Domination Algorithms for Lifetime Problems in Self-organizing Ad hoc and Sensor Networks

Thesis submitted to the
Indian Institute of Technology, Kharagpur
for award of the degree
of

Doctor of Philosophy (Ph.D.)

by

Rajiv Misra

School of Information Technology
Indian Institute of Technology, Kharagpur
West Bengal 721302, INDIA

May, 2008

To my father for his encouragement.

To my mother for her love.

To my wife for being there.

To my children for their future.

Indian Institute of Technology
Kharagpur 721302

Declaration

I hereby declare that the thesis entitled **Domination Algorithms for Lifetime Problems in Self-organizing Ad hoc and Sensor Networks** being submitted by the undersigned, **Rajiv Misra**, a research scholar in the School of Information Technology, Indian Institute of Technology, Kharagpur, for the award of the degree of **Doctor of Philosophy** represents original work and has not been submitted to any other University or Institute for award of any degree or diploma.

Dated : 18 May, 2008
Kharagpur 721302

Rajiv Misra, Research Scholar
School of Information Technology
Indian Institute of Technology, Kharagpur
W. B. 721302, INDIA.

Indian Institute of Technology
Kharagpur 721302

Certificate

This is to certify that the thesis entitled **Domination Algorithms for Lifetime Problems in Self-organizing Ad hoc and Sensor Networks** being submitted by **Rajiv Misra**, a research scholar in the School of Information Technology, Indian Institute of Technology, Kharagpur, for the award of the degree of **Doctor of Philosophy** is an original research work carried out by him under my supervision and guidance. The thesis has fulfilled all the requirements as per the regulations of this institute and, in my opinion, has reached the standard needed for submission.

Dated : 18 May, 2008
Kharagpur 721302

Chittaranjan Mandal, Associate Professor
Department of Computer Sc. & Engg.
and School of Information Technology
Indian Institute of Technology, Kharagpur
W. B. 721302, INDIA.

Acknowledgment

It has been almost four and half years since I started my Ph.D. education at IIT Kharagpur, and when I sit back and think about the number of people who influenced me and helped me complete this thesis, I am overwhelmed! There is no doubt that this would have been impossible without their help. I hope that I can remember everyone who helped me through this difficult yet rewarding process. First and foremost, I would like to thank my advisor, Professor Chittaranjan Mandal, for taking me on as a student about four and half years ago, even though he knew nothing about me at the time. It was an extraordinary piece of good fortune that led to my becoming his student. He has been an ideal advisor in every respect, both in terms of technical advice on my research and in terms of other advice. My choice of research career has been greatly influenced by Chitta and I hope that I can live up to his standards. I look forward to continue working with him and further developing our friendship. I thank Professors Indranil Sen Gupta (Head), Saswat Chakraborty, Aurobinda Gupta and Sujoy Ghosh for serving on my doctoral scrutiny committees. Also I thank Professor S.K.Ghosh PhD coordinator for his valuable support at every moment. I thank every professor for taking a course with them and gain knowledge and gaining a style of teaching from them. My research was supported by the Institute Scholarship. I also thank Professor A. K. Majumdar for providing financial support from his project to continue support after the fourth year of my study. Without all this support I would not have done this work. I thank School of IT for believing in me and giving me the support to come to this IIT Kharagpur to finish this work. I thank all the scholars that we had in the lab for the discussion and that we used to have and share ideas. And special thanks to Soumya, Ashalata for all the good times that we spent in the lab and the good discussion that we used to have sometime we agree and some other time we disagree. I would like to thank all my friends who supported me and thank them for their encouragement to finish this work. I owe a special debt of gratitude to my parents and family. They have, more than anyone else, been the reason I have been able to get this far. Words cannot express my gratitude to my parents, who give me their support and love from across the regions. My wife, Smriti, gives me her selfless support that make me want to excel. I am grateful to her for enriching my life....

Rajiv Misra

Abstract

Wireless sensor networks propound an algorithmic research problems for prolonging life of nodes and network. The domination algorithms can address some of fundamental issues related to lifetime problems in ad hoc and sensor networks. Most of the graph domination problems are *NP-complete* even with unit-disk-graphs. The investigation of the thesis addresses some of lifetime issues in sensor network with the approximate domination algorithm.

In this work, we consider *distributed* algorithms of some important domination problems namely, maximum domatic partition problem (DPP), maximum connected domatic partition (CDP) problem, minimum connected dominating set (MCDS) problem, node-mobility transparent connected dominating set problem in context of unit-disk graphs and obtain solutions using state-of-the-art principles of well-known MIS (maximal independent sets). We incorporated self-organization feature to domatic partition for sensor networks. Domatic partition problems has variety of applications. In sensor networks our deterministic self-organizing domatic partition algorithm is used to provide maximum cluster lifetime in hierarchical topology control of sensor networks. Minimum connected dominating set is reported to provide a virtual backbone for ad hoc networks. The maximum lifetime of connected dominating set felt constrained to support virtual backbone in sensor networks. We modeled the maximum lifetime connected dominating set as connected domatic partition problem. We introduced a distributed algorithm for connected domatic partition problem. To our knowledge no such connected domatic partition is reported in literature.

The minimum connected dominating set has drawn a considerable research interest and several approximation schemes are reported. We have introduced a collaborative-cover heuristic and developed a distributed approximation algorithm for minimum connected dominating set problem using it with a single leader having an approximation factor of $(4.8 + \ln 5)\mathrm{opt} + 1.2$, where opt is the size of any optimal CDS in G. This approximation provides an effective loss-less aggregation backbone for sensor networks. The results show the improvement in prolonging the life of sensor networks. The CDS-backbone gets disturbed by the mobility of nodes. We developed an integrated scheme adapting CDS to the node's mobility transparently and efficiently. Adapting CDS to node-mobility is carried out by using four steps: $i)$ reinforcing a self-organization to a multi-protocol relay(MPR) based connected dominating set, $ii)$ reinforcing self-reconfiguration of CDS when a node becomes mobile or halts after mobile operation, $iii)$ adapting CDS to mobile-node by tracking of mobile node for its location updates and $iv)$ optimizing location updates using weighted CDS based on a Markov model.

Keywords*: Ad hoc networks, clusterhead rotation, Connected Dominating Set, Connected Domatic Partition, node mobility*

Contents

List of Figures

List of Algorithms

Chapter 1

Introduction

Recent advances in VLSI, MEMS and other technologies have led to the growth of tiny, cheap and low power wireless sensor nodes equipped with three main units: radio frequency (RF) transceiver, processor and a sensor unit, which is capable of sensing, computing and communicating by wireless. The battery powered sensor nodes are often deployed in remote geographic locations and their energy source cannot be replenished. Newer applications for surveillance, environmental control and defence are possible by deploying a large number of sensor nodes in the target area and processing the information gathered from them. A wireless network of sensor nodes (WSN) is capable of sensing information of the environment, such as temperature, pressure, humidity, illumination, etc. The network is also capable of compressing, filtering and analyzing the data to some extent. The gathered and processed information is usually communicated to one or more base stations. Nodes route data through intermediate nodes destined eventually for the base station. Thus, the nodes act as routers in addition to sensing. Nodes can directly communicate with nodes within their maximum transmission range. Unit disk graphs (UDG) are intersection graphs of nodes with equal transmission ranges and provide a graph theoretic model for developing algorithms for WSNs.

While conventional networks aim to achieve high quality of service provisioning or high bandwidth, sensor networks protocols must focus primarily on efficiency of communication with an eye on power conservation. For the design of WSN protocols, this tradeoff opens up the option of prolonging operation lifetime at the cost of lower throughput or higher density of node deployment. Network lifetime in sensor networks is referred to as the time elapsed until the first node (or last node) in the network depletes its energy completely. In applications, where all the nodes are critical, lifetime refers to the time when the first node dies.

Many researchers have looked at extending the lifetime of a wireless system through the use of more efficient hardware. However, use of energy efficient or power aware

protocols is a relatively new concept emerging in wireless networking. Until recently, most of the clustering techniques concentrated on hierarchically organizing sensor networks for remote data gathering application. In clustering protocols, clusterhead nodes are loaded with more computational and communication load than non-clusterhead nodes[2, 3, 4]. Clustering protocols in sensor networks aim to exploit in-network data aggregation in reducing number of communication having the tradeoff of reduced quality of solution. Protocol designers then realized the need for load balancing to distribute the computational overheads of aggregating points or clusterheads across the network nodes to save early exhaustion of nodes. Energy consumption of a sensor node can be broadly classified as useful or wasteful. By useful energy consumption, we mean node consuming energy in transmitting or receiving data, local computations and forwarding data to neighboring nodes. Examples of wasteful energy consumption are the overheads due to idle listening, retransmitting, load balancing and generating control packets. Due to high cost of communication and limited energy, it is natural to seek decentralized, distributed algorithms for wireless sensor networks which can prolong network lifetime.

