ACTIVE SET PARTITIONING SCHEME FOR EXTENDING THE LIFETIME OF LARGE WIRELESS SENSOR NETWORKS

A THESIS
SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING
AND THE INSTITUTE OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By
Mustafa Kalkan
January, 2010

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Cevdet Aykanat(Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. İbrahim Körpeoğlu

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Sinan Gezici

Approved for the Institute of Engineering and Science:

Prof. Dr. Mehmet B. Baray
Director of the Institute

ABSTRACT

ACTIVE SET PARTITIONING SCHEME FOR EXTENDING THE LIFETIME OF LARGE WIRELESS SENSOR NETWORKS

Mustafa Kalkan
M.S. in Computer Engineering
Supervisors: Prof. Dr. Cevdet Aykanat
January, 2010

Wireless Sensor Networks consist of spatially distributed and energy-constrained autonomous devices called sensors to cooperatively monitor physical or environmental conditions such as temperature, sound, vibration, pressure or pollutants at different locations. Because these sensor nodes have limited energy supply, energy efficiency is a critical design issue in wireless sensor networks. Having all the nodes simultaneously work in the active mode, results in an excessive energy consumption and packet collisions because of high node density in the network. In order to minimize energy consumption and extend network life-time, this thesis presents a centralized graph partitioning approach to organize the sensor nodes into a number of active sensor node sets such that each active set maintains the desired level of sensing coverage and forms a connected network to perform sensing and communication tasks successfully. We evaluate our proposed scheme via simulations under different network topologies and parameters in terms of network lifetime and run-time efficiency and observe approximately 50% improvement in the number of obtained active node sets when compared with different active node set selection mechanisms.

Keywords: Wireless Sensor Networks, Graph Partitioning, Density Control, Energy Conservation, Active Set Partitioning.

ÖZET

GENİŞ KABLOSUZ SENSÖR AĞLARDA AĞ ÖMRÜNÜ GELİŞTİRMEK İÇİN AKTİF SET BÖLÜMLEMESİ

Mustafa Kalkan
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Prof. Dr. Cevdet Aykanat
Ocak, 2010

Kablosuz Sensör Ağlar, mekansal olarak dağıtılan, enerji kısıtlamaları olan ve bünyesindeki sensörleri kullanarak işbirliği içinde farklı konumlardaki sıcaklık, ses, titreşim veya çevre kirliliği gibi fiziksel ve çevresel koşulları gözlemleyen otonom cihazlardan oluşmaktadır. Bu sensör düğümlerinin kısıtlı enerji kaynaklarına sahip olması nedeniyle, sensör ağlarında enerji verimliliği hassas bir tasarım meselesidir. Bütün düğümlerin eşzamanlı olarak aktif modunda çalışması, ağdaki yüksek yoğunluk dolayısıyla, aşırı enerji tüketimi ve paket çarpışmaları ile sonuçlanmaktadır. Enerji tüketimini azaltmak ve ağ ömrünü uzatmak için, bu tez, sensör düğümleri aktif sensör düğümü setleri şeklinde düzenlemek için merkezi bir çizge bölümleme yaklaşımı sunmaktadır. Şöyle ki, algılama ve haberleşme görevlerini başarılı olarak gerçekleştirmek için, her bir aktif set, istenilen seviyede algılama kapsaması sağlamakta ve bağlı bir ağ oluşturmaktadır. Önerdiğimiz yöntemi, ağ ömrü ve çalışma zamanı açısından farklı ağ topolojileri ve parametreleri altında simülasyonlar aracılığıyla değerlendirdik ve farklı aktif düğüm setleri seçme mekanizmalarıyla karşılaştırıldığında elde edilen aktif düğüm setleri sayısında yaklaşık olarak 50% iyileşme gözlemledik.

Anahtar sözcükler: Kablosuz Sensör Ağları, Çizge Bölümleme, Yoğunluk Kontrolü, Enerji Korunması, Aktif Set Bölümleme.

Acknowledgement

I would like to express my sincere gratitude and thanks to my advisor Prof. Dr. Cevdet Aykanat for the invaluable guidance, support and motivation throughout this research study.

I would like to thank Asst. Prof. Dr. İbrahim Körpeoğlu and Asst. Prof. Dr. Sinan Gezici for reading and evaluating my thesis.

I am grateful to my company, Capital Markets Board of Turkey for the continuous support during this thesis study.

I am also thankful to my family and all of my friends that have any kind of help, motivation or moral support during this study. Especially, I would like to thank my brother Faruk Bilgi for his invaluable friendship.

Finally, I would like to express my deepest gratefulness to Firdevs, my wife, for her persistent love, patience and support in all phases of my life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Rapid progress in wireless networking, production of sensors using micro-electromechanical system technology (MEMS) and embedded microprocessors has made wireless sensor networks possible. These sensor networks are dense wireless networks of spatially distributed, small-sized sensor nodes that collect data from an environment, process and send these data to a sink node directly or via multihop communication using other sensor nodes as relay nodes [22].

Primarily, sensor networks have two different kinds of nodes, namely the sensor nodes that are densely deployed in the target region and the single or multiple sink nodes (base stations) that are located either inside the region or very close to it. Sensor nodes collect and disseminate environmental data about the region and the sink node is the place where the data from sensor nodes are collected for analysis and taking the appropriate actions.

There are varying types of sensors which include seismic, low sampling rate, magnetic, thermal, visual, infrared, acoustic and radar that can monitor different environmental conditions such as: [1]

- Temperature,
- Humidity,

- Vehicular movement,
- Lightning conditions,
- Pressure,
- Soil makeup,
- Noise levels,
- The presence or absence of certain kinds of objects,
- Mechanical stress levels on attached objects,
- Current characteristics such as speed, direction, and size of an object.

Depending on the application requirements, sensors can cooperatively monitor several of the introduced physical or environmental conditions at different locations.

Some of the commercial and military applications of sensor networks include:

- Environmental monitoring: (e.g., traffic, habitat, security)
- Industrial sensing and diagnostics (e.g., appliances, factory, supply chains)
- Infrastructure protection (e.g., power grids, water distribution)
- Battlefield awareness (e.g., multitarget tracking)
- Context-aware computing (e.g., intelligent home, responsive environment) [22, 1]

A sensor node is composed of four main components as shown in Figure 1.1 [1]. These components are sensing, communication, processing and power units. The sensing unit in a sensor node is composed of one or more sensors to observe the environmental conditions. The processing unit enables processing and collabarative operations within a sensor node. The communication unit connects a

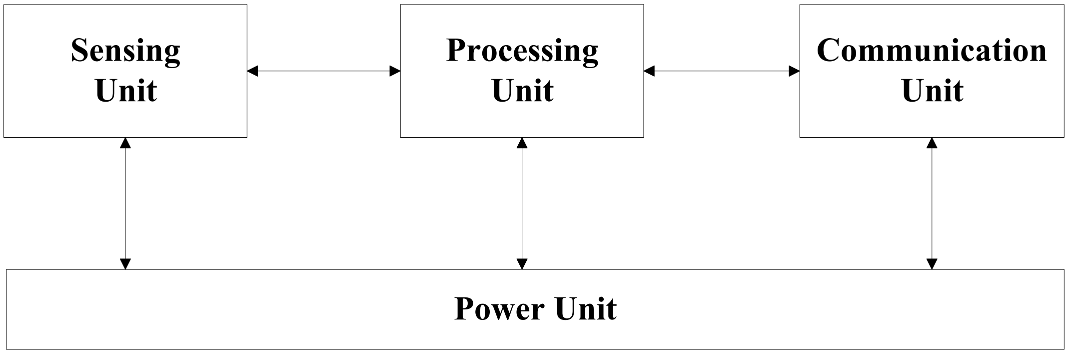


Figure 1.1: Main Parts of a Sensor Node

sensor node to other sensor nodes so that all sensor nodes can form a network as a result. Finally, the power unit supplies energy to the sensor node. From these four main components, the power unit may be the most crucial one in a sensor node considering that all sensing, communication and processing units need energy supplied by the power unit to perform their tasks.

Sensor nodes mostly use batteries in their power units as energy supply and these batteries are most of the time not reachargable and replacable. Due to this energy constrained nature of sensor nodes, energy consumption is one of the fundamental issues in wireless sensor networks. The main energy consumption of sensor nodes is induced by sensing, processing and transmission of data as mentioned. The energy consumption due to these activities in sensor nodes should be minimized in order to increase the network lifetime of a wireless sensor network.

A common approach for minimizing the energy consumption in a wireless sensor network is to leave only some of the sensor nodes in active mode to perform sensing and communication operations and put the remaining nodes into sleep mode. The sensor nodes in the sleep mode are turned off and do not consume energy for sensing and communication operations. At a later time, the sleeping nodes wake up for the sensing and communication tasks when their timer expire.

By scheduling the on-duty times of sensor nodes, energy consumption by sensing and communication tasks in the network can be reduced to some degree. In this way, the network lifetime can be extended. However, sensor nodes are mostly deployed in a random manner. In order to monitor the environment and gather data from the environment efficiently, while putting some of the sensor nodes into sleep mode and keeping only a subset of nodes active, the following two main requirements also need to be taken into consideration:

- **Coverage:** Sensing coverage of a sensor is usually accepted as a circular region around the sensor that it can collect data. While turning off some of the nodes that have the same sensing region, maintaining full or sufficient sensing coverage of the whole monitored area is aimed.
- **Connectivity:** Sensors can send data to the sink directly or via multihop communication. If sensing data goes through a multihop path to sink node, it is important to maintain connectivity among the sensors in order to successfully collect the data generated by sensors at the sink node.

Turning off some sensors and keeping a necessary set of nodes active at a given time is also called density control. There are various studies about density control. Most of the existing density control algorithms are distributed and localized due to the nature of sensor networks. Although distributed mechanisms have some advantages like simplicity and self-organization, centralized density control mechanisms can also be considered, because centralized mechanisms can easily ensure coverage and connectivity objectives at certain desired levels, which is not always possible with distributed algorithms. In addition, optimum solutions can be obtained with centralized algorithms which can be used as a baseline for the distributed algorithms and can be used to study performance limits.

In this paper, we present a centralized density control approach and algorithm for increasing the lifetime of wireless sensor networks, which can be used for wide range of applications. In various wireless sensor network applications, if the network is dense enough, having all the nodes simultaneously working in active mode results in an unnecessary and excessive energy consumption and packet collisions because of high node density and redundant coverage in the network. Consequently, sensors do not survive very long. Our approach helps to avoid this energy waste by leaving only a necessary set of nodes as active at a give time and in this way prolongs the sensor network lifetime.

We propose a graph partitioning based approach that divides a given set of sensor nodes of a WSN into disjoint active node sets (parts) where only one part (one active set) will be active at a given time and parts will alternate to be active and cover a given region. A wireless sensor network to be partitioned will be represented as a graph $G = (V, E)$ where vertices of the graph is the sensor nodes in the WSN and the edges represents the distance between the sensor nodes. The main motivation of our approach is that nodes closer and having overlapped sensing regions do not have to be active at the same time during the operation of network. Then, the algorithm tries to partition the vertices of the graph into disjoint parts such that the number of edges (or the sum of weights) connecting vertices in different parts is minimized. In other words, the method can be regarded as applying declustering on the graph considering that the further nodes in the network are much more likely to be in the same active node set. Therefore, the proposed scheme puts closer nodes in the network to different active sets. This reduces the edge cut between parts after the graph partitioning and also the number of sensors having common sensing regions in the same active set.

The proposed solution essentially tries to organize the sensor nodes into a number of active sensor node sets such that each set maintains the desired level of sensing coverage and forms a connected network to perform sensing and communication tasks successfully. After grouping the sensor nodes into active sets that are capable of sensing and communication tasks individually, we can leave only one of these sets in active mode and rotate this role among all the sets periodically. By this way, operational period of the network can be extended.

The designed algorithm, while minimizing the energy consumption, also maintains the predetermined level of sensing coverage and ensures network connectivity throughout the network. Finally, most existing algorithms attempt to maintain complete coverage, while in reality it might be sufficient to cover a certain percentage of the region. In conjuction, our approach has the capability of finding active node sets that have prescribed level of sensing coverage rather than satisfying complete coverage. Hence, in our scheme, applications can adjust their acceptable sensing coverage level according their needs. If it is enough and acceptable for an application to have a sensing coverage level below complete coverage, this results in an increase in the number of active node sets. Consequently, energy consumption can be further reduced and network lifetime can be extended in applications that have lower predetermined percentage of sensing coverages. And, our scheme provides this flexibility to determine the level of sensing coverage to the applications to prolong the network lifetime.

The rest of the paper is organized as follows. Chapter 2, starts with some background information about sensors and wireless sensor networks. Furthermore, density control in wireless sensor network is discussed and different density control mechanisms, assumptions and objectives are summarized. In Chapter 2, also, an overview of proposed density control mechanisms is presented. In Chapter 3, we will give preliminaries and a formal definition of the problem and will mention about graph partitioning problem. Our centralized graph partitioning approach to the density control in wireless sensor networks is described in Chapter 4. In Chapter 5, we present the simulation environment and the results of our simulations. Finally, conclusion of the work is provided in Chapter 6.

Chapter 2

Background and Related Work

In this chapter, firstly basic information about wireless sensor networks is given. Sensor networks have opened new vistas for many potential applications [18]. We will continue with classifications of these wireless sensor applications in details. Commonly, in the context of different applications, extending the sensor network lifetime is important due to energy constrained nature of sensor devices. Because of this reason, it is accepted that a wireless sensor network is deployed with high density (20 $nodes/m^3$) [18]. In such a high density environment, density control mechanisms, which ensure only a subset of sensors to be active at any time in the network, become important. In density control mechanisms, except the common objective which is maximizing the network lifetime, different sensor applications may have different objectives. For instance, a surveillance application may need the sensor network to have a certain degree of sensing coverage. Other common objectives are network connectivity, high data delivery ratio, high quality of surveillance, scalability, robustness and simplicity. In this chapter, we will also discuss these design objectives together with different design assumptions such as detection model, sensing area, transmission range, location information and distance information considered in different applications [16]. Finally, we will discuss several important properties of density control in an analytical framework and present some of the related centralized and distributed algorithms in the literature.

2.1 Wireless Sensor Networks

In September 1999, Business Week showed wireless sensor network technology among the 21 most important technologies for the 21st century. Essentially, modern research on this sensor network technology dates back to Distributed Sensor Network Program that was initiated by Defence Advanced Research Project Agency (DARPA) around 1980. After all, sensor nodes have evolved by getting cheaper and smaller and current sensor networks may perform functions that could not be dreamed at that time. In Figure 2.1 [5], we can see the evolution of sensor nodes in time. There are 3 generations of sensor nodes in the figure. The first generation of sensor nodes is the TRSS nodes from 1980's and 1990's, which weighted a few kilograms and were as large as a shoe-box or even larger. The next three nodes in the figure are manufactured by Crossbow, Ember and Sensoria companies between 2000 and 2003. These nodes are smaller and lighter than TRSS nodes. Finally, the nodes from Dust, Inc., which have a size of dust particle is possible with the MEMS technology. This rapid progress illustrates us the future of wireless sensor network technology will be overwhelming and change our lives drastically [5].

