodes within a field so that area coverage can be ensured. Our work revolves within this context and aims to contribute a solution for coverage problem in sensor networks in presence of mobile nodes. We propose a localized solution based on node scheduling scheme for treating coverage problem in WSN. Then we apply sensor mobility strategy in the network to see their impact on coverage and network topology

Next, we propose a novel localized and distributed algorithm “CBNMS: Coverage Based Node Movement Strategy” for mobile sensor networks to maximize the network coverage. The monitored area is divided into some grids USs, and these grids are obtained following the Hilbert trajectory with a specific order; to ensure the coverage and connectivity requirement. We use Node scheduling strategy to schedule the activation and deactivation of nodes’ sensor units, where the choice of an active node for each square grid based on the centric ones to gravity center. For our coverage purpose, it has done a local controlling strategy of monitored area when nodes move with remaining the area coverage, only one node per grid is required to be active during a round. A square grid-based movement strategy is used in order to maximize networks the network coverage and to minimize the movement cost. The movements within a Hilbert sub-squares can freely move to other sub-squares. The node movement method consists of migrate the node from high density to low density grid, only nodes closest to boundary of grid US decide move to grids neighbors. Our approach requires the network to be dense enough so that load balancing can be proceeded in the entire sensory field.

Hence, simulation has been done to validate the effectiveness of the suggested solution. The results show that CBNMS in presence of mobile nodes provides an improvement of coverage ratio compared to CBNMS without mobility, whatever the degree of mobility of nodes. As well CBNMS achieves a high coverage ratio in presence of mobile nodes, which outperforms the centralized solutions CDSC, CCSID and the distributed solutions DCovPDS, BCP in terms of coverage ratio which is the most important metric. Nevertheless, CBNMS has a high communication cost resulting from *Reply messages,* which generates much overhead and extends their coverage overlaps

To this end, we are looking for an approach that finds:

- The best schedule in terms of maximizing the current overall coverage with only few working nodes and without generating much overhead in order to reduce their coverage overlaps. Therefore, due to the limited energy resource in each sensor node, we need to utilize the sensors in an efficient manner so as to conserve sensor energy and increase the lifetime of the network especially in static mode.
- An optimal precision and accuracy of node's location, and activation order assignment for nodes per grid. We intent in future to use a classification method as k-means or knn based on nodes distances to the centers of gravity of each grid cell, the accuracy strongly dependent of closest nodes to gravity center.
- A self-grid relocation technique is needed after movements of nodes inter sub-squares; in order to reach the center of the grid US to maximize the network coverage.
- Our future work will focus the coverage and connectivity problem in case of network with damaged nodes, and with the presence of obstacles. Moreover, we let go the assumption of a dense network of sensors, and will review the results of the proposed protocol, for low-density network
- Finally; we aim to improve our protocol to be self-organized, executed independently without any external intervention including Rendezvous nodes. Sensors will be autonomic enough to be more adapted to dynamic networks where the topology changes frequently.

References

[1] G. Wang, G. Cao, and T. LaPorta, "Movement-assisted sensor deployment.," *IEEE Transactions on Mobile Computing(ToMC)*, pp. pp 640–652, 2006.

[2] I. F. Akyildz, W.Su, Y.Sankarasubramaniam, and E.C, "A Survey on Sensor Networks," *EE Communication magazine*, vol. 40, no. 8, pp. 102–114, Aug 2002.

[3] Wang, B., *Coverage Control*. Singapore: Springer-Verlag London Limited, 2010.

[4] R. Mulligan and H. M. Ammari, "Coverage in wireless sensor networks: A survey. ," *Network Protocols and Algorithms*, no. 2(2), pp. 27-53, 2010.

[5] J.M. Bahi, A. Makhoul, and A. Mostefaoui, "Improving Lifetime and Coverage Through Mobile Beacon for High Density Sensor Networks.," *IEEE International Conference on Sensor Technologies and- Applications*, 2008.