WSNs are ad hoc in nature, having no physical infrastructure for support of network services such as routing, broadcasting, in-network aggregation and connectivity management. A virtual backbone can be formed to support such services. Nodes working on the virtual backbone suffer from early energy exhaustion. Large scale deployment of sensor nodes in wireless sensor networks needs an efficient organization of network topology for reducing communication and prolonging life of network. Hierarchical topology control employs load balancing to rotate the role of clusterhead operation across the network nodes to prolong the life of nodes and network.

The rest of the chapter is organized as follows. Section 1.1 presents a compact survey of the literature and brings out the motivation of the work presented in this thesis. Individual chapters contain additional survey that is specific to the problem handled there. Section 1.2 presents an overview of the thesis work and summarises the contributions made. Section 1.3 describes the organization of the thesis.

1.1 Literature Survey and Motivation

In this section, we present a brief survey of literature on the topics of interest to the thesis. The scope of survey is divided into the following areas in bringing out the motivation of the thesis work: hierarchical topology control of sensor networks, domatic partition problems in sensor networks, minimum connected dominating set (CDS) problem and self reorganization of connected dominating sets in sensor networks. This survey provides the motivation of the problems that have been worked on in the thesis.

Clustering for Hierarchical Topology Control of Sensor Networks Clustering techniques can be divided as centralized or distributed, based on whether network wide information or local information is collected to decide the optimal hierarchical topology control. We present the review of a few distributed clustering schemes to reveal the important issues such as re-clustering.

LEACH (Low Energy Adaptive Clustering Hierarchy) [3, 5] introduced the technique of randomly rotating the role of the clusterhead among all the nodes for equal distribution of high energy load. LEACH provides significant energy savings, prolonged network lifetime by applying localized algorithms and data aggregation within randomly self elected cluster heads. The main drawback of LEACH is the periodic re-clustering to elect a new set of clusterheads. Thus, re-clustering has a substantial wasteful energy overhead.

HEED (Hybrid Energy-Efficient Distributed Clustering) is another protocol to prolong network lifetime, using clustering [2] but, using a hybrid approach: clusterheads are randomly selected based on their residual energy and nodes join clusters such that communication cost is minimized. Like LEACH, HEED also involves the periodic re-clustering to elect a new set of clusterheads. Thus, it also suffers from a substantial wasteful energy overhead.

Recently a distributed minimum cost clustering protocol (MCCP) [6] based on cluster centric cost heuristic has been shown to improve network lifetime as compared to the HEED protocol.

In a fixed clustering scheme LEACH-F [5], the clusters identified in the initial round becomes fixed. For load balancing, the clusterhead rotates locally within its fixed clusters. Thus, the fixed clustering scheme results in less energy overhead due to the rotation of clusterheads locally compared to adaptive clustering schemes such as LEACH and HEED. However, fixed clustering results in a major drawback of higher overhead in communication energy due to skewed inter cluster and intra-cluster distances. The advantages of adaptive clustering and fixed clustering motivates the need of an efficient load balancing scheme for clustering protocols which should be rotating the roles of clusterhead with the balanced inter cluster and intra-cluster communication distances.

Domatic Partition Problems in Sensor Networks For a given graph $G = (V, E)$, the domatic partition of V is a partition of V into dominating sets. The domatic number $D(G)$ of G is the size of the largest domatic partition. Note that $D(G) \leq \delta + 1$, where $\delta = \delta(G)$ denotes the minimum degree of G. A graph is said to be domatically full if its domatic number $D(G) = \delta + 1$ (i.e. the maximum domatic number). Finding a maximum sized domatic partition is NP-Complete. Feige[7] reported the first non-trivial approximation algorithm for the domatic partition problem that guarantees

the largest approximation factor $\frac{1}{O(\lg \Delta)}$, where Δ denotes the maximum degree of a node in G. The problem of finding the maximum number of disjoint dominating sets is modeled as the domatic partitioning of a network graph[7, 8, 9, 10]. Three distributed algorithms for finding large k-domatic partition ($k > 1$) for different graph models are reported in [9]. An $O(1)$ round k-domatic partition algorithm is reported in [9] for unit ball graphs (UBG) in Euclidean space where all nodes know their own locations. For UBGs, the k-domatic partition algorithm gives $O(\log^* n)$ time on metric space with constant doubling dimensions and when only pairwise distances between neighboring nodes are known. Finally, for growth bounded graphs using only connectivity information, the k-domatic partition algorithm gives $O(\log \Delta \log^* n)$ time. None of the reported domatic partition schemes consider self organisation aspects, which is required in sensor networks. In the thesis, we consider aspects of self organisation in the domatic partition problem. Dominating sets of domatic partitions in sensor networks often need to be connected. For applications of connected dominating sets, the related problem becomes connected domatic partitioning (CDP). There is only limited coverage of CDP in the literature. This has motivated us to work on the CDP problem.

Minimum Connected Dominating Set Problem in Sensor Networks The possibility of using a CDS as a virtual backbone was first proposed in 1987 by Ephermides[11]. Since, then many algorithms to construct CDS have been reported which can be classified into following four categories based on construction: *i)* centralized algorithms, *ii)* distributed algorithms using single leader, *iii)* distributed algorithms using multiple leaders and *iv)* localized algorithms. The centralized algorithms require network wide global information and hence is not suited for wireless sensor networks which have no centralized control. Due to its large approximation factor, multiple leader based distributed CDS construction is not effective for exploiting lossless in-network aggregation. The localized CDS construction approach, first proposed by Adjih[12], is based on multipoint relays (MPR) but no approximation analysis of that algorithm is known to be reported. Thus, for the problem of lossless aggregation in WSNs, our interest is in works related to distributed algorithms using single leader for the minimum connected dominating set.

Single leader based distributed algorithms for CDS construction[13, 14, 15, 16] assume the availability of an initial leader.The base station is often the initiator or a leader election algorithm is used for the initiator. The distributed algorithm uses the idea of identifying an MIS first and then a set of connectors to connect the MIS is identified to form a CDS. Alzoubi[13] presented an ID based distributed algorithm to construct a CDS tree rooted at leader. This MIS based distributed algorithm for UDGs uses a single initiator to construct a CDS. The approximation factor on the size of the CDS obtained is at most $8\mathrm{opt} + 1$, where opt is the size of any optimal CDS. The time complexity is $O(n)$ and the message complexity is $O(n \log n)$. This algorithm was later

improved by Cardei[14] with approximation of $8\mathrm{opt}$ using degree based heuristics and degree aware optimization for identifying Steiner point as the connectors in CDS construction. The distributed algorithm [14] growing from single leader has $O(n)$ message complexity and $O(\Delta n)$ time complexity using 1-hop neighborhood information. Thus, the problem of minimum connected dominating set with a single leader helps to identify the aggregation backbone in a WSN. The better known approximation guarantees to minimum CDS with a single leader are reported as $8\mathrm{opt} + 1$ [13], $8\mathrm{opt}$ [14] and $(4.8 + \ln 5)\mathrm{opt} + 1.2$ [16].

Node mobility in CDS in Ad hoc and Sensor Networks The CDS backbone gets disturbed mainly due to node failures or node mobility. In this context, we have surveyed some works on self organisation and object tracking in WSNs, which can be classified as: $i)$ mobility profile based tracking[17, 18, 19] and $ii)$ online tracking[20], based on mobility profile history information or online information of mobile node. Online tracking of mobile objects using a hierarchical structure called regional directory service to limit the updates in tracking algorithm was given by Awerbuch and Peleg[20]. This scheme is of interest to us in terms of making location updates while tracking mobile nodes but differs completely with the approach used by our tracking algorithm. Hsing's mobility profile algorithm[19] works independently of mobility history and uses a Markov model based on geometric information to construct the maximum spanning tree for estimating the object crossing rates between sensors. There scheme does not involve mobility of network nodes. This scheme is interest to our work as our scheme also uses Markov chain model but we do not use geometric information. Recently Adjih[12] and Wu[21] reported an approach for small size CDS construction based on multipoint relays. Extended MPR k-hop ($k \leq 3$) local information based small size connected dominating set construction is reported in [21]. The local MPR based CDS scheme is of interest to our work because of its small size and localised construction, can be easily adapted to changes arising due to node mobility. The reported schemes do not consider the adaptability of MPR based CDS construction to node mobility. We have, therefore, worked to develop a scheme for an adaptive MPR based CDS construction for node mobility.