Figure 2.1: Three Generations of Sensor Nodes

Typically, a sensor network consists of large number of low-cost, low-power sensor nodes that we have discussed previously. The positions of sensor nodes are not essentially need to be predetermined and the sensor nodes are usually spread in an arbitrary manner in a field, as shown in Figure 2.2 [2]. Each of the sensor nodes in the field has sensing, processing and communication capabilities. By the use of these functionalities, a collaborative action of data gathering is achieved by constructing a sensor network. As can be seen in Figure 2.2, the target of the

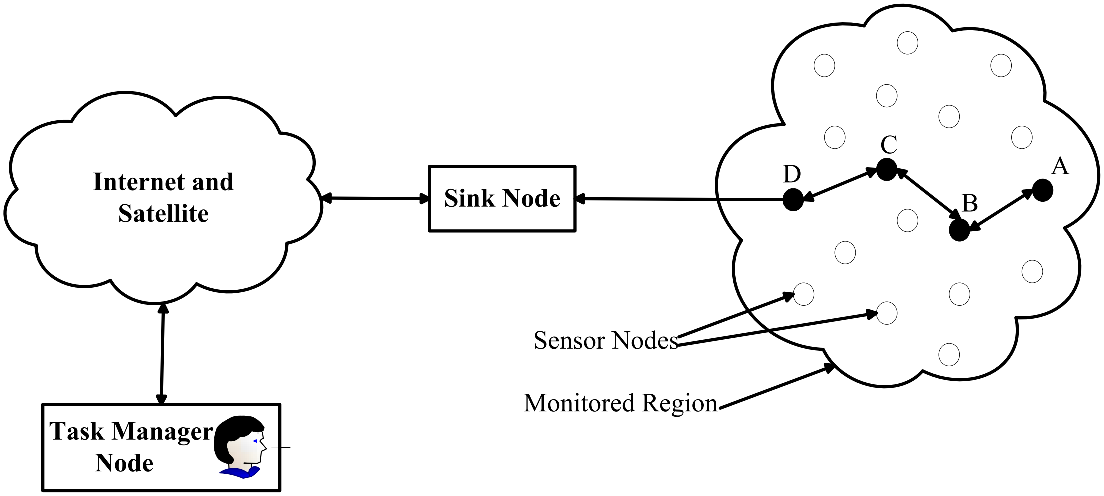


Figure 2.2: Sensor Nodes in a Field

collected data is a sink node that may be located at or near the sensor network. The main responsibility of all the sensor nodes is to send the data to the sink node. After data arrives to the sink node, the sink node takes the necessary actions such as transmitting the data through Internet or a satellite network to a task manager node. The main difference between the sink node and the ordinary sensor nodes is that the sink node is usually assumed to have unbounded energy which is not the case for the sensor nodes. Sensor nodes are usually equipped with batteries which are not rechargeable and usually not replaceable. For this reason, sensor nodes must use their energy supply cautiously.

2.2 Wireless Sensor Network Applications

As mentioned earlier, research on sensor networks has been originally motivated by military applications. There are applications from large-scale acoustic surveillance systems for ocean surveillance to small networks of sensors for ground target detection [5]. However, recent technological advances make many other potential applications possible such as infrastructure security, habitat monitoring, traffic control, etc., that can be broadly categorized into military, environmental, health, home and other commercial areas as follows: [1]

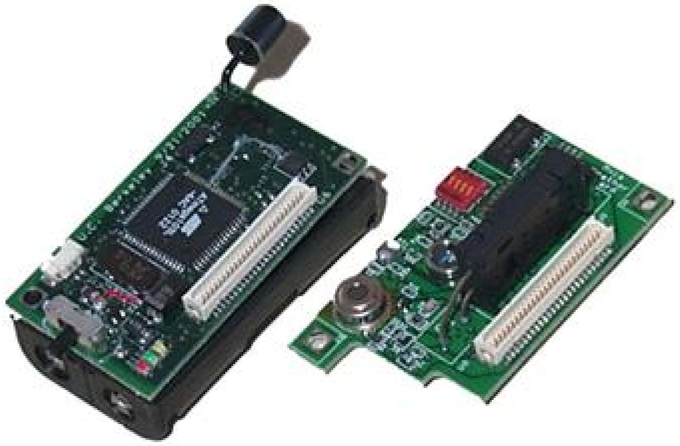


Figure 2.3: Mica Hardware Platform

- **Military Applications:** Sensor networks can be used in battlefields, since sensor nodes are cheap and can easily be scattered around a field in large quantities. Some of the military applications may include monitoring friendly forces, equipment and ammunition in which commanders can see the latest status of the desired equipment, vehicle, etc.; battlefield surveillance in which critical paths and routes can be closely observed for the opposing forces; battle damage assessment just before or after attacks and nuclear, biological or chemical attack detection and reconnaissance in which a sensor network can be used as a chemical or biological warning system.

- **Environmental Applications:** Sensor networks can also be a good approach for environmental monitoring. In Figure 2.3, we see Mica sensor node (on the left) and Mica Weather Board (on the right) developed for environmental monitoring applications. A sensor node with acrylic enclosure deployed in the field can be seen in Figure 2.4. Some of these environmental applications are: forest fire detection in which sensor nodes densely deployed in a forest can relay the exact origin of the fire before it is spread, flood detection in which several types of sensors such as rainfall, water level, weather sensors are used in an alert system to detect floods, and precision agriculture in which level of soil erosion or level of air pollution is monitored in realtime [12].

Figure 2.4: Deployed sensor node with acrylic enclosure

- **Health Applications:** Sensor networks can be used in telemonitoring of human physiological data in which a greater freedom of movement is given to patients than treatment centers, tracking and monitoring doctors and patients in hospitals and drug administration in hospitals in which sensor nodes can be attached to medications and the chance of getting the wrong medication can be minimized.

- **Home Applications:** Home applications include home automation where smart sensor nodes can be put into home devices, such as vacuum cleaners, refrigerators, ovens, to have them interact with each other and with an external network so that they can be managed and controlled locally or remotely; and smart environment where sensor nodes can be put into furnitures and appliances which can communicate with each other and devices in other rooms to learn about the services offered.

- **Other Commercial Applications:** Some of the commercial applications are environmental control in office buildings in which distributed wireless sensor network systems can be installed to control the air flow, interactive museums in which children can interact with objects in museums to learn more about them, detecting and monitoring car thefts, managing inventory control in which each item may have a sensor node that reveals the exact location of the item, and vehicle tracking and detection system [1].

2.3 Density Control in Wireless Sensor Networks

Recent technological advances have made the sensor networks to be used by a wide range of applications as we have mentioned. Some of the applications like environmental monitoring require a high density of sensor devices and these devices have limited battery life. Since the number of sensors is large and a sensor network is deployed in a random manner to remote, hostile environment, it is usually infeasible to recharge or replace the batteries on tens of thousands of these devices. Due to these reasons, a sensor network is designed to run as much as possible. Therefore, it is a necessity not to waste the energy resources in a sensor network. Sensor nodes should have minimal energy consumption. In a high-density network, the same area may be covered by many nodes unnecessarily, causing excessive redundancy and there may be excessive packet collisions, all of which cause energy waste. It is therefore not necessary nor sensible to keep the entire sensor nodes active all the time. Henceforth, to minimize energy consumption and maximize network lifetime, one common strategy that is proposed is keeping only a required number of sensor nodes active at a given time. This is referred as node scheduling, sleep scheduling, or density control.

Density control is the mechanism which puts some of the sensor node into active mode for the sensing and communication tasks and the remaining nodes into sleep mode to save energy. There are various mechanisms proposed in the literature, but different mechanisms may have different assumptions considering different kinds of applications. Additionally, besides a common objective which is extending the network lifetime, different mechanisms may have different objectives depending on the type of application they are designed for. We will look at these design assumptions and design objectives issues in the following sections [16].

2.3.1 Design Assumptions

For all density control mechanisms, the common assumptions are that the sensor nodes have limited amount of energy and that it is very important to have a long durational network. Besides these common assumptions, there are various other assumptions that are made by different algorithms and protocols. They are listed in Figure 2.5. Below, we will discuss each of these briefly.

- **Network Structure:** A sensor network can be flat in the sense that every sensor node has the same role. Also, a sensor network can be hierarchical. For instance, in applications for detection and tracking, some nodes can be used as fusion centers that collect data from sensors, make decisions and send reports to a sink node. These hierarchical networks are generally cluster-based sensor networks.
- **Sensor Deployment Strategy:** The performance of a sensor network is closely related to the initial placement of sensor nodes. There are various deployment strategies including predetermined or random deployment. In predetermined deployment, node locations are planned and predetermined and deployment is done accordingly. In random deployment, nodes are randomly placed to a field. For example, the sensor nodes can be scattered from an airplane to a sensing field. Usually, deployment that is done results in a high-density network initially. This enables some of the sensor nodes to be put into sleep mode until they are needed.
- **Detection Model:** A sensor node can be regarded as detecting an object when the object is inside the sensing range of the node. In addition to this deterministic approach, probabilistic models are also possible.
- **Sensing Area:** Sensing area is usually assumed to be a deterministic circular area or a 3D sphere. Also, sensors are mostly assumed to have the same sensing range.
- **Transmission Range:** Several mechanisms assume that sensor nodes can

change their transmission power to adjust their transmission range. However, actual transmission range may also be affected under a certain fixed transmission power.

- **Time Synchronization:** In most of the mechanisms, sensor nodes are assumed as time synchronized so that they can wake up at the same time to start a new round of scheduling.
- **Failure Model:** Sensor nodes will fail when they run out of energy. . Alternatively, sensor nodes may fail unexpectedly, before the energy is depleted, e.g., sensors in a battlefield can be destroyed by vehicles.
- **Sensor Mobility:** Sensors are presumed to be stationary most of the time. Most papers argue that most real-world sensor networks involve little or no mobility [16].
- **Location Information:** The location information can be predetermined and hardcoded into sensor nodes or sensor nodes may be equipped with GPS or they may run localization algorithms to determine their locations.
- **Distance Information:** Sensor nodes are assumed to be able to determine their distance to their neighbours. This can be, for example, inferred from the strength of the received signals or calculated from the location information.

2.3.2 Design Objectives

Different sensor network applications have different requirements; hence sensor networks usually have different design objectives and priorities among these objectives. Minimizing energy consumption and maximizing network lifetime is the foremost design objective for all type of sensor networks. A sensor network's main task is to perform sensing and delivering the collected data. Hence, a number of quality of service (QoS) objectives, such as maintaining sensing coverage

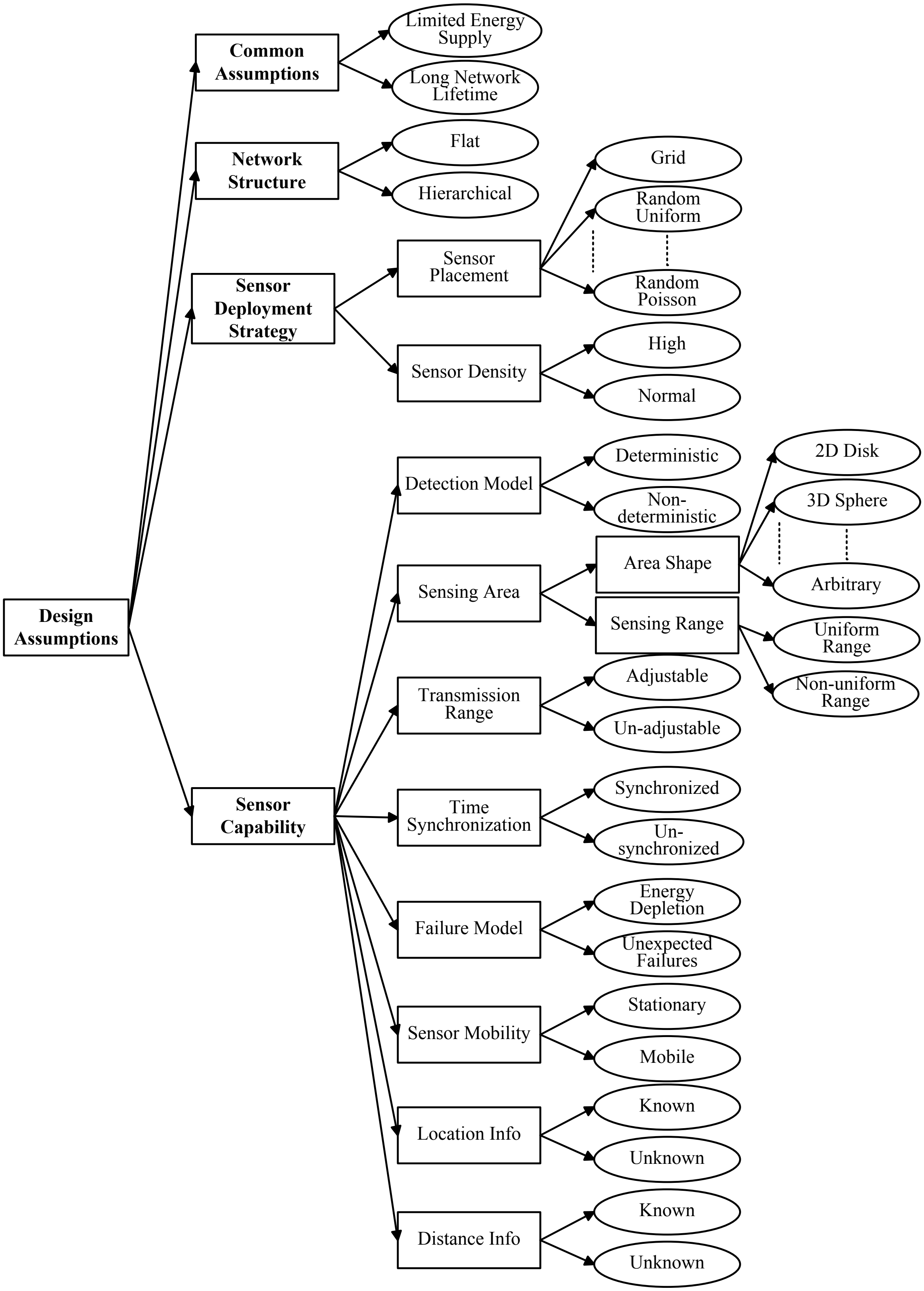


Figure 2.5: Design Assumptions of Density Control Mechanisms

and connectivity are usually considered together with maximizing the network lifetime. A summary of the design objectives is given in Figure 2.6. Below, we briefly discuss these various possible design objectives.

- **Maximizing Network Lifetime:** There are various definitions of network lifetime. As a simple definition, a network is considered to be alive when any of the nodes in the network is alive. Alternatively, a network lifetime can be defined as the time when the alive nodes percentage is above a certain threshold. Network lifetime can also be defined as the duration of time when the sensing coverage, connectivity or data delivery ratio is above an acceptable value.
- **Sensing Coverage:** Sensing coverage is an important metric for a sensor network. There are definitions like 1-coverage or k-coverage in which every point in the field is covered by at least 1 sensor or k sensors accordingly. Also, a sensor network may provide partial coverage or may ensure asymptotic coverage if deterministic coverage is not possible.
- **Network Connectivity:** It is important to maintain network connectivity when multi-hop communication is used among sensor nodes to transport data to the sink node. Some of the density control mechanisms may provide a specific degree of connectivity. Similar to sensing coverage, connectivity can be achieved asymptotically when the number of sensors goes to infinity.
- **Data Delivery Ratio:** A high data delivery ratio is another QoS objective for some of the applications and it is the percentage of data that can reach to the sink node. This ratio is applicable when there is no data aggregation in the network.
- **Quality of Surveillance:** This metric is proposed to measure the performance of target-tracking sensor networks. It is defined as the inverse of the average distance traveled by a target before it is detected by the sensor network. This implies that if the sensor network can detect a moving target within a shorter distance, it is considered to have higher quality of surveillance.