[6] A. C. Viana and M. Dias de Amorim, "Coverage strategy for periodic readings in robotic-assisted monitoring systems.," *Elsevier*, 2012.

[7] M. Kamat, A.S. Ismail, and S. Olariu, "Optimized-Hilbert for Mobility in wireless sensor networks.," *in:Proc. of ,IEEE International Conference on Computational Science and Applications (ICCSA)*, 2007.

[8] L. Gavrilovska, S. Krco, V. Milutinovic, I. Stojmenovic, and R. Trobec, "Application and Multidisciplinary Aspects of Wireless Sensor Networks, chapter6: Mobility Aspects in WSN, Computer Communications and Networks," *Springer London Limited*, 2011.

[9] k. Romer and F. Mattern, "The design space of wireless sensor networks.," *Wireless Communications, IEEE*, vol. 11(6), pp. 54-61, 2004.

[10] M. Cardei, "Coverage problems in sensor networks," *Springer New York. Computer and Electrical Engineering and Computer Science Department, Florida Atlantic University, Boca Raton, FL, USA*, pp. 899-927, 2013.

[11] Y. Derdour, B. Kechar, and F. Khelfi, "The Impact of the Mobile Element on Performance Improvement in Wireless Sensor Network.," *Procedia Computer Science*, vol. 32, pp. 261-268, 2014.

[12] M. Marks, "A survey of multi-objective deployment in wireless sensor networks.," *Journal of Telecommunications and Information Technology*, pp. 36-41, 2010.

[13] L. Mottola and G. P. Picco. , "Programming wireless sensor networks: Fundamental concepts and state of the art," *ACM Computing Surveys (CSUR)*, vol. 43.3, p. 19., 2011.

[14] S. Misra, I. Woungang, and S. C. Misra, "Guide to Wireless Sensor Networks, Computer Communications and Networks, Chapter 24:Wireless Mobile Sensor Networks: Protocols and Mobility Strategies," *Springer London Limited*, 2009.

[15] A. Makhoul, Réseaux de capteurs : localisation, couverture et fusion de données., 2008, THÈSE DOCTEUR en Informatique. Université de Franche- Comté (LIFC).

[16] C. Zhu et al., "A survey on coverage and connectivity issues in wireless sensor networks," *Journal of Network and Computer Applications*, vol. 35 , pp. 619–632, (2012).

[17] H. Zhang and J. Hou, "Maintaining sensing coverage and connectivity in large sen-sor networks," Technical Report UIUC, UIUCDCS-R-2003-2351, 2003.

[18] X. Wang, G. Xing, Y. Zhang, C. Lu, and R. Pless, "Integrated coverage and connectivity configuration in wireless sensor networks.," *ACM. Proceedings of the 1st international Conference on Embedded Networked Sensor Systems*, pp. pages 28-39, 2003.

[19] S. Schmid and R. Wattenhofer, "Modeling sensor networks.," *Algorithms and Protocols for Wireless Sensor Networks*, vol. 62 , no. 77., 2008.

[20] A. Ghosh and S. K. Das, "Coverage and connectivity issues in wireless sensor networks: A survey," *Pervasive and Mobile Computing*, no. 4(3), pp. 303-334., 2008.

[21] L. Liu, X. Zhang, and H. Ma, "Localization-oriented coverage based on Bayesian estimation in

camera sensor networks. ," *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pp. 1–8 , 2008.

[22] M. Li, Z. Li, and A. V. Vasilakos, "A survey on topology control in wireless sensor networks: Taxonomy, comparative study, and open issues.," *Proceedings of the IEEE*, vol. 101(12), pp. 2538-2557, 2013.

[23] A. Gallais, Ordonnancement d'activité dans les réseaux de capteurs : l'exemple de la couverture de surface. Thèse de doctorat en sciences mathématiques (spécialité informatique), Université des sciences et technologies de Lille, France, juin 2007, 2007.