1.2 Overview and Contributions of this Thesis

In this section we first list statements of the problems that have been addressed in this thesis and then give outlines of the methodologies adopted for their solution. We also mention specific contribution made in each case. The problems on wireless sensor networks addressed in the thesis are:

1. Design of a distributed algorithm for self organizing domatic partition problem

2. Design of a distributed algorithm for the maximum connected domatic partition problem

3. Design of a distributed algorithm for the minimum connected dominating set problem for computing the aggregation backbone

4. Design of a node mobility transparent connected dominating set algorithm

A distributed algorithm for self organizing domatic partition problem Aggregation aware clustering algorithms addresses lifetime and scalability goals, but suffers from the twin problems of uncovered coverage area and energy overhead due to clusterhead rotation. Load balancing in existing clustering schemes use global rotation of clusterhead roles in order to prevent any single node from complete energy exhaustion. The problem of clusterhead rotation is abstracted as the graph theoretic problem of domatic partitioning, which is NP-complete[7, 22].

For this problem we assume that the sensor nodes know their location using global positioning system (GPS). Some of the nodes equipped with GPS can also configure the location of rest of the nodes without GPS using localization[23]. Thus, we assume that each node is aware of its location either using GPS or using localization technique. We develop an approximate self organizing domatic partition algorithm to achieve maximum cluster lifetime of G using the following steps: First obtain a clique partition of the network graph. Next, for each partition, obtain a ranking of the nodes so that the set of nodes having the same rank across partitions yields a domatic partition of G. We define the concept of uncovered nodes in order to make our domatic partitioning as self organizing. We further introduce the concept of uncovered clusters to obtained bounded size clique partitioning. We show that this domatic partitioning scheme has an approximation factor of at least 1/16 for UDGs. The simulation results indicate an improvement of 27% over existing approaches in maximizing the size of domatic partition approximation. Our approach when applied to rotation of the roles of clusterhead via domatic partitioning, substantially improves network lifetime compared to existing clustering schemes.

A distributed algorithm for the maximum connected domatic partition problem For this problem, we describe an approximate solution technique to the maximum connected domatic partition (CDP) problem with a view to maximize the overall lifetime of CDSs in a WSN. For this work, it is assumed that nodes in the WSN are unaware of their location and unable to determine precise distances to their neighbors. Thus, a general ad hoc network model is assumed where nodes can know their immediate neighbors

through message communication only. Our solution to the connected domatic partitioning works in three steps: First a proximity aware clique partitioning is performed. Next a proximity ranking of partition members is made and finally nodes having same ranking are matched to generate a connected domatic partition. We have developed and used a proximity heuristic which uses connectivity information only. Our proximity heuristic is used to perform a proximity aware cluster partitioning which satisfies the following properties: $i)$ the distance between nodes in a partition is at most 2 $ii)$ the size of the partition is bounded lower by a constant and $iii)$ the subset of each partition forms a clique.

We show that the size of a CDP identified by our algorithm is at least $\frac{\delta+1}{(\beta)(c+1)} - f$, where δ is the minimum node degree of G and f, β, c are constants for the UDG for the particular network. Results of testing our algorithm on networks of large number of sensor nodes have shown positive results. Our scheme also performs better than related techniques, such as the ID based scheme.

A distributed algorithm for the minimum connected dominating set problem for computing aggregation backbone Here we have developed an approximation algorithm for the minimum connected dominating set (MCDS) of WSNs which can used as the backbone for lossless aggregation. The nodes in MCDS can perform aggregation function on raw data incoming from several sources to reduce communication by forwarding the aggregated data. For the purpose of aggregation, it is desirable to have a smaller size CDS. Thus, nodes in a MCDS should cover large number of non-MCDS nodes in a network to improve the approximation factor for the MCDS problem.

Our approximation technique for MCDS is a heuristic based approach. We have developed a collaborative cover heuristic which is based on two principles: $i)$ domatic number of a connected graph is at least 2, enabling exploration of a maximal independent set (MIS) for locally best coverage and $ii)$ a set of independent dominators with a common connector form an optimal substructure in CDS. We report a new distributed algorithm which identifies a local best cover heuristically, helping to achieve improved global bounds on the CDS size. We show that the collaborative cover heuristic give better bounds than degree based heuristic because degree alone fails to capture information of actual coverage due to overlapping of node coverage in a distributed setting. Our collaborative cover heuristic based distributed approximation algorithm for CDS construction achieves the performance ratio of at most $(4.8 + \ln 5)\mathrm{opt} + 1.2$, where opt is the size of any optimal CDS. We show that the message complexity of our algorithm is $O(n\Delta^2)$, Δ being the maximum degree of a node in G and the time complexity is $O(n)$. We have also observed through simulation that our CDS approach makes a substantial improvement on the energy dissipation for lossless in-network aggregation function.

A node mobility transparent connected dominating set algorithm We have developed a node mobility transparent CDS algorithm which can adapt CDS to node mobility efficiently. Our node mobility adaptive scheme is an integration of three approaches: (i) self reorganising MPR based CDS construction, (ii) Markov model to assign weights on CDS based mobility profile and (iii) tracking of mobile node by highest weighted shortest path CDS node. The solution is developed in two parts. In both parts self reorganising MPR based CDS construction is used. In the first part only simple location updates of mobile non-dominator nodes is done, while in the second part optimized updation is performed, utilizing the Markov model. The latter technique has an overhead of computing the transition probability matrix, which is moved to the base station to save energy of the sensor nodes. The self reorganising MPR based CDS algorithm adapts with a time complexity of $O(n\Delta^3)$, where Δ is the maximum degree of a node in G. That was further improved to work in $O(n\Delta^2)$ time. The complexity of tracking mobile nodes by our algorithm has been shown to be $O(d \log d)$, where d is number of boundary crossings in the movement of single node. The location updates for mobile nodes gives 40% savings using Markov chain based weighted CDS heuristic over the shortest-hop tracking path in CDS.

Contributions The thesis has four contribution, which are summarized below:

1. We have developed a distributed self organizing domatic partitioning algorithm with approximation factor of at least 1/16 for UDGs. The simulation results indicate improvement of 27% over existing approaches in maximizing the size of domatic partition approximation. When applied to rotation of the roles of clusterhead via domatic partitioning, this substantially improves network lifetime compared to existing clustering schemes.

2. We have developed a distributed algorithm for the maximum connected domatic partition (CDP) problem. We show that the size of a CDP identified by our algorithm is at least $\frac{\delta+1}{(\beta)(c+1)} - f$, where δ is the minimum node degree of G and β, c and f are constants for the UDG for the particular network.

3. We have developed a distributed algorithm for the minimum connected dominating set problem with an approximation factor of $(4.8 + \ln 5)\mathrm{opt} + 1.2$. The smaller size CDS helps to approximate the aggregation backbone for WSNs. We introduce a heuristic which identifies a local best cover guaranteeing an improved global bounds on the CDS size. We have shown that the collaborative cover heuristic gives better bounds than degree based heuristic. Our distributed approximation algorithm for CDS gives the approximation factor of at most $(4.8 + \ln 5)\mathrm{opt} + 1.2$, where opt is the size of any optimal CDS. The message complexity of our algorithm is $O(n\Delta^2)$, Δ being the maximum degree of a node in graph and the time complexity is $O(n)$. Simulation results indicate an

improvement on energy dissipation for our CDS algorithm when used for lossless in-network aggregation function.

4. We have developed a node mobility transparent CDS construction algorithm which helps to adapt the current CDS to node mobility efficiently. Here we have developed the following: $i)$ a self reorganising MPR based CDS construction algorithm, $ii)$ Markov model for weighted CDS, $iii)$ a tracking algorithm for mobile nodes to achieve node mobility adaptation in CDS. The self reorganising MPR based CDS algorithm adapts with a time complexity of $O(n\Delta^3)$, where Δ is the maximum degree of a node in G, which was further improved to $O(n\Delta^2)$. Tracking of mobile node algorithm gives $O(d \log d)$ where d is number of boundary crossings in the movement of single node. The location updates for mobile nodes gives 40% savings using weighted CDS.