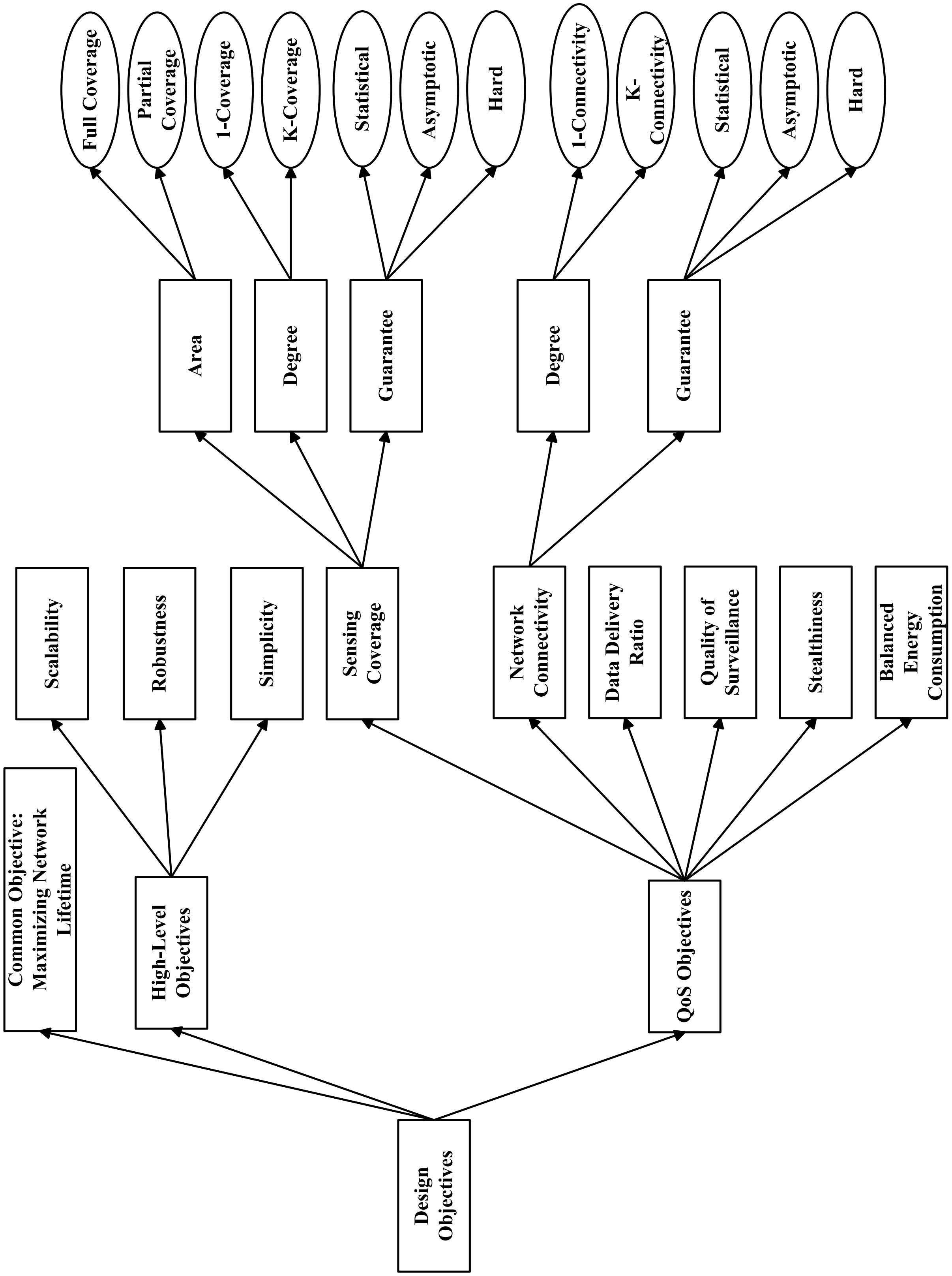


Figure 2.6: Design Objectives of Density Control Mechanisms

- **Stealthiness:** In some of the applications, it is desirable for a sensor network to be less likely to be detected by others. This goal can be achieved by shortening the communication time and reducing the number of control messages.
- **Balanced Energy Consumption:** Balancing energy consumption among sensor nodes may be required since if some nodes deplete their energy, holes may appear in the sensing coverage. Counter arguments also exist that states that there will still be redundant nodes that can be turned on even if those nodes deplete their energy.
- **Scalability:** It is generally undesirable for sensor nodes to have state or computation overhead that increases with the number of sensors.
- **Robustness:** Robustness is the ability of a sensor network to endure unexpected failures. A robust mechanism should not expect everything to go as planned. For instance, it cannot assume all the sleeping nodes to wake up or expect all the active nodes to function without any failures.
- **Simplicity:** Sensor nodes have very limited memory and computation capabilities. Henceforth, simpler mechanisms are much more favourable.

2.4 Algorithms and Schemes for Density Control

Density control is a common technique to prolong network lifetime by ensuring that only a required number of sensor nodes will be active at a time in the sensor network. In the meantime, maintaining a certain level of coverage and connectivity in the sensor network is also crucial for properly sensing the environment and collecting the data to the sink node.

Table 2.1: Radio Transmission Range of Berkeley Motes

Product	Transmission Range
MPR300(*)	30
MPR400CB	150
MPR410CB	300
MPR420CB	300
MPR500CA	150
MPR510CA	300
MPR520CA	300

* MPR300 is a second-generation sensor, while the rest are third-generation sensors.

The relationship between coverage and connectivity is important because designing algorithms to satisfy only one of them is easier than satisfying both of them. If connection infers coverage or vice versa, we can simply consider to satisfy only one of these objectives instead of satisfying both of them. In fact, Zhang and Hou[21] has derived that complete coverage of a convex region infers connectivity of the network if the transmission range of a node is at least twice its sensing range. To state this more precisely, let the sensing range and the transmission range of a sensor node be denoted as r_s and r_t, respectively. Then the lemma is as follows: [18]

Lemma 2.1 *Assuming the number of sensors in any finite area is finite, the condition of*

$$r_t \geq 2 \times r_s \tag{2.1}$$

is both necessary and sufficient to ensure that complete coverage of a convex region implies connectivity.

The necessary condition is shown by constructing a scenario in which coverage does not imply connectivity under the condition $r_t < 2 \times r_s$. The sufficient condition is shown by contradiction assuming that the network is not connected even though complete coverage is satisfied. Then, the resulting network must have a pair of disconnected nodes that has the shortest distance among all disconnected pairs. The proof proceeds by finding another pair of disconnected nodes with shorter distance.

Table 2.2: Sensing Range of Several Typical Sensors

Product	Sensing Range (m)	Typical Applications
HMC1002 Magnetometer sensor	5	Detecting disturbance from automobiles
Reflective type photoelectric sensor	1	Detecting targets of virtually any material
Thrubeam type photoelectric sensor	10	Detecting targets of virtually any material
Pyroelectric infrared sensor (RE814S)	30	Detecting moving objects
Acoustic sensor on Berkeley Motes	-1	Detecting acoustic sound sources

As shown in Tables 2.1 and 2.2, the condition that the transmission range of a sensor node is at least twice the sensing range (Eq. 2.1) holds for a wide spectrum of sensor devices. Therefore, focusing on the coverage problem in the network can be adequate most of the time rather than considering both coverage and connectivity. Also, if the transmission range is too large as compared to sensing range, than the network may be subject to excessive radio interference although its connectivity is ensured. Therefore, it would be better for sensor nodes to adjust their transmission range around twice the sensing range.

2.4.1 Power Management Schemes in Wireless Ad Hoc networks

In wireless ad hoc networks, minimizing energy consumption and extending the network lifetime are the main design objectives. Among many studies done in this area, there are studies that deal also with maintaining coverage and connectivity.

Geographical Adaptive Fidelity (GAF) algorithm [19] assumes nodes know their location information and nodes use this location information to associate themselves with virtual grids. In the definition of the virtual grid, it is required that sensor nodes in adjacent grids can communicate with each other. Sensed area is assumed to be divided into these rectangular grids and only one sensor node stays awake in each grid at any time to perform sensing and communication tasks. Henceforth, working sensor node count is decreased and energy saving is achieved in the network.

SPAN [4], also, in order to minimize energy consumption and extend network lifetime, keeps some of the sensor nodes in active mode and others in sleep mode. The ones that finally stay active are called coordinator nodes. Nodes locally determine whether they should join to the coordinator set of nodes for accomplishing sensing and communication tasks. If two neighbours of a non-coordinator node cannot communicate with each other directly or via intermediate coordinators, then the node becomes a coordinator node as well. These decisions of nodes are exchanged among neighbours with HELLO messages. After constructing a backbone of coordinators, data is routed in this forwarding backbone so that each coordinator tries to relay the data to another coordinator that is closer to the sink node.

The main difference between wireless ad-hoc networks and sensor networks are twofold from the perspective of energy savings. First of all, algorithms used for wireless ad hoc networks do not take into consideration the sensing coverage issue. Secondly, minimizing energy consumption is a common design objective for both of the networks, but schemes used for wireless ad hoc networks try to maximize the lifetime of individual nodes, whereas mechanisms used for wireless sensor networks try to maximize the whole network lifetime while trying to assure a certain level of sensing coverage and connectivity. As long as there is a sufficient sensing coverage and network connectivity, a sensor network can be regarded as functioning properly although some nodes die earlier than the others.

In the following sections, we will discuss some of the centralized and distributed density control mechanisms that ensure sensing coverage and connectivity in sensor networks. Categorization into centralized and distributed algorithms is done roughly. Most of the studies are based on distributed mechanisms. Centralized algorithms can be a baseline for distributed algorithms and can be used to study performance limits [18].

2.4.2 Centralized Algorithms

Slijepcevic et al. [13] propose a solution that focuses on finding the maximum number of node sets in the network such that each set provides full coverage of the sensing area. In their study, they show the NP-completeness of the problem. They define the active node sets that fully cover the sensing area as *cover* and give a heuristic algorithm to find the maximum number of covers. Initially, all the points in the monitored area are put into disjoint fields, namely, the maximum number of points covered by the same set of nodes. At each iteration of the cover determination, from the unchosen sensor nodes, a node is continuously selected and added to the current candidate cover, until full coverage is achieved by the cover. Node selection is done such that the node selected has the highest objective function among the nodes that cover the critical field, i.e., field covered by the smallest number of unchosen nodes. If the candidate cover set provides full coverage of the region, the cover is added to the set of the covers and the algorithm goes to next iteration. The iterations of the algorithm continue until the remaining nodes cannot fully cover the whole sensing area.

In [6], Gupta et al. design an algorithm to find a subset of nodes called connected sensor cover that satisfies both coverage and connectivity objectives. Initially, a sensor node is added to a set A which is a connected sensor cover, randomly. In each pass, the set of candidate sensors that have the overlapping sensing region with the sensors in A is determined. Each of these candidate sensors has a candidate path that is connecting it to a sensor node in A. From these candidate paths, the path that has the greatest number of sub-elements per sensor is added to A where sub-element is the maximum number of points covered by same sensor nodes.

2.4.3 Distributed Algorithms

ASCENT [3], is a self-organizing mechanism that consists of different phases to find the set of active sensor nodes in the network. At the beginning of the protocol, nodes start with neighbour discovery phase where only a small portion of the nodes are active in the network. After this phase, nodes enter to a join decision phase to decide whether to be active or not based on factors like active neighbour size, whether it received a help message from a neighbour node that indicates a high data loss. According to the decision made, nodes can either enter the active mode to participate in network operations or enter the adaptive phase to get into sleep mode. ASCENT, using different heuristics in several phases, does not ensure the full coverage of the sensing area.

Tian et al. [15] propose a sponsored area approach that provides full coverage of the monitored area. A node X's sponsored area provided by a node Y is the overlapping sensing area of both nodes' sensing areas. At each round, nodes initially send a Position Advertisement Message (PAM). Each node calculates the sponsored area provided by its neighbours after getting this PAM message from each neighbour. If sponsored areas provided by its neighbours cover the sensing area of the node completely, then the node can safely be put into sleep mode. In this method, nodes need location information and are time-synchronized in order to know the beginning of the rounds. Random back-off mechanism is used to prevent simultaneous actions, and there is a message overhead of advertising location information and scheduling.

PEAS [20], is a mechanism that can extend the lifetime of a high-density sensor network in a harsh environment. The algorithm assumes that nodes may fail frequently and unexpectedly. For that reason, a sleeping node wakes up and probes its environment at certain time intervals to see if there is a working neighbour node. If there is not any working neighbour, then the node enters the active mode. After probing, if there is an active node nearby, then the node goes to sleep mode to wake up at a later time. When the node becomes active, it will stay in the active mode until it depletes its energy, which may result in unbalanced energy consumption. And also, the mechanism does not guarantee complete coverage.

Wang et al. [17], propose an integrated coverage and connectivity configuration (CCP) scheme. The protocol tries to maximize the number of nodes in sleeping mode while maintaining k-coverage, i.e., sensing range of a node is covered by at least k sensor nodes other than itself. They examine the relationship between coverage and connectivity and they prove that if the transmission range is greater than twice the sensing range, then the k-coverage of the region implies k-connectivity in the network. In CCP, all nodes begin in active mode. Nodes go to sleep mode when they determine that their sensing area is k-covered by neighbour nodes. A node in sleeping mode periodically goes to listen mode and checks if its sensing area is k-covered. If not it switches into active mode. If the transmission range is less than twice the sensing range, CCP does not ensure the connectivity of the network. In this case, they propose to integrate SPAN [4] into their protocol such that network connectivity is provided while putting unnecessary sensor nodes into sleep mode.

OGDC [21] is a scheme for maintaining both coverage and connectivity while trying to maximize the number of sleeping nodes. They prove that coverage implies connectivity if the transmission range is greater than twice the sensing range. Assuming this condition and time synchronization of nodes, the scheme is designed to operate in rounds. Each round consists of a selection phase and a steady-state phase. In selection phase, initially, some random sensor nodes are selected as starting working nodes. Then, a sensor node chooses to be in active mode if it minimizes the overlapping area with the existing working nodes and covers the intersection point of two working nodes. Location information is used to do these. In the steady state phase, nodes keep their active/sleep modes until the next round. OGDC can also relax its assumptions. If the transmission range is not at least twice the sensing range, OGDC extends the selection mechanism such that a node is turned off only if its sensing coverage is covered by other nodes and connectivity is not affected from putting it in sleep mode.

Chapter 3

Overview

In this section, first we will provide some preliminary information and definitions about our network model. Then the problem of finding active sets to prolong network lifetime while satisfying a desired sensing coverage is introduced. After we give a formal description of the problem, we will introduce and discuss the notion of graph partitioning which is the approach that we use in determining active sets.

3.1 Network Model

In this paper, we consider a sensor network that is composed of sensor nodes $s_i, i = 1...n$. Hence there are n sensor nodes that are initially deployed to a target region. Each sensor node has an associated sensing range and transmission range. The sensing range associated with each sensor node s_i is considered to be a circular area around its position, that can be monitored successfully by the sensor node. Similarly for the transmission range, each sensor node s_i can communicate with the sensor nodes in its transmission range, i.e., the communication coverage is the circular area centered at the position of the sensor node and with a specific radius. We assume all sensor nodes have the same fixed sensing and transmission range.

Let's sensing range and transmission range of a sensor node are represented by r_s and r_t, respectively. We will give some formal definitions about sensing and transmission ranges of sensor nodes as follows: [14]

Definition 3.1 ***Sensing Coverage of a Point:*** *A sensor node s_i is considered to cover a point p if and only if the Euclidean distance $d(p, s_i) \leq r_s$, i.e. , point p is in the sensing range of the sensor s_i.*

Similarly, we can define the sensing coverage of an area as below:

Definition 3.2 ***Sensing Coverage of an Area:*** *A sensor node s_i is considered to cover an area if and only if for every point p in the area, the condition $d(p, s_i) \leq r_s$ is achieved.*

Two sensor nodes can communicate with each other if and only if each sensor node is in the transmission range of the other node. Formally:

Definition 3.3 ***Direct Communication:*** *Sensor nodes s_i and s_j can communicate directly with each other if and only if the condition $d(s_i, s_j) \leq r_t$ is fulfilled.*

Definition 3.4 ***Communication Graph:*** *Given a sensor network composed of n sensor nodes, the communication graph of the network is an undirected graph $G = (V, E)$ where V is a set of sensor nodes and E is a set of edges that exist between sensor nodes having direct communication.*

In the case that only a subset of nodes are active and working, the communication subgraph induced by the set of these active sensor nodes is the subgraph of G that has only the set of active sensor nodes and the edges of G that connect active nodes.

In addition, in this work, we consider the following issues about a sensor network:

- Sensor nodes are randomly distributed in a region and densely deployed.
- Base station is at the center of the monitored region and has the location information of the sensor nodes.
- All sensor nodes and base station are immobile.
- All the nodes are homogeneous, have the same initial limited amount of energy.
- Base station is resource-rich, i.e., it does not have any computation, communication and energy limitations.
- Data is sent periodically from the sensor nodes to the base station.
- A sensor node s_i is in active mode if the node participates in sensing and communication tasks; it is in low-energy state, i.e., in sleep mode, if it is not required to participate in the sensing, processing and communication operations and therefore turned off.