[24] J. Luo and J. P. Hubaux, "Joint mobility and routing for lifetime elongation in wireless sen-sor networks," *INFOCOM 2005, 24th Annual Joint Conference of the IEEE Computer and Communications Societies, Proceedings IEEE*, 2005, Miami, USA.

[25] S. R. Gandham, M. Dawande, R. Prakash, and S. Venkatesan, "Energy efficient schemes for wireless sensor networks with multiple mobile BSs," *IEEE GLOBECOM*, vol. 1, pp. 377-381, 2003.

[26] Z.M. Wang et al., "Exploiting sink mobility for maximizing sensor networks lifetime," *in Proceedings of the 38th Annual Hawaii International Conference on System Sciences (Hicss'05) Washington, DC: IEEE Computer Society*, 2005.

[27] A. Chandrakasan, A. Sabharwal, and B. Aazhang, "Using predictable observer mobility for power efficient design of sensor networks," *The Second International Workshop on Information Processing in Sensor Networks (IPSN)*, 2003.

[28] R. C. S. S. R. Sushant, "Data MULEs: modeling a three-tier architecture for sparse sensor networks," *Proceedings of the IEEE Workshop on Sensor Network Protocols and Applications*, 2003.

[29] W. Wang, V. Srinivasan, and K.-C. Chua, "Using mobile relays to prolong the lifetime of wireless sensor networks," *in MobiCom '05Proceedings of the 11th Annual International Conference on Mobile Computing and Networking. Cologne, Germany: ACM*, 2005.

[30] T. Benerjee, B. Xie, J.H. Jun, and D.P. Agrawal, "LIMOC: Enhancing the LIfetime of a Sensor Network with MObile Clusterheads," *IEEE Vehicular Technology Conference (VTC)*, 2007.

[31] Mo Li, Zhenjiang Li, and Athanasios V. Vasilak, "A Survey on Topology Control in Wireless Sensor Networks: Taxonomy, Comparative Study, and Open Issues.," *IEEE*, 2013.

[32] A. Gallais, J. CARLE, D. SIMPLOTRYL, and L. STOJM, "Localized sensor area Coverage with low communication overhead.," *IEEE Transactions on Mobile Computing (TMC)*, vol. 7, no. 5, pp. p. 661–672., 2008.

[33] B. Pazand and A. Datta., "Minimum Dominating Sets for Solving the Coverage Problem in Wireless Sensor Networks.," *In Proceedings of the International Symposium on Ubiquitous Computing Systems (UCS 2006)*, pp. 454-466, 2006.

[34] Khanouche, M. E., Le traitement du problème de la couverture dans les réseaux de capteurs sans fil, Mémoire magistère en ECOLE DOCTORALE RESEAUX ET SYSTEMES DISTRIBUES, Promotion 2007-2008.

[35] A. Khalil, Coverage and Connectivity Protocol for Wireless Sensor Networks. Mémoire magistère, 2013.

[36] A. Dabba, Couverture de frontières dans les réseaux de capteurs sans fil. Mémoire magistère, Promotion 2013-2014.

[37] H. Zhang and J. C. Hou, "Maintaining Sensing Coverage and Connectivity in Large Sensor Networks," *NSF International Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless and Peer-to-Peer Networks*, Feb.2004.

[38] N. Heo and P. K. Varshney, "Energy-efficient deployment of intelligent mobile sensor networks.," *IEEE Transactions on Systems, Man and Cybernetics, Part A35(1)*, pp. 78–92, 2005.

[39] A. Howard, M. J. Mataric, and G.S. Sukhatme, "Mobile sensor network deployment using potential fields: a distributed, scalable solution to the area coverage problem.," *In: Proceedings of*

the 6th international symposium on distributed autonomous robotics systems(DARS),Fukuoka, Japan, 2002.