1.3 Organisation of the thesis

The thesis has four working chapters, besides chapters on introduction, a review of domination algorithms for lifetime problems in wireless sensor networks and conclusions. The organization of the thesis is as given below.

Chapter 1: Introduction This chapter contains an introduction, literature survey, motivation and an overview of the thesis.

Chapter 2: A review of domination algorithms for lifetime problems in wireless sensor networks Here an overview of topics related to domination in graphs and techniques commonly used in domination algorithms dealing with lifetime issues in WSNs is given.

Chapter 3: Efficient clusterhead rotation via domatic partition In this chapter we describe a self organizing domatic partition algorithm with the objective of providing hierarchical topology control for sensor networks to prolong the life of the network. In this work it is assumed that nodes are aware of their location co-ordinates. An approximation factor for the size of the maximum domatic partition obtained has been derived. Simulation results have been provided to demonstrate the effectiveness of the algorithm for extending network lifetime of WSNs.

Chapter 4: Rotation of CDS via Connected Domatic Partition In this chapter we present a distributed algorithm for constructing the maximum connected domatic partition with the objective of maximizing the lifetime of the CDS in a WSN. In this work it is assumed that nodes are only aware of their local neighbourhoods but not their coordinate locations. Lower bound on the size of the connected domatic partition obtained by the algorithm is also given. Simulation results have been provided to demonstrate the effectiveness of the algorithm for extending the network lifetime for WSNs in providing virtual backbone based on connected domatic partition.

Chapter 5: CDS construction using a collaborative cover heuristic Here a distributed algorithm for the minimum connected dominating set problem based on a single leader is given. An approximation factor for the computed MCDS has been derived. Simulation results demonstrating the usefulness of this technique for effective aggregation over other competitive CDS schemes are given.

Chapter 6: Node mobility transparent CDS construction algorithm In this chapter we present our technique for node mobility transparent connected dominating set construction. Simulation results to demonstrate its effectiveness of algorithms is given.

Chapter 7: Conclusions In this chapter we summarize the contributions of this thesis and present our conclusions. Possible future extensions to this work are also identified.

Chapter 2

A Review of Domination Algorithms for Lifetime Problems in Wireless Sensor Networks

Consider a sensor node as a tiny device consisting of a processing unit with limited computational power and limited memory, sensors (temperature, pressure, humidity, chemical), a wireless communication device (radio transceivers) and a power source in the form of a battery. These sensors when deployed in large numbers and left unattended form an ad hoc network to compute cooperatively. Typically one or more base stations having higher energy resources are present to communicate with the external world. Such a situation often makes difficult in the sensor network to recharge node batteries. Thus, their intended tasks have to be performed under rigid energy restrictions that forces the protocol designers to impose a judicious power management and scheduling on the computing load and energy demands. These constraints and the possibility of node mobility leads to a need for self organisation and dynamic topology control, centralized control being a remote possibility.

Several applications using sensor networks often require only an aggregate value to be reported to the base station. In this situation, physical proximity of sensor nodes (i.e. within transmission ranges of each other) is exploited in a way where sensors in different regions can collaborate to come out with a consolidated report and provide more accurate information about the target region sensed to the base station. Data aggregation reduces the communication overhead in sensor network, leading to a significant reduction in energy usage. The energy load of aggregating node which accounts for computational load of coordination, correlation, compression and long range communication is often well in excess compared to energy requirements for normal operation of a node. Load balancing often rotates the responsibility of high energy overhead to avoid draining the battery of any one sensor node in the network leading to signifi-

cant increase in the lifetime of the node and the network. Optimal scheduling deals with improving the load balancing with bounded extension of the network lifetime. We excluded the optimal scheduling from the scope of this work to address domination techniques which lead to a substantial improvement of network lifetime. Therefore, we assume some optimal scheduling in place for the work in the thesis.

We now present introductory material as background for the subsequent chapters.

2.1 Graph theoretic model for ad hoc and sensor networks

Let a given sensor network contain n nodes and nodes in sensor network are in the ground and each node is equipped with an omnidirectional antenna of maximum transmission range R. Thus, the footprint of such a wireless sensor network is a unit disk graph $G = (V, E)$, where the transmission range of each node is unit disk of radius at most R, $|V| = n$, $E = \{(u, v)|u, v \in V and\ ||u, v||_2 \leq R\}$.

Any two vertices in $V(G)$ are *independent* if they are not neighbors. An *independent set* of G is a subset of $V(G)$ such that all its vertices are mutually *independent*. An *independent set* of G is called *maximal independent set (MIS)* $I(G)$ if any vertex $v \in V(G)$ not in independent set $v \notin I(G)$ has a neighbor in the independent set $v \in N(I)$. Thus, the MIS is a dominating set of G. A *dominating set* $D(G)$ of G is a subset $D \subseteq V(G)$ such that any node $v \in V(G)$ not in $D(G)$ (i.e. $v \notin D(G)$), has at least one neighbor in $D(G)$. A dominating set $D(G)$ is called connected dominating set (CDS), if it also induces a connected subgraph of G. Finding a minimum cardinality connected dominating set in UDGs is NP-Hard[24].

2.2 Models for sensor networks

The UDG model idealizes the real scenario where the radios of all wireless nodes have equal transmission ranges (normalized to 1) such that two nodes can communicate whenever they are within each others transmission range. In ad hoc and sensor networks, the most important graph model is the unit disk graphs (UDG). It is assumed that all nodes are in a Euclidean plane.

2.2.1 Unit Disk Graphs (UDG)

Unit disk graphs are the intersection graphs of equal sized circles in the plane. They provide a graph theoretic model for broadcast networks. Many standard graph theoretic problems remain NP-complete on unit disk graphs such as: coloring, independent set, domination, independent domination and connected domination [24]. There are three kinds of models in unit disk graphs for representing the ad hoc networks:

1. *Proximity model:* Nodes in the network form the vertices of graph and the edges between nodes are formed if the Euclidean distance between nodes is some specified bound d. For example in the clustering problem, to find a maximum subset of points so that no two are at distance exceeding d is modeled as maximum clique partitioning using the proximity model.

2. *Intersection model:* Nodes in the network form the vertices of graph and the edges between nodes are formed when circles formed around the nodes with maximum transmission range intersect. Note that tangent circles are also said to intersecting. For example the problem of frequency allocation in wireless networks is modeled as coloring problem in intersection model.

3. *Containment model:* Nodes in the network form the vertices of graph and the edges between nodes are formed when circles formed around the nodes with maximum transmission range and if one of the corresponding circle contain the others center. For example finding a minimum set of transmitters which can transmit to all remaining stations is modeled as domination problem using the containment model. This is the model we use in this work.

2.2.2 Generalised model

1. *Unit Ball Graphs:* A generalization of UDG is unit ball graph (UBG). Assume that nodes are in some metric space. Two nodes are connected if and only if their distance is at most 1. Each node knows the distances to all its direct neighbors. The UBGs depend on the doubling dimension of the underlying metric. The doubling dimension of a metric is defined as the smallest ρ such that every ball can be covered by at most 2^{ρ} balls of half the radius.

2. *Growth Bounded Graphs:* The most general class of graphs. The growth bounded graphs capture in a simple way the geometric property of wireless networks that if many nodes are close to each other, they will tend to hear each other's transmission and therefore only a small number of these can be mutually independent [25]. For a fixed r, the size of the largest independent set in any r-neighborhood is bounded above by a constant.

2.2.3 Radio model

The radio transmission power level of a sensor nodes is controllable often by software. Let the network density be expressed as $\mu(R)$ in terms of number of nodes per stated coverage area. If N nodes are deployed in a region of area A and the stated range of each node is R, then stated network density $\mu(R) = \frac{N\pi R^2}{A}$. Assume that the receiver and transmitter gains remains the same, the stated transmission range of a radio R is typically a function of its transmit power level P_t. According to the free space radio propagation model (Friss), the received power at distance d is $P_r(d) \propto \frac{P_t}{d^2}$. If the threshold power for reception is P_{th}, then $P_r(R) = P_{th}$. Thus, $R \propto P_t^{\frac{1}{2}}$.