3.2 Problem Definition

To minimize energy consumption induced by sensing and communication tasks in high-density sensor networks, we can partition a given sensor network (i.e. a given set of sensor nodes) into large number of disjoint subset of nodes (active node sets) such that only one subset of nodes will be active at any time and subsets will alternate to be active. We call such a subset as an active node set candidate or simply an *active set*. Hence the network will be partitioned into a number of active node sets. Each active node set should satisfy the following properties:

- Each active node set must provide the desired sensing coverage level of the monitored region.

- In order to send the sensory data, sensor nodes in the active set should successfully communicate with each other, i.e., the network formed by the nodes in the active set must also be connected.

In a densely deployed sensor network environment, sensing coverage of a sensor node is usually overlapped with the sensing coverages of other sensor nodes as well. As a result, the sensing area of some nodes may be completely covered by other nodes. Thus, it is not necessary for all of the sensor nodes in the network to be active at the same time. We can use this redundancy to increase the network lifetime by keeping some of the sensors in active mode and the remaining ones in sleep mode while satisfying the desired constraints. Then, we can define the problem that we will deal with as follows:

Consider a set of sensor nodes initially placed randomly in a region. In order to extend the network lifetime, we are interested in organizing the given sensor nodes into a number of active node subsets, such that each of these subsets is able to individually monitor the area of interest while maintaining the requested level of sensing coverage of the region.

The formal definition of active node set is as follows. Given a sensor network consisting of n sensor nodes, a set A of sensor nodes is called active node set if it satisfies the following two conditions:

1. $Coverage_{req} \leq \cup_{a \in A} Coverage(a)$ where $Coverage_{req}$ is required level of the sensing coverage of the region and $Coverage(a)$ is the sensing coverage of sensor node $a \in A$.
2. The communication graph induced by the set A is connected.

3.3 Graph Partitioning Problem

We use graph partitioning as the approach to obtain active sets from a given set of sensor nodes. Therefore we would like to introduce and discuss the graph partitioning problem in this section.

Graph partitioning is an important problem that can be used in the solutions of a wide range of applications in many areas such as scientific computing, VLSI design, storing and accessing spatial databases on disks and transportation management. Graph partitioning problem is to partition the vertices of a graph into two or more parts such that the number of edges connecting different parts is minimized. The formal definition of the problem is as follows:

Definition 3.5 *Given a graph* $G = (V, E)$ *with* $|V| = n$*, partition* V *into* k *roughly equal subsets,* $V_1, V_2, ..., V_k$ *such that* $V_i \cap V_j = \emptyset$ *for* $i \neq j$*,* $|V_i| = n/k$ *and* $\bigcup_i V_i = V$*, and the number of edges of* E *whose incident vertices belong to different subsets is minimized.*

Graph partitioning problem can be extended to graphs that have weights associated with the vertices and edges. In this case, the goal is to partition the vertices of a graph into k disjoint subsets such that sum of the weights of the edges connecting vertices in different sets is minimized and the sum of the vertex weights in each set is balanced. Given a partition, the number of edges (in the case of weighted graphs the sum of edge weights) whose incident vertices belong to different subsets is called edge-cut of the partition. Generally, the task of minimizing the edge-cut can be considered as the objective, and the requirement that the partitions will roughly have the same size can be considered as the constraint. Then, the extension we mentioned to the graph partitioning problem, i.e., multi-constraint graph partitioning problem is formally defined in [10] is as below:

Definition 3.6 *Given a graph* $G = (V, E)$ *such that each vertex* $v \in V$ *has a weight vector* w^v *of size* m *associated with it and each edge* $e \in E$ *has a scalar weight* w^e*. We will assume that* $\sum_{\forall v \in V} w_i^v = 1.0$ *for* $i = 1, 2, ..., m$*.*

Let P *be the partitioning vector of size* $|V|$*, such that for each vertex* v*,* $P[v]$ *stores the partition number that* v *belongs to. For any such* k*-way partitioning vector,* *the load imbalance l_i with respect to the i^{th} weight of the k-way partitioning is defined as*

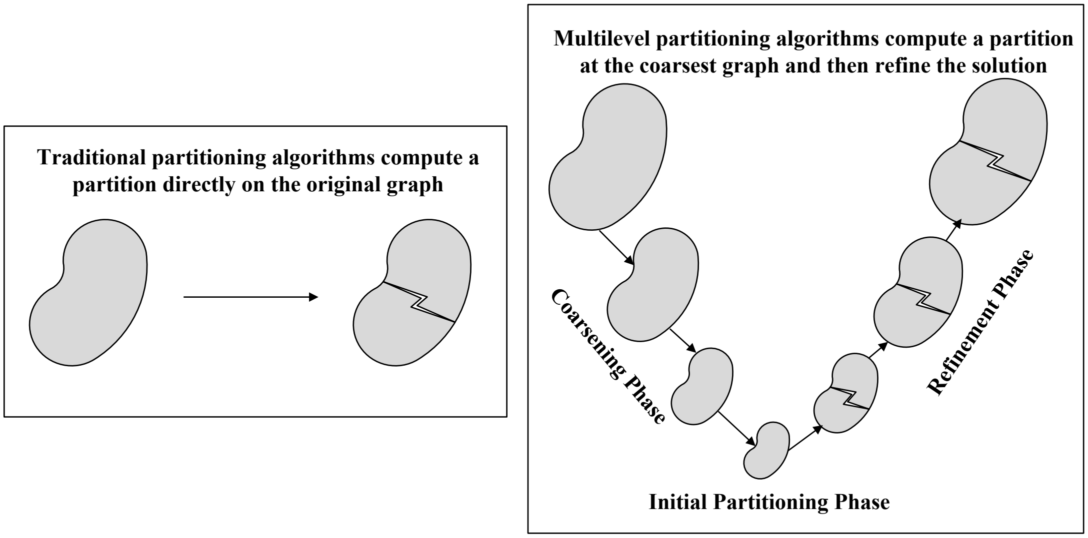
$$l_i = k \times max_j(\sum_{\forall v:P[v]=j} w_i^v) \tag{3.1}$$

A load imbalance $l_i = 1 + \alpha$ indicates that the partitioning is load imbalanced by α%. Then the multi-constraint graph partitioning problem is to find a k-way partitioning P of G such that the sum of the weights of the edges that are cut by the partitioning is minimized subject to the constraint

$$\forall i, l_i \leq c_i, \tag{3.2}$$

where c is a vector of size m such that $\forall i, c_i \geq 1.0$

The graph partitioning problem is NP-complete. However, there are various heuristics that compute fairly good partitions such as spectral partitioning methods, geometric partitioning algorithms and multilevel graph partitioning algorithms. Spectral partitioning methods are commonly used for different problems but these methods are very expensive in the sense that they require high processing time. Secondly, there are geometric partitioning algorithms and as the name implies, these algorithms use the geometric information of the graph to compute a partition of the graph. Geometric partitioning algorithms tend to be fast but often produce partitions that are worse than the partitions of spectral algorithms. Third class of algorithms are multivel graph partitioning algorithms [9, 7, 11]. These type of algorithms have completely different approach than the other traditional graph partitioning algorithms. While traditional graph partitioning algorithms compute the partitions of the graph by operating directly on the original graph, multilevel graph partitioning algorithms construct a coarser graph and partition the coarsest graph which is then refined locally. Traditional and multilevel partitioning algorithms are illustrated in Figure 3.1 [8, 9, 10]. In Figure 3.1(a), we see that the traditional partitioning algorithms perform a partition directly on the original graph. In Figure 3.1(b), the phases of the multilevel



(a) (b)

Figure 3.1: Traditional and Multivel Partitioning Algorithms

partitioning algorithms is illustrated.

Multilevel framework is the current state of the art in graph partitioning. As mentioned earlier, these multilevel graph partitioning algorithms consist of different phases which are constructing a coarse graph, partitioning the coarsest graph and uncoarsening the partition. Firstly, the size of the graph is reduced by collapsing vertices and edges. Then, the algorithm partitions the smallest graph at a low cost. Finally, there is a refinement phase in which algorithm uncoarsenes the smallest partition to construct a partition to the original graph. By this way, these type of algorithms can be implemented in time proportional to the original graph size and therefore good partitions of the graph can be obtained at low cost [9].

Chapter 4

Active Set Partitioning Scheme

In this section, we present our proposed method for partitioning a given set of nodes of a sensor network into a number of disjoint subsets where the nodes in only one subset will be active at a given time while the nodes in all other subsets will be sleeping. Additionally, each subset should be a connected subset. We call our scheme as Active Set Partitioning scheme (ASP), a network partitioning scheme to obtain set of subsets to run independently at different times. While dividing the network into a set of connected active node sets, ASP aims at keeping the sensing coverage of the network above a certain desired level while saving energy by just keeping one subset of nodes as active.

While obtaining multiple active set candidates, the objective of our scheme is to minimize the overlap among the sensing coverages of sensor nodes that go to the same set. If we consider the distances among nodes, we can say that the closer the nodes are, the higher the overlap among their sensing coverages. Based on this observation, nodes having overlapping coverages do not have to be active at the same, but can be active at different times, hence be part of different subsets.

Additionally, it is important to have a scheme that does not take too much time while determining subsets of nodes. In other words, the algorithm should be runtime efficient. Our proposed scheme aims at partitioning the nodes into a number of appropriate active node sets as fast as possible for a given large wireless sensor network.

We use the idea that more distant nodes in the region should be in the same subset since they are less likely to cover the same region unnecessary. In other words, the sensor nodes that are closer to each other must be active during different time intervals to conserve energy; hence, they must be in different active node sets.

We propose a top-down approach based on recursive graph bipartitioning for generating active node sets from a given set of sensor nodes. The proposed algorithm works as follows.

We are given a set of nodes, their positions, and a fixed maximum transmission range that is the same for all nodes. Using this information, we can represent the sensor network as an undirected graph $G = (V, E)$ where V is the set of vertices and E is the set of edges. In such a graph, vertices of the graph correspond to the sensor nodes in the WSN and there is an edge between two vertices if the corresponding sensor nodes are in transmission range of each other. Each of these edges has an associated weight that is equal to the distance between the sensor nodes.

Our ASP algorithm starts with the construction of the initial graph from the given sensor nodes and their positions. It then keeps moving from this bigger graph to the smaller graphs by obtaining a bipartitioning of each respective graph (Figure 4.1). Each bipartition $\prod(G) = \{V_1, V_2\}$ is decoded as inducing two tentative sets as follows.

The sensor nodes corresponding to the vertices on each part of a bipartitioning induces an active node set. After each bipartitioning step, each of the two induced active node sets V_1 and V_2 are checked for sensing coverage and network connectivity constraints. If an active node set is found to satisfy both coverage and connectivity constraints, then the vertex induced subgraph $G_1 = (V_1, E_1)$ of G is further considered for bipartitioning. Here, $G1$ is the subgraph of G induced by the vertex set $V1$ of $\prod$. That is, $G1$ contains the vertices and internal edges of part $V1$, where the cut edges are discarded.

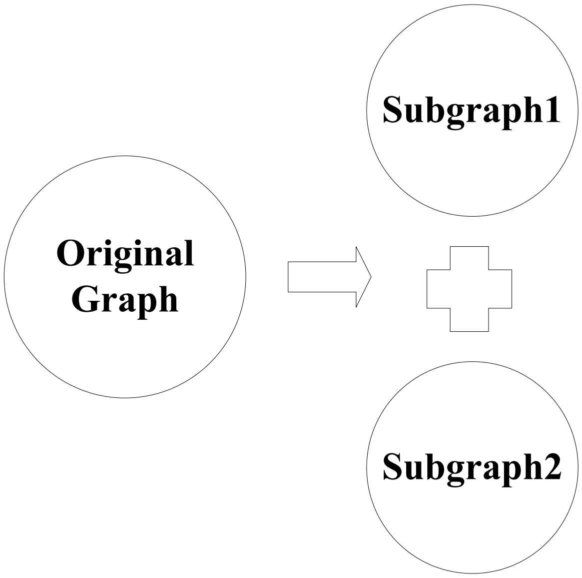


Figure 4.1: Two-way Partitioning of a Graph

At each bipartitioning step, it is checked if the desired conditions hold for the obtained two subgraphs (i.e., parts or subsets). There are three cases which can be classified as follows:

- Both subgraphs satisfy the constraints,
- Neither of them satisfy the constraints,
- One of the subgraphs satisfies the constraints.

If both subgraphs induced by a bipartitioning step satisfy the constraints, we will continue partitioning with the subgraphs that satisfy the constraints. If neither of the subgraphs ensure the coverage and connectivity objectives, we will select the parent graph as the active node set and the recursive bipartitioning is terminated at that point of the overall recursive bipartitioning tree. In the last case, if either of the subgraphs does not meet our requirements, we will keep the subgraph not meeting the requirements to be examined later. The partitioning step is repeated until there is no remaining subgraph to be investigated.

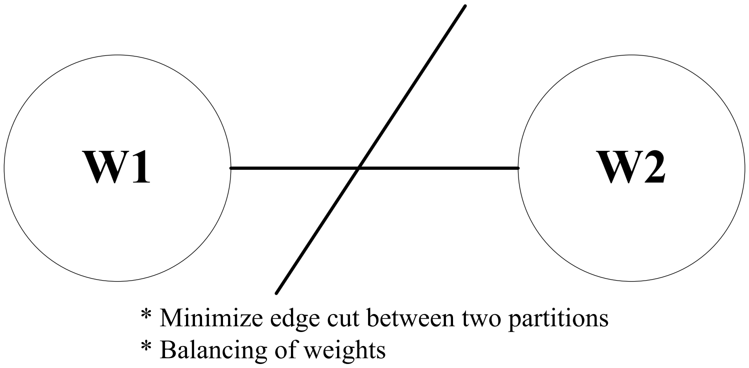


Figure 4.2: Graph Partitioning Problem Objectives

At the end of the first iteration of the algorithm, we obtain a set of subgraphs where the nodes in a subgraph can act as an active node set satisfying the desired constraints. At the end of the first iteration, we also have some subgraphs that come from the third case above, i.e., subgraphs that do not satisfy the desired constraints. These are child subgraphs obtained after a bipartitioning and that do not satify the constaints even though the corresponding sibling subgraphs satisfy the constaints. In the second iteration of the ASP algorithm, first we merge the subgraphs that do not ensure coverage and connectivity into a bigger graph. Then we start a new partitioning process on the obtained graph from the merge operation, and we try to find additional active node sets in a similar way as performed in the first phase of the algorithm.

Our algorithm stops after the second iteration since there is not much improvement that can be obtained with further iterations. Theoretically, however, the algorithm can be modified to iterate until there is no remaining set of subgraphs that do not meet the requirements.

The complete active set determination algorithm pseudo-code can be seen in Algorithm 1.