[40] S. Poduri and G.S. Sukhatme, "Constrained coverage for mobile sensor networks.," *IEEE International Conference on Robotics and Automation*, pp. 165–172, 2004.

[41] Wang G., Cao G., and La Porta T., "Movement-assisted sensor deployment.," *IEEE Transactions on Mobile Computing(ToMC)*, pp. pp 640–652, 2006.

[42] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization in distributed sensor networks.," *ACM Transactions on Embedded Computing Systems 3(1)*, pp. 61–91, 2004.

[43] S. Chellappan, X. Bai, B. Ma, D. Xuan, and C. Xu, "Mobility limited flip-based sensor networks deployment.," *IEEE Transactions on Parallel and Distributed Systems18(2)*, pp. 199–211, 2007.

[44] S. Chellappan, W. Gu, X. Bai, D. Xuan, and B. Ma, "Deploying wireless sensor networks under limited mobility constraints.," *IEEE Transactions on Mobile Computing6(10)*, pp. 1142–1157, 2007.

[45] J. Wu and S. Yang, "Optimal movement-assisted sensor deployment and its extensions in wireless sensor networks.," *Simulation Modelling Practice and Theory (Elsevier)15(4)*, pp. 383–399, 2007.

[46] S. Yang, M. Li, and J. Wu, "Scan-based movement-assisted sensor deployment methods in wireless sensor networks.," *IEEE Transactions on Parallel and Distributed Systems 18(8)*, pp. 1108–1121, 2007.

[47] V. Coskun, "Relocating sensor nodes to maximize cumulative connected coverage in wireless sensor networks.," *Sensors*, vol. 8, no. ISSN 1424-8220, pp. 2792–2817, 2008.

[48] N. Bartolini, T. Calamoneri, E. G. Fusco, A. Massini, and S. Silvestri, "Snap and spread: a self-deployment algorithm for mobile sensor networks.," *In: Proceedings of the 4th IEEE/ACM international conference on Distributed Computing in Sensor Systems (DCOSS).*, pp. 451-456, 2008.

[49] G. THIBAULT, Indices de formes et de textures : de la 2D vers la 3D., 2009, PHD These , Laboratoire LSIS,AIX -MARSEILLE university.France.

[50] C. Faloutsos and S. Roseman, "Fractals for Secondary Key Retrieval," *Proceedings of ACM SIGMOD Conference on Management of Data*, pp. pp. 247–252, 1989.

[51] A. C. Viana, Locating and routing in large scale self-organizing networks: From distributed hash tables to adaptive addressing structures, 2005.

[52] E. W. Weisstein. (2015, Avril) Cellular Automaton From MathWorld--A Wolfram Web resource,. [Online]. http://mathworld.wolfram.com/CellularAutomaton.html.

[53] Y. C. Wang, C. C. Hu, and Y. C. Tseng, "Efficient placement and dispatch of sensors in a wireless sensor network.," *IEEE Trans. Mobile Comput.*, pp. 7, 2, 262–274., 2008.

[54] Y. C. Wang and Y. C. Tseng, "Distributed deployment schemes for mobile wireless sensor networks toensure multi-level coverage.," *IEEE Trans. Parallel Distrib. Syst.*, pp. 19, 9, 1280–1294., 2008.

[55] G. Kremer and N. Melot, "Mobility and Synchronizations in wireless sensors networks.," 2010.

Résumé

Plusieurs algorithmes ont été proposés dans la littérature pour résoudre le problème de la couverture dans les RCSF, qui sont basées sur des stratégies différentes. Pour un réseau contenant des nœuds mobiles, la stratégie de mouvement de nœud a été utilisé où l'objectif est de tenir compte de nœuds mobiles pour contrôler la couverture du réseau. Les nœuds mobiles changent les caractéristiques de couverture du réseau via ce déplacement vers les emplacements désirées.