At very short ranges of radio shadowing effects can attenuate specific frequencies, so the frequency hopping techniques are used. Although the correlation of range with transmit power in many cases may be non-ideal, non-radial, non-monotonic and concave, the multiple power levels can still provide coarse adjustment of network density.

If $R^2 = \eta P_t$, where η is constant depends on radio parameters, then doubling the transmit power level can achieve twice the network density given by $\mu(R) = \frac{N\pi\eta P_t}{A}$ [26].

2.2.4 Battery model

1. *Linear model:* An ideal battery is usually viewed as a reservoir of charge from which an amount equal to the load can be subtracted until capacity falls to zero. If C is the capacity of battery at any time, then after the operation duration t_d of continuous discharge of a constant current I, the remaining capacity of battery C' is given by: $C' = C - It_d$. The simple battery model allows the measurement of the efficiency of application.

2. *Discharge Rate Dependent Model:* The assumption of constant current discharge does not model real life batteries. In real life, the battery often drains at increasingly higher rate than the rated current. Thus, the capacity of the battery is dependent on the rate of discharge which is often a *non-linear* discharge behavior. Non-linearity implies that the battery drains at increasingly faster rate when higher loads are applied. Thus, when current I is applied for duration t_d, then remaining battery capacity can be written as: $C' = C(\frac{C_{\text{eff}}}{C_{max}}) - It_d$, where $\frac{C_{\text{eff}}}{C_{max}}$ is excess rate dependent discharge. The value of $\frac{C_{\text{eff}}}{C_{max}}$ at any point of time t is dependent on rate of discharge. *Peukert's law* expresses discharge rate dependent phenomena as a power law relationship, $C' = C - t_d I^{\alpha}$. The exponent α provides a simple way to account for rate dependence. Though easy to configure and use, Peukert's law does not account for time varying loads as most of batteries in

portable devices experience widely varying loads [27].

3. *Relaxation model:* Batteries such as lithium-ion cells show non-linear behavior: rate capacity effect and recovery effect. Research has shown that battery performance can be highly increased by using pulsed discharge instead of constant discharge due to an electrochemical reaction. The periods of rest allow the electrochemical analysis in battery to recover a small part of its charge that is called recovery effect. The recovery effect can be explained by electrochemical analysis. When power is drawn from a battery, the concentration of the active material around the electrode drops, called as polarization effect. When lots of energy is drawn from the battery, it discharges quickly called as rate capacity effect. When the discharge process stops through an introduction of idle period, the polarization effect overcomes which results to a small recovery of the charge of the battery. The amount of recovered energy depends on the current charge of the battery and the duration of the rest time [27].

2.2.5 Network model

The simplicity of the network depends upon the information a node posses. Thus, the amount of information on which the network model rely on can be divided into three types:

1. *Geographic information:* By geographic information, we mean that all nodes know their position in global coordinate system in an Euclidean space. The global coordinate system is meant to configure the nodes with their location using some multilateration technique. The nodes equipped with geographical positioning system(GPS) can configure its position in global coordinate system. Some of the nodes equipped with GPS can configure the location of rest of nodes without GPS in a global coordinate system using localization algorithms. Thus, the nodes in network forms an Euclidean space using geographical information.

2. *Geometric information:* By geometric information, we mean to characterize a network model in which nodes do not have access to the geographical positioning system. The network model assumes that nodes are not aware of position in global coordinate system, but the nodes can sense distances to neighbors. Although pairwise distances may not form an Euclidean space, the pairwise distances induces a metric with constant doubling dimension.

3. *Connectivity information:* By connectivity information, we mean that nodes in network model has neither the position information nor distance information of

its neighbors, therefore rely on connectivity. The model using connectivity information is the most general network model which does not rely on geometric or geographic information so it uses network connectivity information.

2.2.6 Some major issues of sensor networks

2.2.7 Energy efficient schemes

The tiny sensor nodes in wireless sensor networks are deployed and left unattended to observe the target phenomena. The dense deployment and unattended nature of WSNs make it quite difficult to recharge node batteries. Therefore energy efficiency is a major design goals in these network to make it attractive for applications requiring spontaneous deployment and its unattended operations. Energy efficiency is often gained by adding more than optimal number of nodes or by accepting a reduction in network performance. Although such systems do not have renewable energy resource, the system gains lifetime by saving in energy from wasteful energy source.

2.2.8 Fault tolerance

Network and the nodes are prone to failures which needs self organization approach for networks adaptive to fault tolerance. Since large number of sensor nodes is deployed generally much more than optimal number, hence protocols should have in-built fault tolerance mechanism to support uninterrupted operation even though of intermittent faults.

2.2.9 In-network aggregation

Several WSN applications require only an aggregate value to be gathered. Sensors deployed in different regions of the target field can collaborate to aggregate their sensed data and only provide a consolidated report about their local regions. In addition to improving the fidelity of reported measurements, data aggregation reduces the communication overhead and the network loads, leading to significant energy conservation. In-network aggregation takes place as the data flows through multi-hop path to the destination.

2.2.10 Localization

The problem of estimating spatial coordinates is known as localization. Small form factor, cost and energy constraint restrain the use of GPS on all nodes. The localization algorithm are based on beacon broadcasting of nodes with its location information. The nodes on receiving the estimates of pairwise distance uses the *multilateration* algorithm for position estimation. In case of low density of beacons unable to estimate its position to form centroid, the beacon information is propagated through multiple hops to enable locations in areas of low beacon density. Since, localization is beyond the scope of this work, we assume some nodes to be equipped with GPS and some localization algorithm in place to self configure network with its location.

2.3 Node clustering in sensor networks

The role of clustering approach is to provide a hierarchical topology organization in ad hoc network. It means that the goal is to control the topology of the graph representing the communication links between network nodes, with the purpose of maintaining some global graph property while reducing the energy consumption. The topology control has implication on wireless channel of reducing channel contention as many nodes ($\sim 90\%$) can transmit short ranges without interference. Much of the related research reported in wireless sensor networks addresses nodes that come equipped with battery and it cannot be replenished. Hence, maximizing life of node and networks by minimizing energy consumption becomes a research challenge.

Clustering has been shown to improve network lifetime. By clustering WSN, we can partition nodes into a number of small groups called clusters such that each cluster has a coordinator referred to as a clusterhead and a number of member nodes. Clustering results in a two-tier network organization in which clusterhead nodes (CHs) form the higher tier while member nodes form the lower tier. This hierarchical organization supports data aggregation, in which CHs aggregates the data coming from its members and other CHs and forwards aggregated data to other CHs to reach central base eventually. The energy efficiency comes from member nodes comprises of major population ($\sim 90\%$) need to communicate in short ranges and CHs comprises a few ones ($\sim 10\%$) often transmit data over longer distances. Thus, data aggregation and short range communication makes hierarchical organization as most efficient organization of WSN. The network lifetime is defined as time elapsed in operation until the first/last node in network depletes its energy and time until a node is disconnected from the base station. Note that as lifetime is an application specific concept, there is no unified definition for it exists.

Clustering techniques are classified into two types: $i)$ Randomized, and $ii)$ Iterative. The randomized (or probabilistic) approach for node clustering ensures quick convergence while achieving properties such as balanced cluster size. The nodes decides its chance of becoming clusterhead on the basis of weighted function of criteria. This ensures low message overhead on one hand and rapid convergence the other. In LEACH protocol[3] assumes that every node is reachable in a single hop and load distribution is uniform among all nodes. LEACH assigns a fixed probability to every node so as to elect itself as CH. The clustering process involves only one iteration, after which a node decides whether to become CH or not. Nodes take turn in carrying the role of CH. HEED protocol [2] considers multi-hop network and assumes all the nodes are equally important. A node uses its residual energy as the primary parameter to randomly elect itself to become CH. In case of a tie between two CHs, the secondary parameter such as node degree or average distance to neighbors. This results in the uniformly distribution of the elected set off CHs across the network. In HEED each node executes a constant number of iterations. Kuhn's randomized technique [28] to elect CHs depends on node degree. The convergence of their technique which depends on number of nodes and node degree is much faster than iterative techniques.