In Figure 4.2, a broad view of our approach to active set partitioning problem is given. At the core of the approach, there is the partitioning of a bigger graph to smaller graphs ensuring needed requirements. That is to say, the solution to active set determination problem is closely related to the solution to the partitioning

Algorithm 1 Active Set Partitioning Algorithm

```
Require: G is the initial graph of alive sensors
    Q is a queue of graphs
    A is a collection of active node sets
    B is a set of graphs where $\forall b \in B : SatConst(b) = True$
    S is a set of graphs where $\forall s \in S : SatConst(s) = False$
if SatConst(G) = False then
    return
ENQUEUE(Q, G)
while $|Q| \neq \emptyset$ do
    $g \leftarrow DEQUEUE(Q)$
    Partition g into $g_1$ and $g_2$
    if $SatConst(g_1) = True$ and $SatConst(g_2) = True$ then
        ENQUEUE(Q, $g_1$)
        ENQUEUE(Q, $g_2$)
    else if $SatConst(g_1) = False$ and $SatConst(g_2) = False$ then
        $A \leftarrow A \cup g$
    else if $SatConst(g_1) = True$ then
        $B \leftarrow B \cup g_1$
        $S \leftarrow S \cup g_2$
    else
        $B \leftarrow B \cup g_2$
        $S \leftarrow S \cup g_1$
if $|S| > 1$ then
    $g \leftarrow MERGE(S)$
    if SatConst(g) = False then
        $A \leftarrow A \cup B$
        return
    ENQUEUE(Q, $g$)
    for all $b$ such that $b \in B$ do
        ENQUEUE(Q, $b$)
    Clear the sets $B$ and $S$
    while $|Q| \neq \emptyset$ do
        $g \leftarrow DEQUEUE(Q)$
        Partition g into $g_1$ and $g_2$
        if $SatConst(g_1) = True$ and $SatConst(g_2) = True$ then
            ENQUEUE(Q, $g_1$)
            ENQUEUE(Q, $g_2$)
        else if $SatConst(g_1) = False$ and $SatConst(g_2) = False$ then
            $A \leftarrow A \cup g$
        else if $SatConst(g_1) = True$ then
            $A \leftarrow A \cup g_1$
        else
            $A \leftarrow A \cup g_2$
```

problem. Our observation that closer sensor nodes in the network should be active at different time durations for energy efficiency corresponds to the objective of minimizing the edge-cut in the graph partitioning problem (Figure 4.2). During the partitioning process of a graph, the graph partitioning algorithm is expected to put the sensor nodes that are closer to each other to distinct subgraphs instead of putting distant nodes in the network to distinct graphs. Consequently, graph partitioning algorithm serves our purpose and helps to distribute the nodes in different active node sets, while maintaining coverage and connectivity of the region.

As we already know, while obtaining partitions for a graph, a multi-constraint partitioning algorithm tries to balance the given constraints among the partitions if there exists constraints on the vertices. In order to obtain better final partitions, we can consider several factors such as node size, residual energy, degree or distance to base station of sensor nodes. We try to use these considered factors as balancing constraints for the graph partitioning process. By the help of these balancing factors, we try to achieve partitions that better meet our objectives, i.e., we focus on saving energy consumption of the network, but at the same time, we should also maintain network coverage and the network must be connected to function properly.

The proposed algorithm, basically, aims to expand and examine all child subgraphs that are originated from a two-way partitioning of an initial graph that corresponds to a wireless sensor network. The child subgraphs obtained by partitioning a subgraph are added to a queue if it is acceptable to do so. It exhaustively continues to perform a two-way partitioning of subgraphs that are not examined, i.e. subgraphs in the queue, until the queue is empty. In essence, the algorithm resembles breadth first search algorithm and can be seen as constructing a tree of subgraphs where each subgraph is subject to a two-way partitioning. As can be seen in the pseudocode of Active Set Partitioning algorithm 1, G is taken as an input that is the initial graph of alive sensors. Q is the collection of graphs in which the graphs to be investigated are kept in order. Initially, Q is empty and the algorithms starts with the addition of initial graph G to the queue. Subsequently, the graphs in the queue are removed one by one from the queue and are partitioned into smaller subgraphs that are investigated for fulfilling the requirements needed. A is the collection that we keep the determined active node sets. There are also two collection sets B and S that we keep some of the child subgraphs temporarily. As we have mentioned previously, according to the decision made for satisfying the requirements needed, we keep some of the child subgraphs in these temporary collections. A is the collection that consists of the final active node sets that are determined and is the output of the algorithm.

The time complexity of the algorithm has two factors. The first factor is related to the number of vertices and edges in the tree showing how many subgraphs we will have. This tells us how many two-way partitioning we will perform. The second factor is the complexity of the partitioning algorithm. Here, it is convenient to use a multilevel partitioning algorithm because it is shown to produce rather high quality partitions in a fast and stable manner. Expressing the complexity of a multilevel partitioning algorithm is a difficult task, as it consists of three main phases and each phase has a number of different algorithms to be investigated. But, we can roughly consider the complexity of the partitioning proportional to the number of the edges in the input graphs ($O(|E|)$). Then the total runtime complexity of active set partitioning algorithm is proportional to the level of subgraph tree (k) times the number of edges (E) in the input graph. That is, the complexity of the ASP algorithm for finding active node sets of a graph $G = (V, E)$ which is the representation of a given sensor network, is $O(k\,|E|)$.

Chapter 5

Simulation and Results

In this chapter, we evaluate the performance of our active set partitioning approach through simulations. First of all, we will discuss the assumptions and parameters of the simulation environment. Then, we will present and discuss the results of simulations that we performed.

5.1 Simulation Setup

In our simulations, we evaluated our proposed scheme under both uniform and nonuniform distributions of sensor nodes to a region. For the nonuniform case, the deployment of nodes is around the location of the sink node and is according to Gaussian distribution having mean and standard deviation as parameters.

In each simulation trial, we considered 5 different initial placements and the average results of these experiments is reported. In the simulations, if it is not stated explicitly, we considered a uniform distribution of sensor nodes into a region of $500 \times 500m^2$, which is considered to be virtually divided into $50 \times 50m^2$ grid cells. Then, the sensing range of a sensor node (r_s) is accepted as $50\sqrt{2}m$ where the transmission range (r_t) is set to $2r_s$. Also, we consider a set of nodes as an active node set if its sensing coverage level is above 90%.

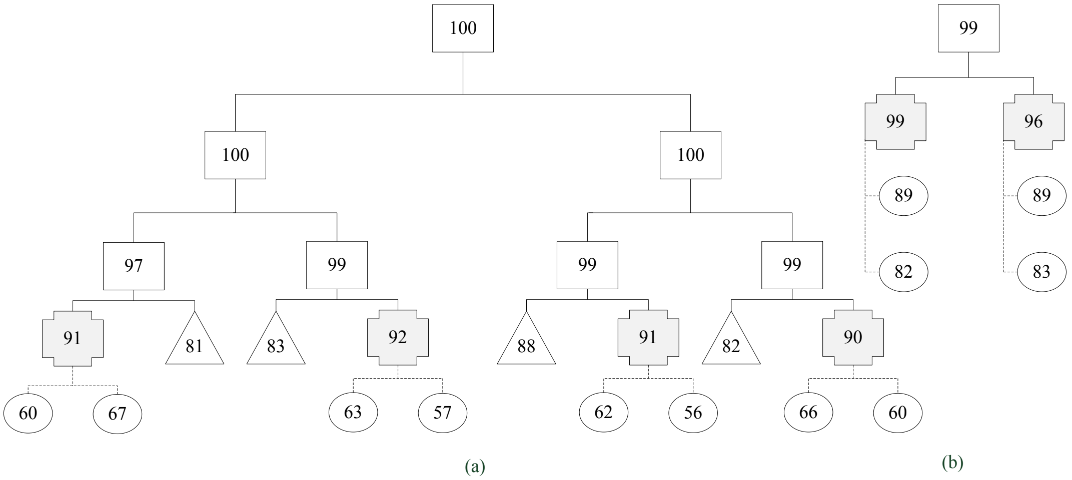


Figure 5.1: Illustration of Active Set Partitioning Scheme ($Coverage_{req} = 90\%$)

We consider measuring the coverage as follows. The area is divided into 1m by 1m tiny grid cells and each tiny grid cell is assumed to be covered if the center of it is covered by at least one sensor node. Then, the coverage of the region is the ratio of the total number of covered tiny grids to the number of all tiny grids [18].

We performed our simulations in Linux environment with a custom simulator implemented with Java programming language. The Linux machine has Centos operating system with 6 GB RAM and 2.4 GHz Quad-Core Processor.

The proposed approach uses graph partitioning. Therefore, the success of our approach depends on the effectiveness of the suggested solution to the graph partitioning problem. Graph partitioning problem is a well-known problem that can be solved using a variety of available techniques. We have previously discussed these various techniques to solve the graph partitioning problem. As we mentioned earlier, multilevel graph partitioning is the current state of the art. Therefore, in our work, we use multilevel graph partitioning technique.

Our aim is to maximize the number of active sets with an algorithm that is runtime efficient, i.e., that has short run-time. Therefore we used a fast graph partitioning approach that is producing still good quality partitions. For that reason we used one of the existing partitioning tools that is based on the state-of-the art multilevel graph partitioning paradigm. The tool is called *Metis* and includes a family of multilevel partitioning algorithms which are based on multilevel recursive bisection scheme given in [9] and multilevel k-way partitioning scheme described in [11]. Both of the schemes produce high quality partitions and are extremely fast. In ASP, considering bipartitioning of a graph $G = (V, E)$, the time complexities of both schemes are comparable and take $O(|E|)$ time. We prefer to use the multilevel partitioning algorithm that is based on multilevel recursive bisection scheme, which is claimed to give better results for partitioning a graph into small number of partitions [8].

5.2 Simulation Results

In this section, we will present the results of the simulation experiments that we performed. First, we will provide an example for our active set partitioning algorithm. Then, we investigate the runtime efficiency of our algorithm through some experiments considering different parameters. Then, we continue with the evaluation of our algorithm under various assumptions and network topologies.

In Figure 5.1, we see the illustration of active set partitioning scheme under distribution of 200 sensors in the field. In the figure, subgraphs that are determined as active node sets satisfying the desired constraints, have plus sign. If only one of the subgraphs obtained by bipartitioning does not meet the requirements needed, it is illustrated with triangular shape. In the first phase of the algorithm, we see that there are multiple two-way partitionings which results in different subsets of nodes. After the first phase, some of the subsets satisy the coverage and connectivity constraints and some of them do not meet the desired objectives. From the obtained subsets of nodes, subsets of nodes that fall into third condition are merged to obtain a bigger graph and start a new partitioning process. As can be seen in Figure 5.1, six active node sets that meet the desired requirements are obtained at the end of scheme.

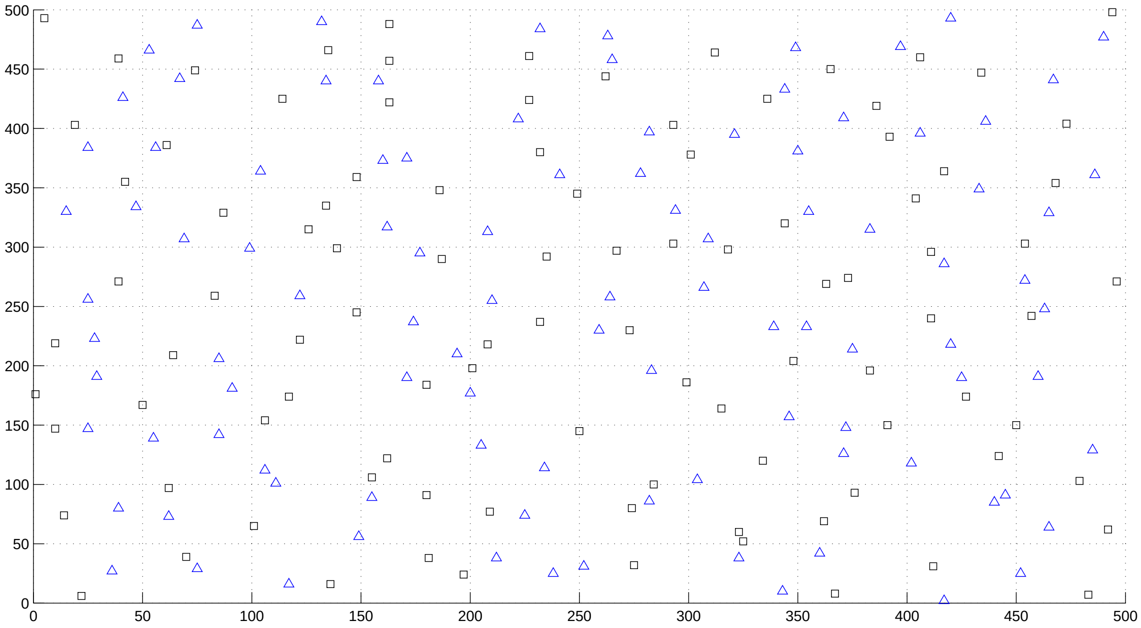


Figure 5.2: Two subgraphs (□, △) obtained by one level bipartitioning of a graph with 200 nodes.

Figure 5.2 shows a sample graph during the partitioning process and the subgraphs obtained from a two-way partitioning of this graph. Here, it is seen that our motivation that closer nodes should be in different active node sets holds for obtained subgraphs. This is because, if we look at closer nodes, they are distributed between the two subgraphs that are obtained by one level bipartitioning.

We performed experiments to evaluate the partitioning algorithm with different parameters. We investigated the partitioning process runtime efficiency by trying different network graphs with varying sizes as input to the partitioning algorithm (Table 5.1). All times reported are in seconds. Two different input graphs is considered. Initially, we construct a dense graph in which there is an edge between all the nodes in the graph. In the second case, in order to decrease the input graph size, we consider not to put an edge between sensor nodes if the distance is smaller than sensing range/4. Our opinion is that the removal of shorter edges will not affect the partitioning quality. Because, graph partitioning main objective is to minimize edge cut and removing short edges in the graph is convenient to this idea. In conjuction with, we observe how different choices affect the number of active node sets and see that the final number of partitions

Table 5.1: Average Running Times

Deployed Nodes	Dense Graph	Sparse Graph
500	7.82	8.11
1000	19.16	17.72
2000	64.53	51.77
4000	235.08	160.73
8000	1727.77	952.12

in both case is closer to each other. Results of the conducted experiments can be seen in Figure 5.4. As a result of the experiments we can clearly state that the proposed scheme having a sparse graph as input gives lower running times compared to dense graph. When there is an increase in the number of deployed nodes from 500 to 8000 nodes, the difference between their running times increases to nearly 20% (Figure 5.3). Meanwhile, the scheme applied to the sparse graph preserves the number of active node sets in the same level of active node set size that is obtained through a dense input graph.

Figure 5.5 shows the effect of decreasing the acceptable sensing coverage on the number of active node sets. As seen in the figure, with decreasing acceptable sensing coverage, obtained active node sets size increases. Then, applications can adjust the desired level of sensing coverage according to their needs. Higher number of active node sets can result in an higher network lifetime if we consider rotating the role of region monitoring among all the sets periodically. In addition, how the average number of nodes in an active set changes with an increase in the acceptable coverage level is illustrated in the next figure, where the average results of deploying 8000 nodes in the region is given. As can be seen in the Figure 5.6, maintaining full coverage of the region requires greater number of nodes to be active at the same time. As we decrease the level of the required coverage of the region, less number of nodes is sufficient to ensure the desired coverage level. Here, it is observed that the relationship between the number of nodes in an active set and the required coverage level is not linear.