Ici, on désigne un nouvel algorithme distribué basé sur la méthode de mouvement sur grille appelé CBNMS (*Coverage Based Node Movement Strategy*) afin d'assurer les exigences de la couverture et de la connectivité. L'algorithme divis la zone cible en m × m grilles de carrés en utilisant les concepts mathématique de l'espace de Hilbert, les capteurs dispersés densément et uniformément en grilles; et un header de grille doit sélectionner par grille comme un super-nœud placé dans chaque grille.

Notre méthode de mobilité permet de déplacer les nœuds d'une grille de haute densité vers celle de faible densité ; Nœuds de capteurs se déplacent à travers les sous-carrés de Hilbert avec un chemin limité. Notre solution proposée est localisée. Aucune découverte du voisinage n'est nécessaire. Aucune connaissance préalable sur les positions des nœuds. La zone cible plane sans obstacles.

Les résultats de simulation montrent que CBNMS atteint un taux de couverture élevé en comparaison à certains protocoles existants.

Mots-clés: *Réseaux de capteurs sans fil, couverture, connectivité, mobilité, approche basée sur les grilles, espace de Hilbert.*

Abstract

Several algorithms were proposed in the literature to solve the coverage problem in WSN, which are based on different strategies. For network containing mobile nodes, node movement strategy has been used where the objective is to leverage mobile nodes to control network coverage. Mobile nodes change network coverage characteristics via moving to the desired locations.

Herein we design a novel distributed algorithm based on square-grid movement method called CBNMS (*Coverage Based Node Movement Strategy*) in order to ensure the coverage and connectivity requirement. The algorithm divides target areas into m × m square grids using the concept of space-filling curve with scattered sensors densely and uniformly into grids; and selects a grid header in each grid as a super-node placed in each grid.

Our movement method consists of migrating nodes from high density to low density grid; Sensor Nodes move within their Hilbert sub-squares with limited distance. Our proposed solution is localized. No neighbor discovery is needed. No previous knowledge about nodes positions. The plane target area with no obstacles.

Simulation results show that CBNMS achieves a high coverage ratio in comparison to some existing protocols.

Keywords: *Wireless Sensor Networks, Coverage, Connectivity, Mobility, grid-based approach, Hilbert space-filling.*

ملخص

توجد عدة خوارزميات مقترحة من أجل حل مشكلة التغطية في شبكات الاستشعار اللاسلكية ، والتي تقوم على استراتيجيات مختلفة. من أجل شبكة تحتوي على عقد متنقلة، تستخدم استراتيجية تحريك العقد حيث تهدف لاستغلال العقد المتنقلة لضبط تغطية الشبكة. العقد المتنقلة تغير من خصائص تغطية الشبكة وذلك عبر تنقلها إلى الأماكن المطلوبة.

في هذه المذكرة، قمنا بتصميم خوارزمية ترتكز على طريقة الحركة في شبكة مربعات ، تسمى CBNMS (*Coverage Based Node Movement Strategy*) من أجل ضمان متطلبات الربط و التغطية . الخوارزمية تقسم المناطق المستهدفة إلى م × م شبكات مربعة باستخدام مفهوم فضاء هيلبرت مع نشر أجهزة الاستشعار بكثافة وبشكل منظم ما بين الشبكات المربعة. مع اختيار مسؤول في كل شبكة مربعة بمثابة عقدة متخصصة.

طريقة حركة العقد ما بين مربعات شبكة هيلبرت تكون من مربع الشبكة الاكثر كثافة إلى الاقل كثافة و بمسافة محدودة. الحل المقترح محلي. ليس هناك حاجة إلى اكتشاف الجار. لا معرفة سابقة عن مكان العقد. المنطقة المستهدفة من دون عقبات.

أظهرت النتائج التجريبية لـ CBNMS نسبة تغطية عالية بالمقارنة مع بعض البروتوكولات الموجودة.

كلمات البحث: *شبكات الاستشعار اللاسلكية، التغطية ، الربط، التنقل، طريقة مربع الشبكات، فضاء هلبرت*