In iterative clustering techniques, nodes with high weights are preferred to decide about its intention of becoming clusterhead. The problem with iterative schemes is that their convergence speed is dependent on the network diameter (i.e. path with largest number of hops). The DCA algorithm [4] require $O(\sqrt{n})$ iterations to converge for n nodes deployed in an two-dimensional area. Besides the worst case of $n-1$ iterations, the performance is highly sensitive to packet losses. Some schemes enforce a bound on the number of iteration for each node. After some bounded (i.e say 5) many iterations nodes have enough information to achieve stable average cluster size. Schemes also allows a cluster to include nodes that are D hops away from the CH. A node executes $2D$ iterations before making a decision. This results in a constant number of iterations for convergence.

The important issues in node clustering are:

1. Cluster size

2. Clusterhead election

3. Re-clustering for rotation of clusterhead roles

4. Periodicity of re-clustering

2.3.1 Clusterhead election

The main goal of clustering in sensor network is to elect clusterheads. The set of clusterheads forms a dominating set induced by the underlying graph of sensor network. The election of larger degree nodes as clusterheads becomes minimum dominating set problem. Determining optimal dominating set is an NP-complete problem [22], therefore clustering algorithm for sensor networks are heuristic in nature. The clustering technique is classified based on the selection criteria for electing clusterheads: $i)$ ID based clustering, $ii)$ Degree based clustering and $iii)$ Highest remaining energy based clustering.

2.3.2 Rotating the role of clusterheads

The rotation of role of clusterheads among the network nodes is an important technique to extend the life of sensor network by preventing any single node from an early exhaustion of its energy source. There are several issues involved to rotate the clusterhead roles among nodes such as: $i)$ re-electing $ii)$ switching $iii)$ scheduling rotation and (iv) frequency of rotation. That how often the rotation should take place to maximize the lifetime of network is global optimization problem. Finding an optimal schedule is also a global optimization problem. Many randomized algorithm use heuristics for finding optimal scheduling problem.

Periodic re-clustering is necessary mainly due do two reasons:

1. heal the disconnected regions arising due to dead nodes with their complete energy exhaustion and
2. load balancing, to distribute high energy load across all nodes.

The clustering in sensor networks deals with dynamic parameters such as: remaining energy, node degree, etc. which needs re-clustering to remain complaint. Whereas clustering for data processing typically considers static parameters such as: distances between nodes which are reliable. In this section, we review the related clustering approaches that are reported for sensor networks to highlight clustering criteria, assumptions and overheads.

2.3.3 Frequency of rotation of clusterhead roles

Randomized clustering schemes use the number of times a node has been assigned the roles of clusterhead. This ensures high probability to get the clusterhead role than the

node which has already completed clusterhead role in previous rounds. This ensures the load balancing using frequency of rotation as a parameter [29]. An improvement over this is to consider the remaining energy of node so that nodes with good energy resource are preferred to become clusterhead [2].

2.3.4 Cluster size

Most clustering algorithms assumes a fixed transmission range for nodes which generally results in uniform cluster size. The optimal cluster size which can give minimum power for inter-cluster and intra-cluster communication. This problem has been investigated analytically through the centralized approach given the knowledge of complete network [5]. Role rotation in the fixed clusters results in skewed load distribution of cluster heads nodes. LEACH-F has noticed that CHs closer to base station carry more inter-cluster traffic and hence depletes faster their battery resource resulting in reduced life of nodes.

2.4 Hierarchical topology control

There are two approaches for topology control in sensor networks: $i)$ hierarchical topology control and $ii)$ transmission range control. In the thesis work we consider the hierarchical topology control mechanism only. In this part, we briefly review the clustering protocols reported in literature to organize sensor network hierarchically. Clustering can be performed either as centralized or distributed. Centralized clustering can achieve optimal clustering using global knowledge but is energy expensive, hence distributed clustering solution is desirable. However, achieving optimal or near optimal solutions is more difficult in a distributed manner.

LEACH (Low Energy Adaptive Clustering Hierarchy) [3, 29, 5] introduced the technique of randomly rotating the role of the clusterhead among all the nodes for equal distribution of high energy load. In this scheme during setup phase, the nodes organize themselves into clusters with one node serving as the clusterhead in each cluster and a predetermined percentage of the nodes serve as local clusterheads in each round, on average. At the end of a given round, a new set of nodes becomes clusterheads for the subsequent round. Clusterhead change randomly over time in order to balance the energy dissipation of nodes. The clusterhead schedules the nodes in its cluster in TDMA schedule. During the transmission phase, the clusterheads collect data from nodes within their respective clusters and apply data fusion before forwarding them directly to the base station. LEACH provides significant energy savings, prolonged network lifetime by applying localized algorithms and data aggregation within randomly

self elected cluster heads.

HEED (Hybrid Energy Efficient Distributed Clustering) is another protocol to prolong network lifetime, which is also achieved using clustering [2] but, using a hybrid approach: clusterheads are randomly selected based on their residual energy and nodes join clusters such that communication cost is minimized. A node has six discrete transmission power levels. Clustering is triggered periodically to select new clusterheads. Clustering starts with an initial percentage of clusterheads among all nodes. A node sets its probability of becoming a cluster based on estimated current residual energy in the node and maximum energy for the nodes. The algorithm terminates on the probability value of a node, falling below a certain threshold, which is selected to be inversely proportional to maximum energy of nodes. The protocol terminates in a constant number of iterations, independent of the network diameter. The simulation result shows that HEED prolongs network lifetime and expends less energy in clustering compared to generalized-LEACH, although its clustering process requires more than one step for each node.

In a fixed clustering scheme LEACH-F [5], the clusters identified in the initial round becomes fixed. For load balancing, the clusterhead rotates locally within its fixed clusters.

The main drawback of LEACH is the periodic re-clustering elects the new set of clusterheads globally by iterating algorithm. Thus, re-clustering has a substantial wasteful energy overhead. Like LEACH, HEED also involves the periodic re-clustering to elects the new set of clusterheads globally by iterating algorithm repeatedly. Thus, its re-clustering also suffers from a substantial wasteful energy overhead. The fixed clustering scheme results in less energy overhead due to the rotation of clusterheads locally compared to adaptive clustering schemes such as LEACH, HEED. However, the fixed clustering results in a major drawback of higher overhead in communication energy due to skewed inter-cluster and intra-cluster distances. The advantages of adaptive clustering and fixed clustering motivates the need of an efficient load balancing scheme for clustering protocols which should be rotating the roles of clusterhead with the balanced inter-cluster and intra-cluster communication distances.

2.5 Maximum lifetime problem in WSNs

A wireless sensor network once and for all looses possibility of maintenance after its deployment such as node's battery recharge. Nodes in sensor network which are equipped with battery bounds a life span which lasts from the point of deployment till its battery survives. Thus, from the point of its deployment, battery reserve defines the lifetime of nodes and network and the battery resource becomes a valuable resource because the

battery cannot be replenished.

Network lifetime can be defined as the time interval, which the network is capable of performing its intended tasks. In other words, the network lifetime often indicates the time elapsed until the first node drains its battery which is responsible to die down the network. Improving the network lifetime is a challenging issue for system design in sensor networks that can conserve energy resource.

Hierarchical topology control in sensor network using clustering has been accepted for energy conservation in several applications such as data gathering. The clustering maps to the dominating set problem. Finding a small cardinality dominating set maps to minimum dominating set problem. The nodes in dominating deals with higher computational and communication loads than other nodes in network resulting to faster energy depletion of dominating set. Often this poses a problem of maximizing the lifetime of dominating set to improve the network lifetime in sensor network.

For many applications in wireless ad hoc and sensor networks, bare dominating sets are not a well suited organization. Often dominating set needs to fulfill the additional criteria is to be connected. Several applications in sensor networks such as routing and aggregation, often requires a backbone based on connected dominating sets. This poses a maximum lifetime connected dominating set problem for improving the lifetime of sensor networks.

2.6 Self organization in ad hoc and WSNs

The term self organization distinguishes from external control or distributed systems that are based on global state information. Self organization in ad hoc networks is a concept for building scalable system of large number of autonomous nodes using local interaction based coordination and collaboration to provide a desired global aims. The properties of self organization are summarized as:

1. interaction of multiple components.
2. interactions is carried out locally.
3. local interactions achieves a global aim.
4. lack of centralized control.