In Figure 5.7, we compare the number of active node sets with varying ratios of sensing range and transmission ranges of sensor nodes under deployment of

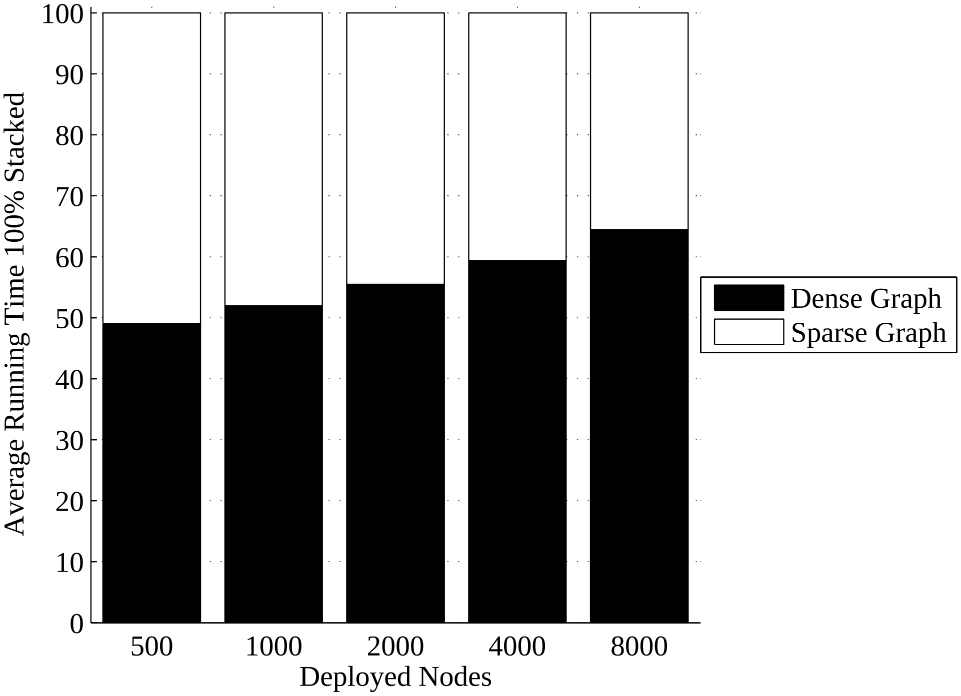


Figure 5.3: Average Running Times

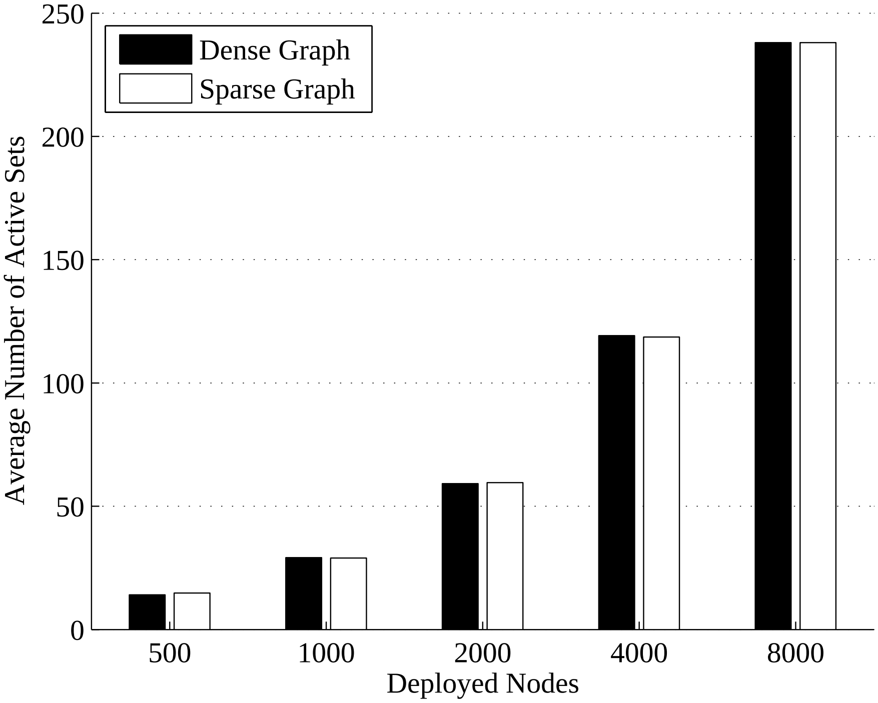


Figure 5.4: Number of Active Node Sets

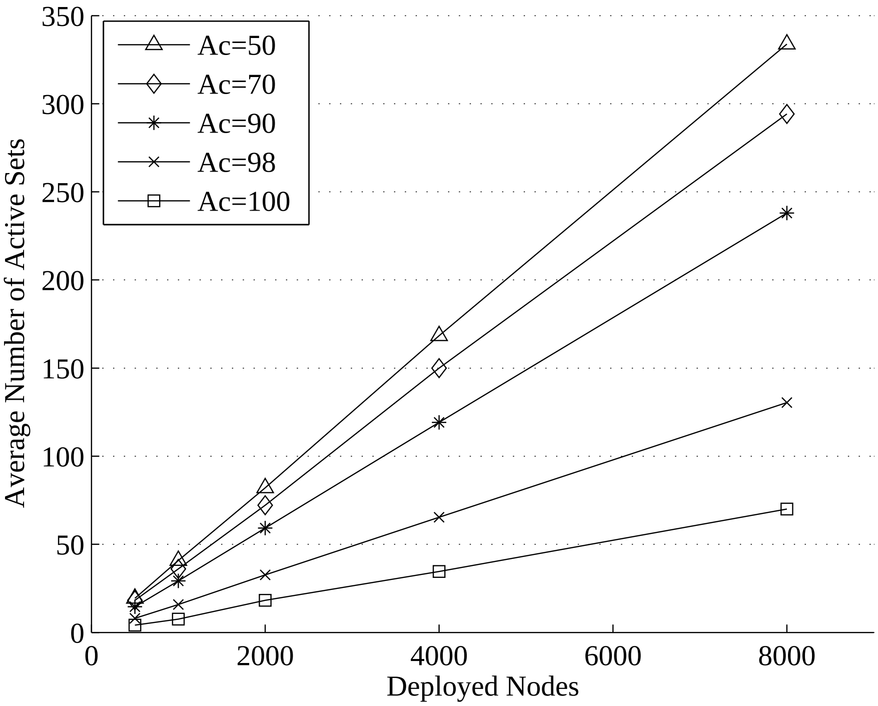


Figure 5.5: Number of Active Sets under Different Acceptable Coverages

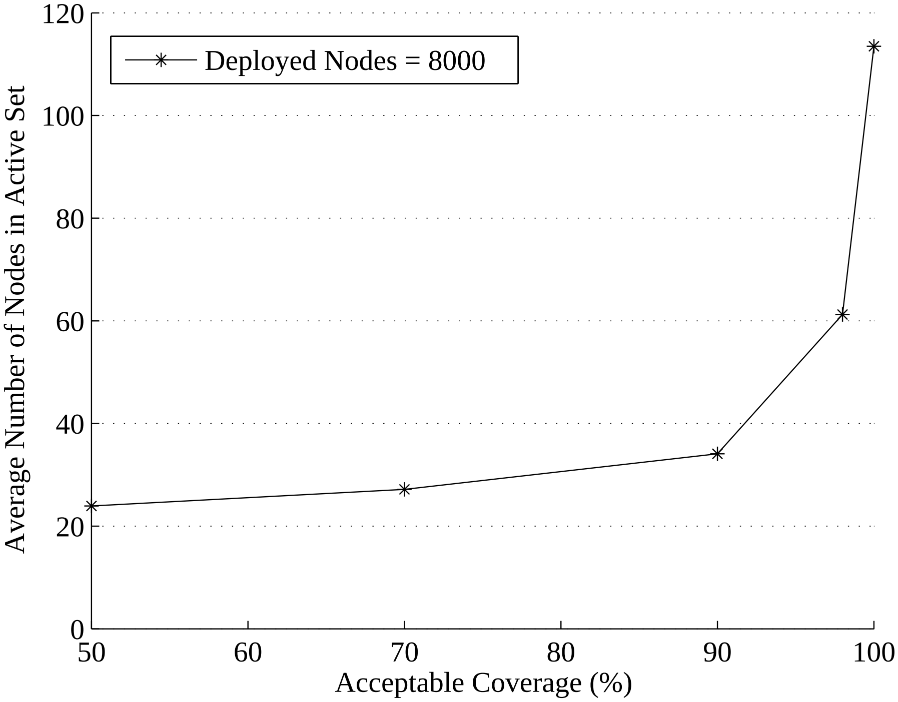


Figure 5.6: Number of Nodes in an Active Set under Different Acceptable Coverages

nodes with different densities. Sensing range is fixed and transmission range has the value that varies between 1 and 3 times the sensing range. Initially, increasing the transmission range coefficient results in an increase on the number of active sets determined. But after the coefficient gets bigger than 2, the number of active node sets is stabilized and does not change with an increase in transmission range. Actually, by increasing the transmission range, we are ensuring that the nodes in the network can communicate with much more sensor nodes in the network. At the beginning, the increase in the transmission range enables us to determine more active node sets. However, after a particular TR/SR ratio, we cannot find more active node sets. This is because, the density of deployed nodes and the sensing range value of a node becomes insufficient and limit us to acquire more active node sets.

In Figure 5.8, the results of the conducted experiment, which illustrates how the number of active sets changes on different network scales, is given. Here, while the size of the monitored area increases, the density of deployed nodes in each 50x50m grid is hold fixed. From the figure, it can be seen that we have roughly same number of active node sets under networks of different scales while the deployed node density is fixed.

We also evaluate the scheme under uniform and nonuniform distributions of sensor nodes. In Figure 5.9, we see the results of simulations that we performed on different network topologies. There are more number of active node sets in uniform distribution compared to nonuniform case. Smaller the deviation in gaussian distribution, more nodes are located around the sink location. Therefore, it becomes difficult to determine active node sets that have prescribed level of sensing coverage of the region. Increasing the gaussian distribution deviation, distribution of the nodes takes more uniform status. Hence, the number of active node sets obtained also increases and approaches the uniform distribution values.

Subsequently, we investigate the effects of applying different constraints to partitioning algorithm. The constraints considered are:

- Number of nodes: With this constraint, the number of nodes in subgraphs

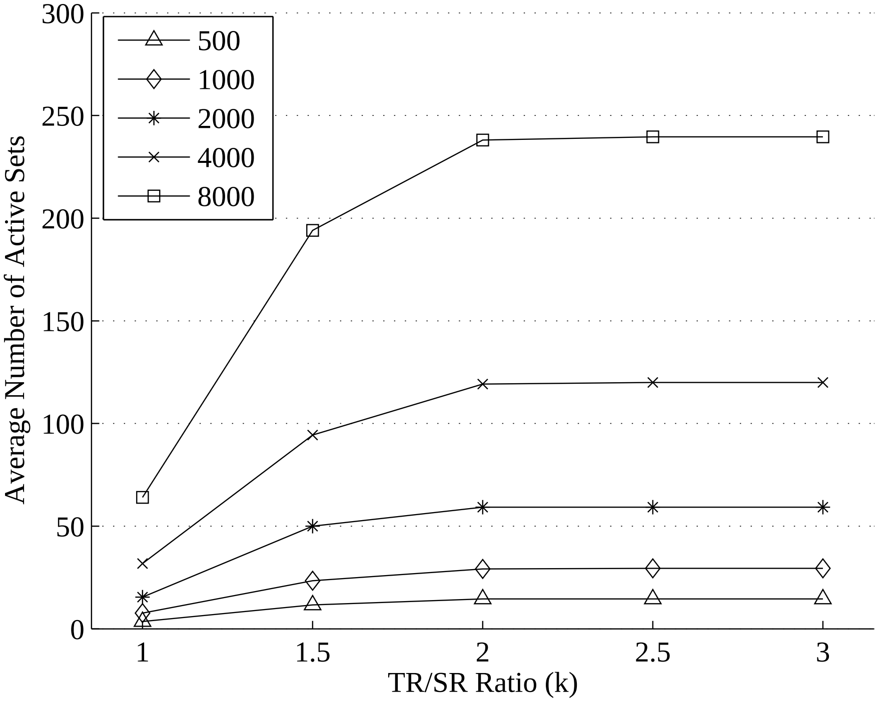


Figure 5.7: Number of Active Sets under Different Transmission Ranges

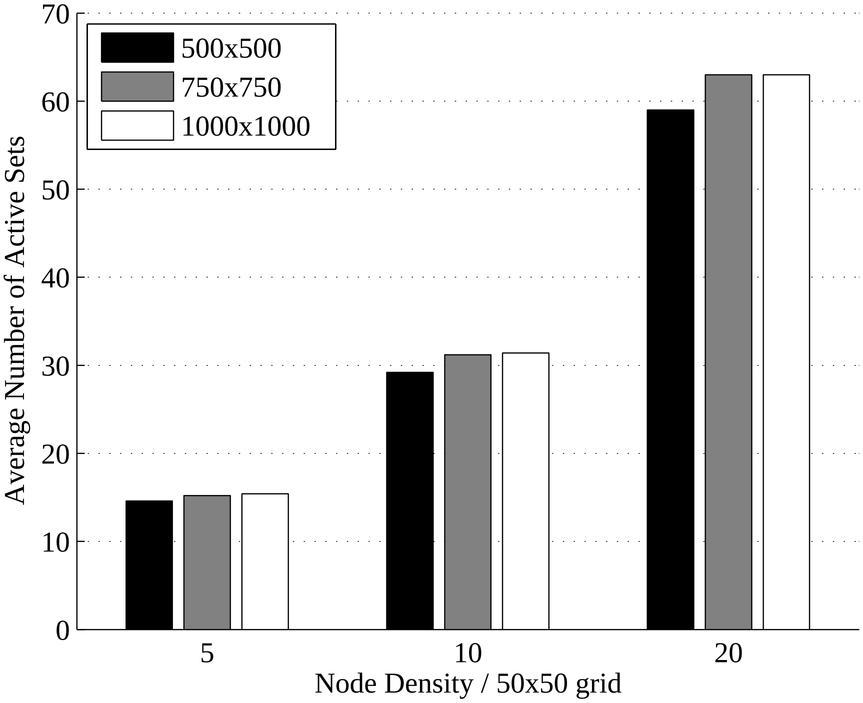


Figure 5.8: Number of Active Sets under Different Network Scales

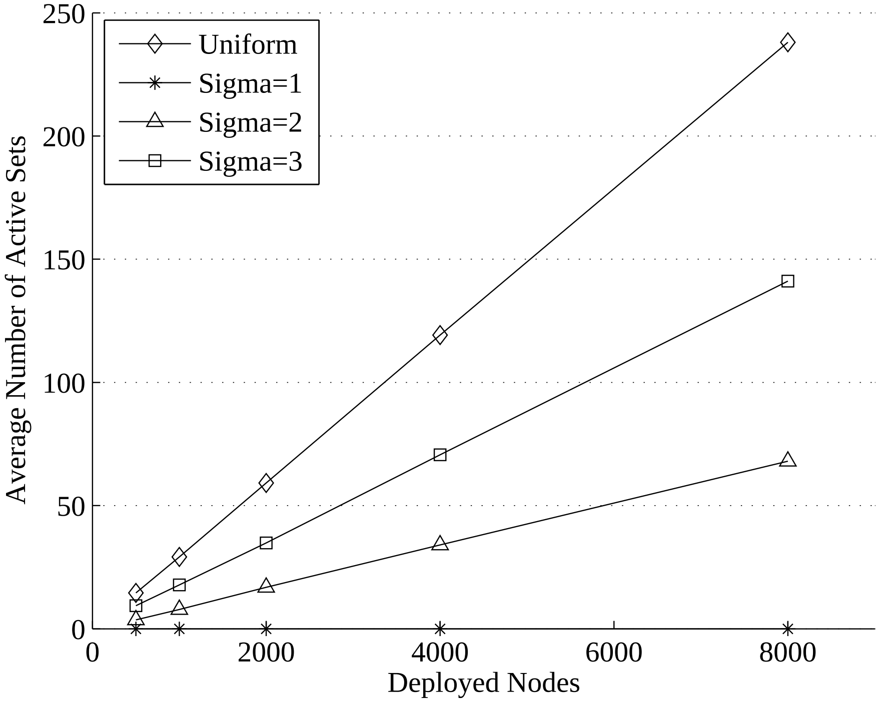


Figure 5.9: Number of Active Sets under Different Network Topologies

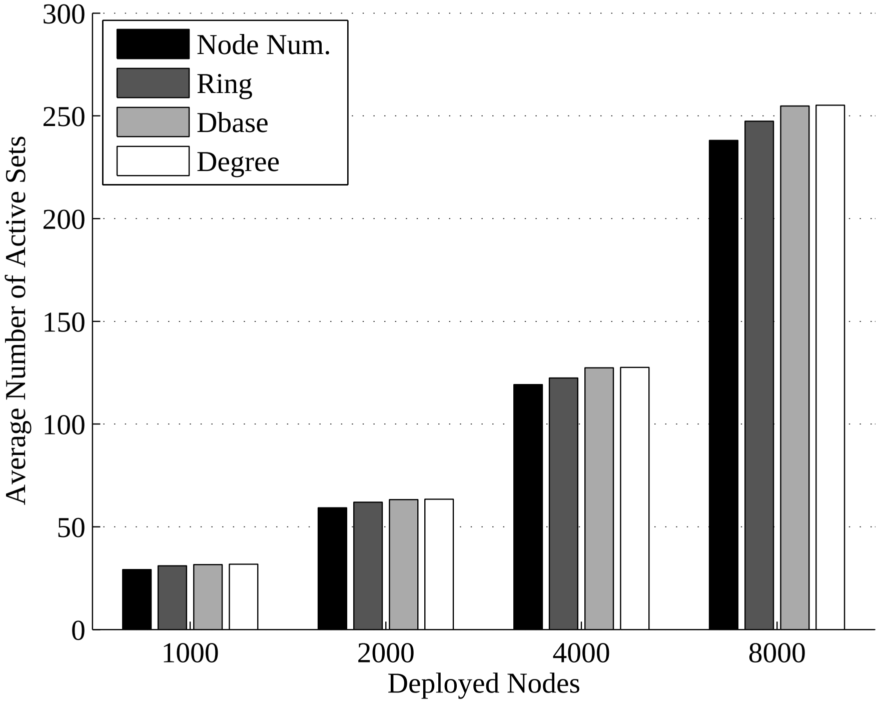


Figure 5.10: Number of Active Sets Under Different Balancing Constraints

obtained by partitioning is tried to balanced.