In ad hoc networks, self organization makes the network configured using spontaneous interaction of the multiple nodes over wireless radio connections without any

external control. Self organization is a paradigm which provides solution to many problem in ad hoc networks efficiently but it is not the only remedy. Other variants such as: Self stabilization is a theoretical framework of non masking fault tolerant distributed algorithms proposed by Dijkstra in 1974. Self stabilizing algorithms can start execution from an arbitrary (illegitimate) system configuration, and eventually reach a legitimate configuration. Ad hoc networks consists of a large number of nodes can use self organization paradigm efficiently than self stabilization a stronger paradigm for distributed systems involving many state changes becomes inefficient for the solution of many problems in ad hoc networks. The self organization is achieved with other following capabilities [30]:

1. *Self configuration:* methods for generating adequate configurations depending on the current situation in terms of environmental circumstances such as connectivity, quality of service parameters etc.
2. *Adaptation:* adaptation to changing environmental conditions such as changing number of neighboring nodes etc.
3. *Self healing:*methods that allow to detect, localize and repair failures automatically. Primarily distinguished by the cause of the failures such as: breakdown, overload, malfunction etc.

2.7 Algorithms for MIS in WSN

Maximal independent set problems has developed interest amongst the wireless networking community due to its practical importance in wireless network applications. In wireless ad hoc and sensor networks, an MIS induces the clustering which is used in various applications such as efficient routing and broadcasting.

The distributed MIS algorithms becomes trivial if network assumes that every node has a unique identifier. In this model MIS algorithm works as follows: every node joins the MIS if it has the smallest ID among its neighbors and if none of its neighbors has already joined the MIS. Thus, obtaining a MIS using deterministic algorithm is easily done if each node knows it exact location and the location of its neighbors.

The deterministic algorithm computes MIS in $O(\log^* n)$ time, if the nodes do not have any position information but can sense the distance to their neighbors [25]. It is an important aspect for sensor network that nodes do not require to know any position information.

The deterministic algorithm computes MIS in $O(\log \Delta \log^* n)$ time, when the nodes do not require any position or distance information where Δ denotes the maximal de-

gree in the network graph. In this model, the only information available with the node is the connectivity information to its neighbors.

Distributed MIS construction with only connectivity information use edge induced subgraphs of bounded degree. The algorithm tries to eliminate unexplored nodes from unexplored set of network until single, locally independent nodes are left. When an unexplored node has no unexplored neighbors, such node joins the independent set. Iteratively, a constant degree graph consisting of unexplored nodes and edges of G is computed.

2.8 Algorithms on MCDS for WSN

In graph theory a *minimum connected dominating set problem* is to find the minimum cardinality among all the connected dominating sets of a graph called minimum connected dominating set(MCDS). Computing an MCDS in a unit disk graph is NP-Hard [24]. Many distributed approximation algorithms for MCDS problem are reported in literature, among them we consider to overview some of the representative MCDS algorithm based on single leader.

8-approximation degree based CDS algorithm by Cardei The algorithms assumes that each vertex knows its distance-1 neighbors and distance-2 neighbors. Algorithms also assumes a designated leader node. The construction of CDS is carried out in two phases: In the first phase the construction of maximal independent set is grown from leader node outwards for the graph G. The initial leader becomes dominator, which identifies its distance-2 independent neighbors. The selection of distance-2 independent neighbors is based on highest degree heuristic. The distance-2 independent neighbors now turns dominators and further initiates the construction in its region. The first phase finishes when all the nodes are explored for the construction of dominators based on maximal independent set of G.

The second phase of algorithm connects the maximal independent computed in the first phase using a Steiner tree to connect all the vertices in MIS.

The second phase of algorithm, connects all the vertices in MIS computed in the first phase. Phase-II uses a Steiner tree, which is a modified distributed depth first search tree. The leader explore within its neighbor the nodes greedily which can connect to the largest number of dominators from MIS. The depth first search tree continues to explore until the leader finds all its neighbors explored which connected all the nodes in MIS.

Approximation analysis outline Let the size of MIS identified in the first phase is $|\mathrm{MIS}|$. The approximation analysis of the CDS size is dominated by the connector which can connect at least 3 dominators in MIS. Therefore, size of connectors required to connect $|\mathrm{MIS}|$ dominators becomes as $\{|\mathrm{MIS}| - 2\}$. The size of CDS is the sum of connectors and dominators computed in two phases of algorithm. Thus $|\mathrm{CDS}|$=$\{2 * |MIS| - 2\}$. Since, the size of MIS is given by $|\mathrm{MIS}| = 4\mathrm{opt} + 1$ reported in [31], on substituting it we get the $|\mathrm{CDS}|$=$8\mathrm{opt}$, where opt is the size of any optimal CDS.

8-approximation ID based CDS algorithm by Alzoubi The distributed CDS construction consists of two phases: an MIS and a dominating tree. First step is to construct a rooted spanning tree. Considering the level number of root node as 0, the ordered pair consisting of level number and node-ID forms the rank of a node. The next step is labeling, which begins from root and ends at the leaves. The root becomes black and sends messages to its 2-hop away independent neighbors through its distance-1 neighbours. If the node at 2-hop receives forwarded messages from all of its lower rank neighbors, then it becomes black and initiates identifying black in its region. The set of black form an MIS incorporating alternate levels of spanning tree. The second phase connects the nodes in MIS to form CDS using message communication. Initially, leader node which is black labeled in phase-I becomes black. When a distance-2 black receives a forwarded message for the first time, it joins the dominator tree along with node which forwarded the message. The second phase ends when all black labeled nodes joins the CDS.

Approximation analysis outline The approximation analysis utilities the properties of unit disk graphs to establish their bounds on the size of CDS obtained by algorithm. From the property of UDG, the number independent neighbors a node have is at most 5. Extending it to a pair of connected nodes, one may ask for the number of its independent neighbors, which gives the value 9 using a simple geometry based on UDG as $4\mathrm{opt} + 1$. In a dominator tree a connector connects at least two independent dominators, thus connectors=$|\mathrm{MIS}| - 1$. The size of CDS=$\{$number of dominators + number of connectors$\}$ = $2 * |\mathrm{MIS}| - 1$ = $8\mathrm{opt} + 1$.

2.9 Algorithms for maximum DP for WSN

In graph theory a *maximum domatic partition problem* is to find the maximum number of disjoint dominating sets of G. The maximum domatic partition problem in G is NP-complete [7]. The maximum number of disjoint dominating sets that can be established is called *domatic number* of a graph G. A greedy heuristic algorithms for finding do-

matic partition is based on idea of pulling out the small dominating sets iteratively until the reminder is no longer a dominating set. The approximation algorithm that guarantee the largest fraction of domatic partition, $\frac{1}{O(\ln \Delta)}$, is due to Feige [7]. The definition of domatic partition can be extended to k-domatic partitions. A k-domatic partition is a domatic partition of $V(G)$ into disjoint k-dominating sets of G. A k-dominating set k-$D(G)$ of G is a subset of $V(G)$ such that each vertex in $V(G)$ is either in k-$D(G)$ or has a k-neighbor in k-$D(G)$. Any vertex $u \in N_k(v)$ is called a k-neighbor of vertex v, if for a given $k \geq 1$ there exists a shortest uv-path of length at most k measured by counting number of edges(or hops) in the path. Intuitively, as k increases, the size of largest k-domatic partition to also increase.

Algorithms for k-domatic partition problems ($k \geq 2$) are reported in [9] for different graph models for sensor networks. These k-domatic partition uses node IDs for partitioning, hence we refer them as ID based domatic partition algorithms in the thesis. We briefly describe the following salient features of ID based k-domatic partition algorithms for different graph models reported in [9]:

Algorithm for k-domatic partition of unit ball graphs (UBG) that reside in Euclidean space ($k \geq 2$) An $O(1)$ round algorithm reported in [9], that computes, for any $k \geq 2$), a k-domatic partition of size at least $(\delta_{k-1} + 1)/c_k$, for some constant c_k, for UBGs that reside in Euclidean space and whose nodes are aware of their global coordinates.

Assumptions in [9] are made that nodes in network reside in d-dimensional Euclidean space for some fixed d and that these nodes are aware of their d-dimensional coordinates with respected to some fixed global coordinate system. It also assumes that nodes have unique node IDs.

The algorithm in [9] computes domatic partition in the following steps: first step is to place a grid of small enough square cells on the plane of dimensions $\frac{1}{\sqrt{2}} \times \frac{1}{\sqrt{2}}$. This induces a clique partition $\nu = \{V_1, V_2, \ldots, V_t\}$ of $V(G)$. Second step, for each clique V_i, assigns a distinct colors $r = \{1, 2, \ldots, |V_i|\}$ to each vertex in V_i. Third step, for each color r, the set of all vertices colored r form a k-dominating set.