- Degree: We will determine each sensor node's degree as a constaint, so that nodes with varying degrees is equally distributed between subgraphs.
- Ring: Starting from the location of sink node, we consider rings with increasing radius that is proportional to the transmission range of a sensor node. In the partitioning of a graph, we try to equalize the number of nodes in each ring to provide better subgraphs.
- Distance to base station: Each sensor node is considered to have a weight that is equal to the distance to base station. By this way, closer and further nodes in the network is thought to be equivalently divided between sensor nodes.
- Residual energy: Throughout the network lifetime, sensors have varying degrees of remaining energy levels. We can consider to balance remaining energy levels of active node sets.

Figure 5.10 shows the number of active node sets with different balancing constraints applied under deployment of different number of nodes. Number of active node sets increases with an increase in the number of deployed nodes. We can observe from the figure that balancing degree constraint gives slightly better results compared to other considered constraints.

In the following experiments, we consider virtual grids as in the case of [19]. The area is divided into $50 \times 50m^2$ grid cells. Then, we set the transmission range of a sensor node to be $50\sqrt{5}$ which ensures the communication of sensor nodes in neighbour grids. Also, we determine the sensing range as the half of the given transmission range.

We then perform simulations for network lifetime comparison of applying residual energy as a balancing constraint with the best of the constraints applied. In order to investigate the residual energy constraint, we perform active node set partitioning periodically on the network, according to existing remaining energy values of sensor nodes. After active node sets are determined, each active node set

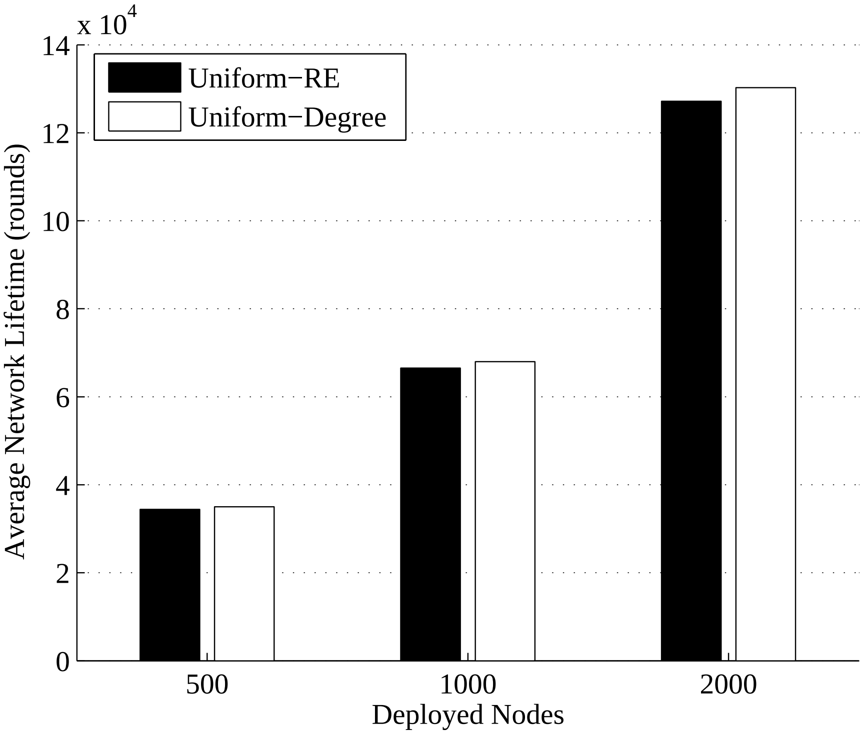


Figure 5.11: Network Lifetime Comparison of Degree and RE Constraints (Uniform Distribution)

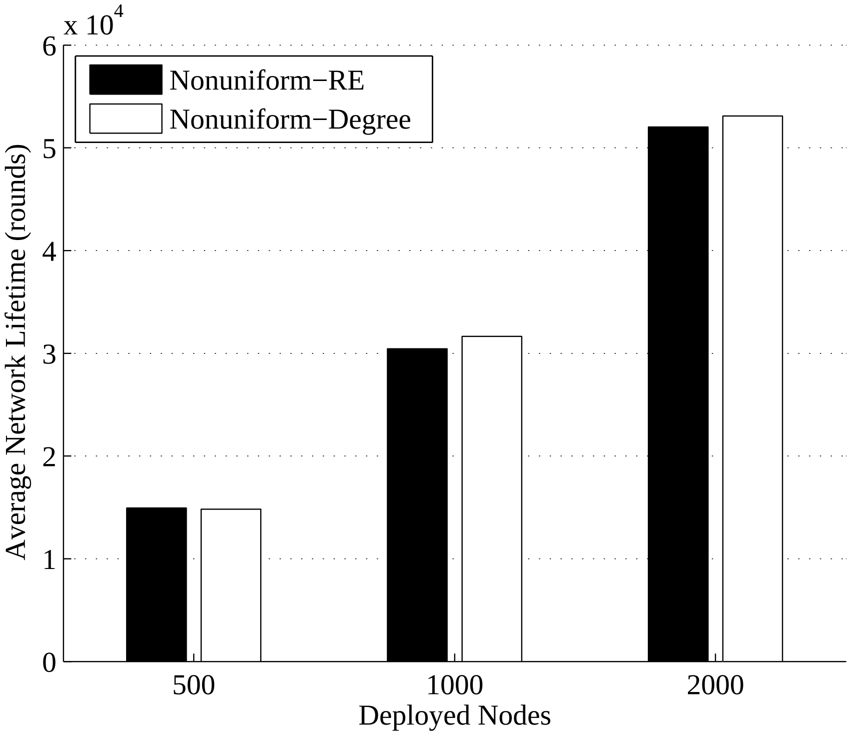


Figure 5.12: Network Lifetime Comparison of Degree and RE Constraints (Gaussian Distribution with SD=2)

senses the region for 100 rounds. In this experiment, the network is considered to be alive, if we can obtain active node sets having above 90% sensing coverage of the region. In addition, we assume an energy consumption model in which each sensor node spends equal amount of energy for both transmitting and receiving the same amount of data. Sensor nodes assumed to make data aggregation and we neglect the energy consumption induced by data aggregation. In the experiments, we also assume a routing tree is constructed in the network. Let E be the amount of energy needed to send or receive one data packet between sensor nodes and N be the number of neighbour nodes (child nodes) that a sensor node get packets. Then, in each round of data transmission phase, total amount of energy spent by a sensor node is equal to $E \times (N + 1)$. If a sensor node senses the region for R rounds then total energy spent by a sensor node while it is in active mode, is $R \times E \times (N + 1)$. Transmission and receiving of a data packet(E) is assumed to consume 1 units of energy where the initial energy of a sensor node is determined as 10.000 units. We have compared the considered constraints under both uniform and nonuniform distributions. In gaussian distribution the standard deviation is set to 2. The results of the comparisons can be seen in Figures 5.11 and 5.12. According to the results, applying the degree constraint on the active set partitioning scheme, performs better and has longer network lifetime than applying the RE constraint. This behaviour is counter-intuitive, i.e. we expect to see a better effect on the network lifetime by balancing the residual energies between active node sets. These final results might be attributed to the possibility that obtaining active node sets with equally distributed RE weights have nearly the same consequence with obtaining active node sets with equally distributed degrees. This is because, sensor nodes having higher number of child nodes means sensor nodes spending higher energy values for sensing and communication tasks.

We evaluate different selection methods and present the comparison of these methods with respect to average number of active node sets. We assume that the network lifetime is proportional to the number of active sets in these experiments. There are two selection mechanisms that we have compared our approach. The first method is GAF-like selection. In this approach, an active node set is constructed from the sensor nodes that are selected from each virtual $50 \times 50m^2$

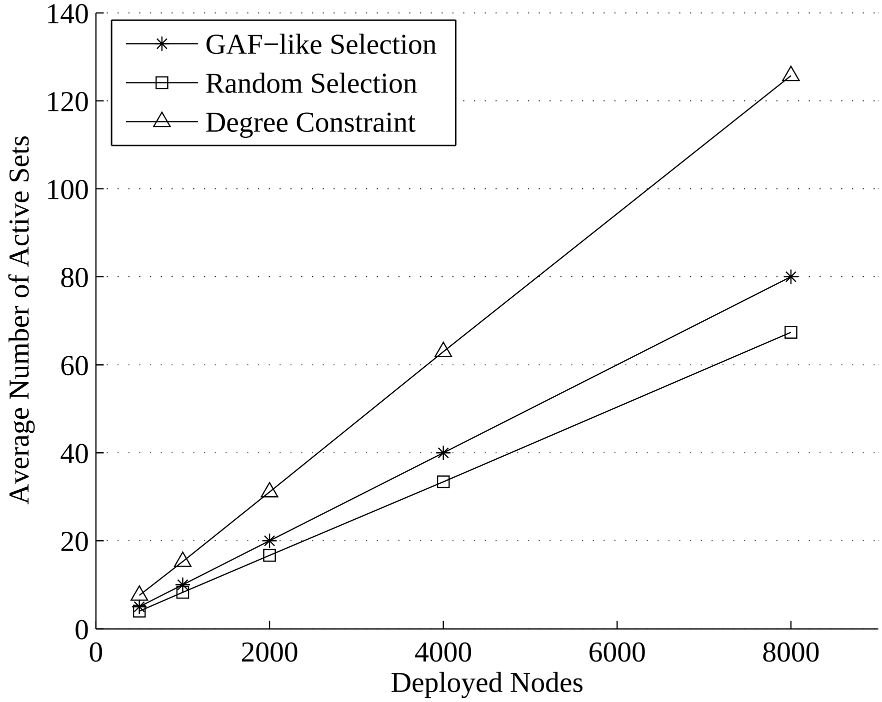


Figure 5.13: Comparisons of Different Active Node Selection Methods

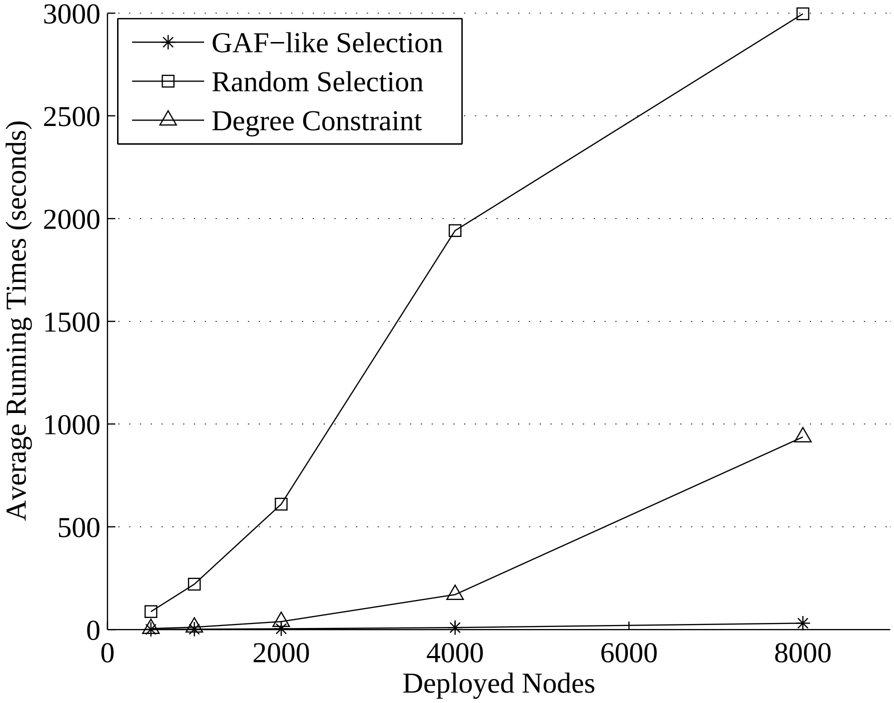


Figure 5.14: Different Active Node Selection Methods Running Times

grid. The second method is a naive scheme, in which we repeatedly select random sensor nodes unless the selected nodes preserve desired level of sensing coverage and connectivity. In Figure 5.13, we see the comparison of these different selection mechanisms under nearly complete coverage of the region($Cov_{req} = 98\%$). It can be seen in the figure that the scheme proposed gives better results than other selection mechanisms with respect to the number of active node sets. In Figure 5.14, the average running times of these different active set selection mechanisms is given. It is observed that the running time of GAF-like selection scheme lies nearly on the x-axis and has the smallest running time overhead from the compared mechanisms. Random selection mechanism, having average number of active set results similar to GAF-like selection mechanism, has the greatest running time because of checking the desired objectives at each selection step. When we have taken into consideration the average number of active sets obtained and the running time overheads, ASP algorithm has better results of active node sets with an acceptable and short running time overhead compared to other mechanisms.

Next, we performed experiments under different acceptable coverages and deployment of sensor nodes in the region. Here, we consider uniform distribution as well as gaussian distribution with standard deviation 2 and 3. In addition, acceptable sensing coverage levels 70% and 90% is also taken into consideration. In Figures 5.15 and 5.16, we see the results of the experiments under uniform distribution. It is observed that decreasing the acceptable sensing coverage, results in an increase in the number of obtained active node sets. In both experiments, the scheme proposed with degree constraint performs better than naive scheme. Furthermore, the results of the experiments performed under gaussian distribution with SD=3 is given in Figures 5.17 and 5.18. In these experiments, we also see that the average number of active sets obtained by ASP is greater than obtained by the naive scheme. However, the average number of active sets obtained by naive scheme becomes worse when compared with the results under uniform distribution. Finally, in Figures 5.19 and 5.20, simulation results under gaussian distribution with SD=2 is given. Average number of active node sets that are obtained in these experiments has lower values compared with the results under gaussian distribution with SD=3. This is because, decrease in the standard deviation of gaussian distribution increases the nonuniform distribution of nodes around the sink node which results in a decrease in the number of active node sets ensuring the required coverage and connectivity.

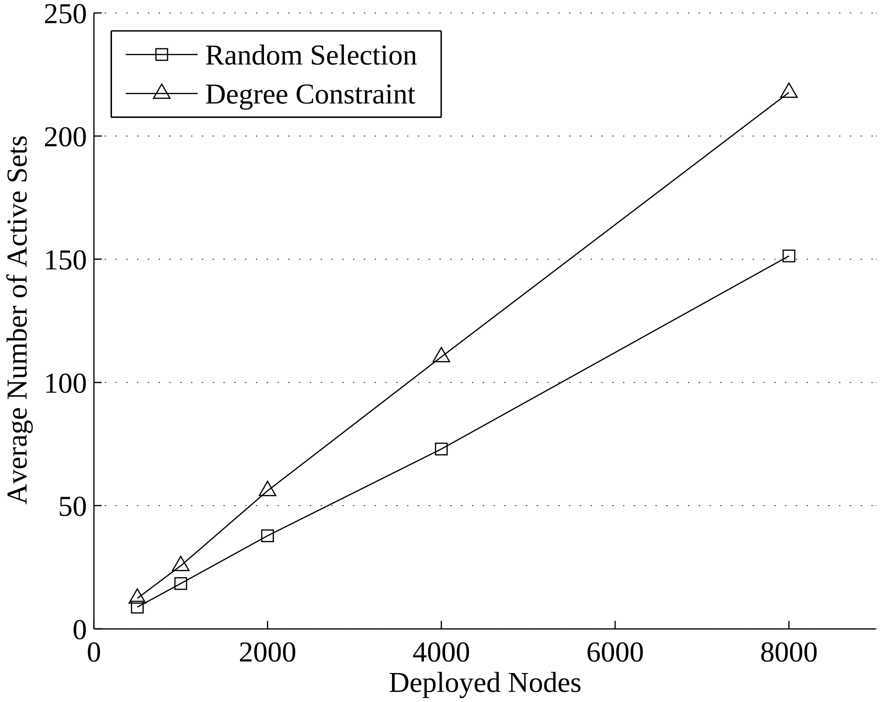


Figure 5.15: Uniform Distribution, $Cov_{req} = 70\%$

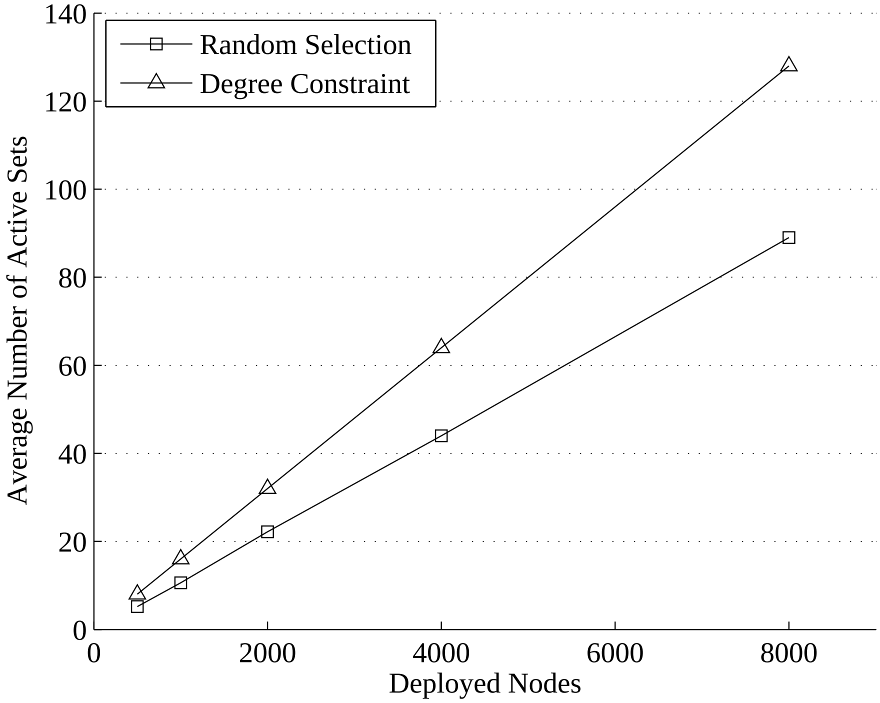


Figure 5.16: Uniform Distribution, $Cov_{req} = 90\%$

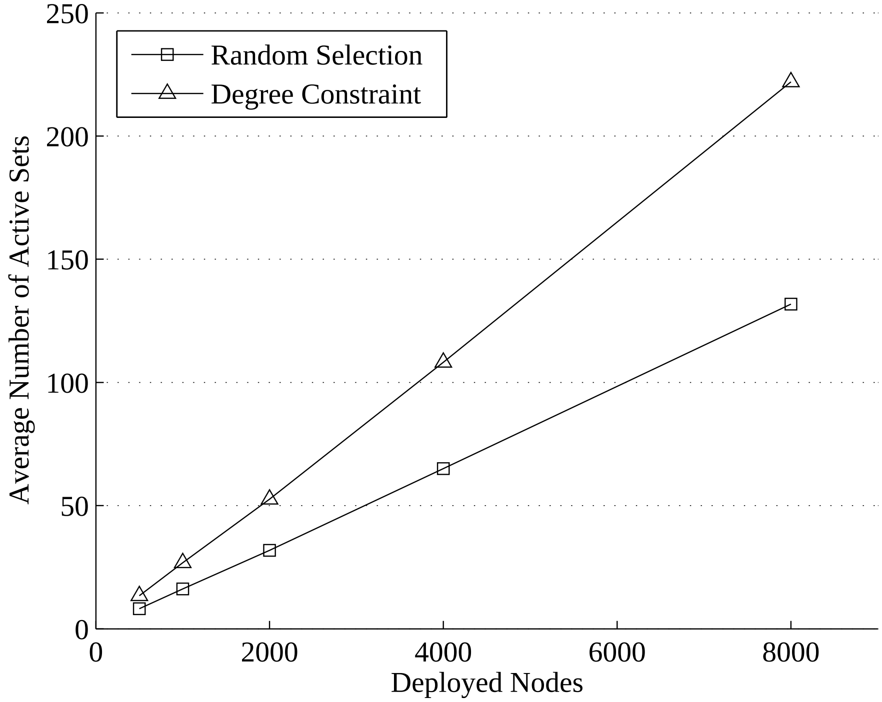


Figure 5.17: Gaussian Distribution with SD=3, $Cov_{req} = 70\%$

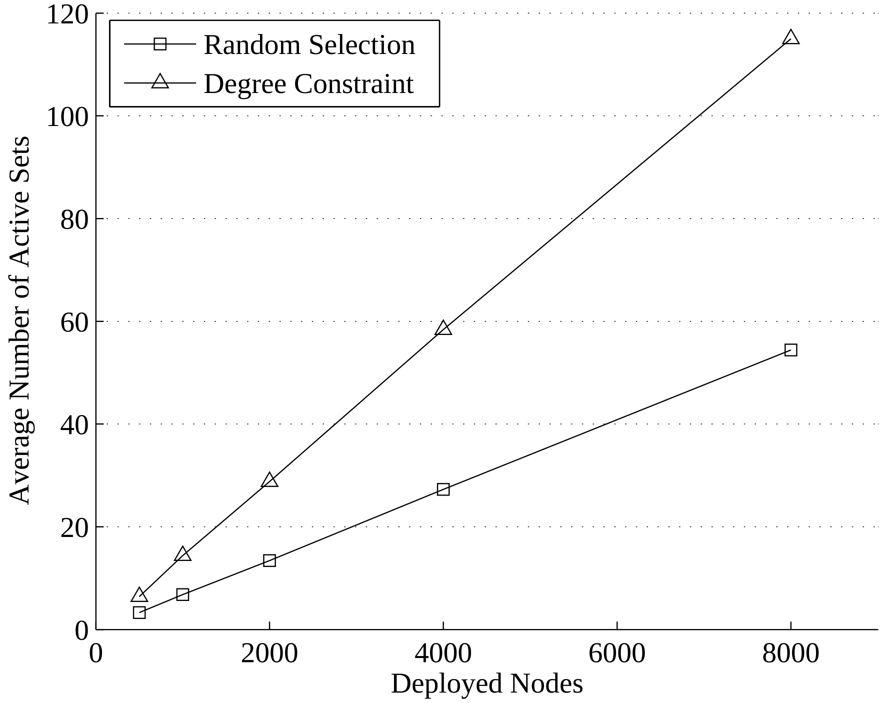


Figure 5.18: Gaussian Distribution with SD=3, $Cov_{req} = 90\%$

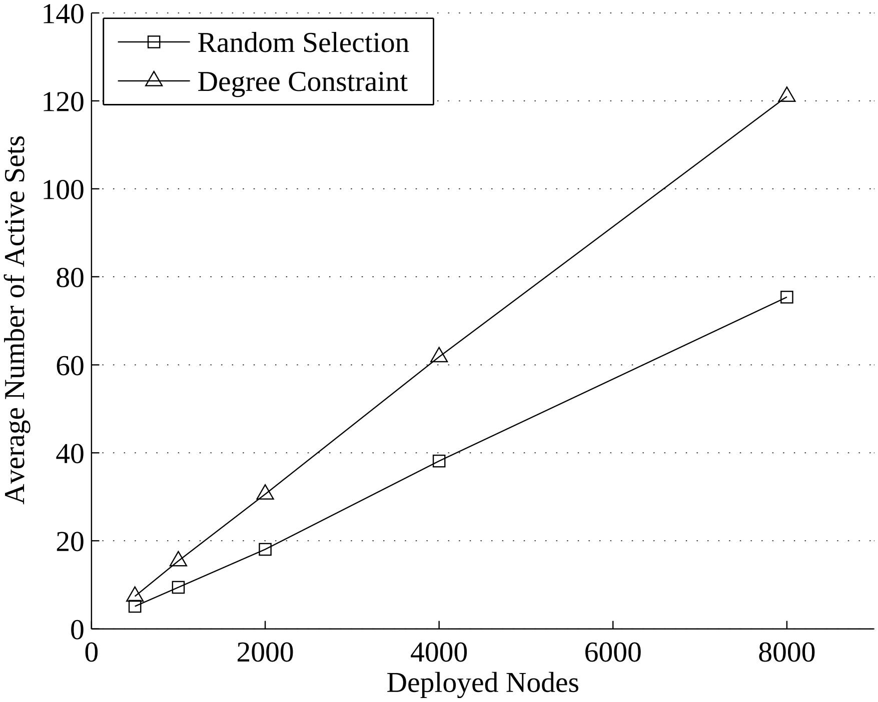


Figure 5.19: Gaussian Distribution with SD=2, $Cov_{req} = 70\%$

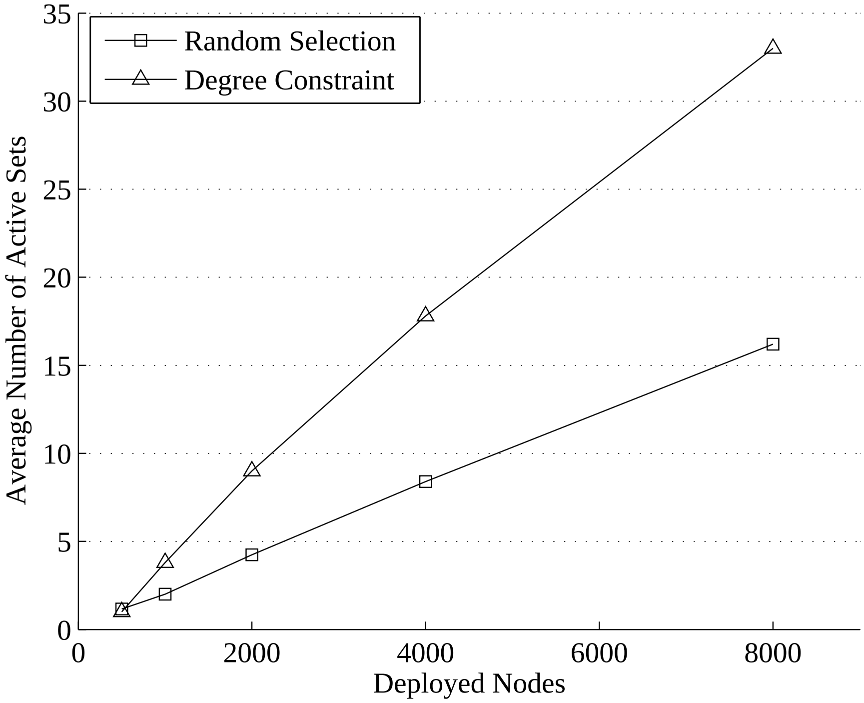


Figure 5.20: Gaussian Distribution with SD=2, $Cov_{req} = 90\%$

Chapter 6

Conclusion

In high density wireless sensor networks, an effective way to conserve energy is to leave only a subset of nodes in active mode to perform sensing and communication tasks. By this way, we can reduce the energy consumption and extend the network lifetime.

In this paper, we have investigated the problem of finding connected subsets of nodes that satisfy a certain level of sensing coverage of the region. In order to solve the problem, we represent the sensor network as a graph and propose a scheme called ASP to partition the input graph into subgraphs that meet the requirements needed. Essentially, the proposed scheme can be considered as applying declustering on the graph considering that closer nodes in the network should be in different active node set. The ASP algorithm gives us the flexibility of adjusting the required sensing coverage of the region, considering that for an application it might be sufficient to maintain a certain level of sensing coverage.

Our scheme is evaluated under different network topologies and parameters through extensive simulations and is compared with different alternatives. In addition, to further reduce the execution time of the algorithm while maintaining the necessary objectives, decreasing the size of the input graph is considered. The simulation results show that ASP algorithm performs approximately 50% better than evaluated alternatives in terms of number of active sets obtained.

Bibliography

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks (Amsterdam, Netherlands: 1999)*, 38(4):393–422, Mar. 2002.

[2] I. F. Akyildiz, S. Weilian, Y. Sankarasubramaniam, and E. E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, 2002.

[3] A. Cerpa and D. Estrin. Ascent: Adaptive self-configuring sensor networks topologies, 2004.

[4] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In *ACM Wireless Networks Journal*, pages 85–96, 2001.

[5] C.-Y. Chong and S. Kumar. Sensor networks: evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91(8):1247–1256, Aug. 2003.

[6] H. Gupta, S. R. Das, and Q. Gu. Connected sensor cover: Self-organization of sensor networks for efficient query execution. pages 189–200, 2003.

[7] B. Hendrickson and R. Leland. A multilevel algorithm for partitioning graphs. Technical Report SAND93-1301, Sandia National Laboratories, 1993. appeared in Proc. Supercomputing 95.

[8] G. Karypis and V. Kumar. *Metis A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices*, 1995.

[9] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs, 1998.

[10] G. Karypis and V. Kumar. Multilevel algorithms for multi-constraint graph partitioning. In *Supercomputing '98: Proceedings of the 1998 ACM/IEEE conference on Supercomputing (CDROM)*, pages 1–13, Washington, DC, USA, 1998. IEEE Computer Society.

[11] G. Karypis, V. Kumar, and V. Kumar. Multilevel k-way partitioning scheme for irregular graphs. *Journal of Parallel and Distributed Computing*, 48:96–129, 1998.

[12] A. M. Mainwaring, D. E. Culler, J. Polastre, R. Szewczyk, and J. Anderson. Wireless sensor networks for habitat monitoring. In C. S. Raghavendra and K. M. Sivalingam, editors, *WSNA*, pages 88–97. ACM, 2002.

[13] S. Slijepcevic and M. Potkonjak. Power efficient organization of wireless sensor networks. pages 472–476, 2001.

[14] M. T. Thai, F. Wang, D. H. Du, and X. Jia. Coverage problems in wireless sensor networks: designs and analysis. *Int. J. Sen. Netw.*, 3(3):191–200, 2008.

[15] D. Tian and N. Georganas. A coverage-preserving node scheduling scheme for large wireless sensor networks, 2002.

[16] L. Wang and Y. Xiao. A survey of energy-efficient scheduling mechanisms in sensor networks. *MONET*, 11(5):723–740, 2006.

[17] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In *Proceedings of the first international conference on Embedded networked sensor systems (SenSys-03)*, pages 28–39, New York, Nov. 5–7 2003. ACM Press.

[18] J. Wu. *Handbook on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks.* Auerbach Publications, Newyork, 2006.

[19] Y. Xu, J. S. Heidemann, and D. Estrin. Geography-informed energy conservation for ad hoc routing. In *MOBICOM*, pages 70–84, 2001.

[20] F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang. Peas: A robust energy conserving protocol for long-lived sensor networks. pages 28–37, 2003.

[21] H. Zhang and J. C. Hou. Maintaining sensing coverage and connectivity in large sensor networks. *Wireless Ad Hoc & Sensor Networks*, 1(1-2):89–123, 2005.

[22] F. Zhao and L. J. Guibas. *Wireless sensor networks : an information processing approach*. Morgan Kaufmann, Amsterdam ; San Francisco, 2004.