Algorithm for k-domatic partition for doubling UBG with nodes able to sense distances to neighbors ($k \geq 2$) An $O(\log^* n)$ round algorithm reported in [9], that computes, for any $k \geq 2$), a k-domatic partition of size at least $(\delta_{k-1} + 1)/c_k$, for some constant c_k, for doubling UBGs whose nodes are able to sense distances to neighbors.

Assumptions in [9] are made that nodes are not aware of their coordinates, but they can sense distances to neighbors. It also relies on the fact that these distances form a metric of constant doubling dimension. It also assumes that nodes have unique node

IDs.

The algorithm in [9] computes k-domatic partition in the following steps: First step, is to compute a maximal independent set(MIS) I in $G_{1/2}$. Second step, each node $u \in V(G) - I$ attaches itself to the partition in $v \in I$ that is its neighbor in $G_{1/2}$. Note that each partition dominated by $v \in I$ forms a clique partition of bounded size. Because, for any partition $v \in I$, the distance between v and any of its member vertex u is less than $1/2$. Thus, given any two vertex u_1, u_2 of partition $v \in I$, the distance is at most 1 (using triangle inequality $|vu_1| \leq 1/2$,$|vu_2| \leq 1/2$, implies $|u_1u_2| \leq 1$). Third step, for each partition $v \in I$, the nodes assign color r, where r is the rank of node based on lexicographical order using node IDs information. Note that this coloring not necessarily proper. Fourth step, the set of nodes of rank r forms k-dominating set of the k-domatic partition of size at most $(\delta_{k-1} + 1)/c_k$.

Algorithm for k-domatic partition for growth bounded graphs ($k \geq 2$) An $O(\log \Delta \log^* n)$ round algorithm reported in [9], that computes a k-domatic partition of size at least $(\delta_{k-1} + 1)/c_k$, for some constant c_k, for every $k \geq 2$), for growth bounded graphs. The term Δ is the largest degree of a vertex in the graph whereas δ the smallest vertex degree.

Assumptions in [9] are made that nodes are not aware of their coordinates nor do they know distance information of its neighbors. It uses only connectivity information. It also assumes that nodes have unique node IDs.

The algorithm in [9] computes k-domatic partition using only connectivity information in the following steps: using connectivity information it was found difficult in [9] to compute a clique partition. In the first step, algorithm computes a partition called uniform partition of G with the following properties: $i)$ each partition induces a subgraph of G of diameter ≤ 2 and $ii)$ for a constant C, the lower bound of each partition is $(\delta_1 + 1)/C$. Second step, for each partition $v \in I$, the nodes assign color r, where r is the rank of node based on lexicographical order using node IDs information, the coloring not necessarily proper. Third step, the set of nodes of rank r forms k dominating set of the k-domatic partition.

2.10 Maximum CDP problem in WSNs

A connected domatic partition of G is a partition of the vertex set $V(G)$, into connected dominating sets. The maximum number of subsets in such a partition is called the connected domatic number of G. Finding of maximum connected domatic partition is of great interest to the wireless ad hoc and sensor network. The maximum lifetime con-

nected dominating set maps on to the problem of maximum connected domatic partition of G. In classical graph theory, some works on connected domatic number is reported for general graphs [32, 33], but for UDGs none of the works on connected domatic partition has been reported before.

2.11 Problems considered in this thesis

The emphasis of the thesis is on the development of domination algorithms for lifetime problems in the design of sensor networks. Some of the domination algorithms investigated as part of the thesis work are as follows: maximum domatic partition problem, maximum connected domatic partition problem, minimum connected dominating set and self organizing connected dominating set for providing transparent node mobility. The problem statement which are identified from detailed literature survey and motivated the need to be addressed in the thesis are stated as:

1. Development of a self organizing domatic partition algorithm for sensor networks,
2. Development of a connected domatic partition in sensor networks,
3. Development of an approximation algorithm for minimum connected dominating sets for aggregation problems for sensor networks,
4. Development of a self organizing MPR based connected dominating set algorithm adaptive to node mobility for ad hoc networks.

In the next chapter, we describe a self organizing algorithm for domatic partition problem in sensor networks.

Chapter 3

Efficient clusterhead rotation via domatic partition

Nodes in wireless sensor networks (WSN) are deployed in an unattended environment with non rechargeable batteries. Thus, energy efficiency becomes a major design goals in WSNs. Clustering becomes an effective technique for optimization energy in various applications like data gathering. Although aggregation aware clustering addresses lifetime and scalability goals, but suffers from excessive energy overhead at clusterhead nodes. Load balancing in existing clustering schemes often use rotation of clusterhead roles among all nodes in order to prevent any single node from complete energy exhaustion. We considered important aspects of energy and time overhead in rotation of the clusterhead roles in various node clustering algorithms with goals to further prolong the network lifetime by minimizing the energy overheads in rotation setup. The problem of clusterhead rotation is abstracted as the graph theoretic problem of domatic partitioning, which is also NP-complete. The dense deployment and unattended nature rules out the possibility of manual or external control in existing domatic partition techniques to be used for WSNs. To our knowledge no self organizing technique exists for domatic partitioning. We developed a distributed self organizing 1-domatic partitioning scheme with approximation factor of at least 1/16 for unit disk graphs.We demonstrate the benefits of self organization without sacrificing the quality of domatic partition, when used in clustering improves lifetime.

3.1 Introduction

Wireless sensors deployed randomly in an inaccessible terrain makes it difficult to recharge their energy source emphasizing the need of designing energy efficient and self organizing protocols. Sensor network application includes remote geographic data gathering where the sensed data is communicated to base station (also known as sink). Clustering protocols address improving lifetime [34] and scalability goals for data gathering applications. In order to prevent any single node from complete energy exhaustion, clustering protocols often rotate nodes having high energy overhead roles to extend the lifetime of network. In adaptive clustering protocols, re-clustering identifies new set of nodes for clusterhead roles for next round [3]. In fixed clustering scheme, the new set of nodes are identified within the fixed clusters [29]. Re-clustering in adaptive clusters has high energy overhead due to its global reset operation [2]. Even though fixed clustering has no overhead in rotation, it has the disadvantage of rigid clusters resulting in high energy overhead due to asymmetric communication. This motivates the need to conserve energy in clusterhead role rotation for sensor networks.

In clustering, every node is initialized such that either it is a clusterhead or a neighbor of clusterhead. Thus, the set of clusterhead nodes in the ad hoc network is modeled as the dominating set problem of graphs in graph theory [22]. Dominating set of a graph $G = (V, E)$ is a subset $S \subseteq V$ such that each node of G is either in S or has neighbor in S. Using dominating sets, only nodes in dominating sets (dominators) must be active or assigned high energy roles while all other nodes (dominatees) can remain in energy saving mode. In this situation lifetime of network depends on the lifetime of dominating set. Maximizing lifetime of dominating set could be finding large number of disjoint dominating sets [35] and activating them successively. The problem of finding maximum number of disjoint dominating sets is called domatic partitioning and the maximum number of disjoint dominating sets is called domatic number of graph. Thus, the problem of rotating the responsibility of being clusterhead (or coordinator) is abstracted as the domatic partition problem. In [10], a schedule for dominating sets in domatic partition is presented which enables nodes in current active dominating set to be active for a fixed period of time. Maximizing lifetime of dominating sets therefore motivates the need to investigate domatic partition problem in perspective of sensor networks.

In this work the problem of rotating the responsibility of being a clusterhead has been abstracted as the domatic partitioning problem. We argue here that ordering of elements in domatic partitions enable the local activation of dominating set which makes our scheme suitable for efficient clusterhead rotation. Here we present an approach for local activation of dominating sets in domatic partition for maximum lifetime problem. We present a distributed, self organizing protocol for domatic partition which enables efficient activation of dominating sets for rotating the responsibility of being

clusterhead. Besides this, we also consider constructing the domatic partition to have an additional property of dominating sets being connected. The extension from domatic partition to connected domatic is not trivial. Although connected domatic partition problems is not the primary scope of this work, it does motivate future study in this perspective. This chapter is organized as follows: section 3.2 presents the preliminary notions, problem abstraction and our contributions. In section 3.3 we describe work related to the domatic partition problem. Section 3.4 presents our approach for self organizing domatic partition. Section 3.5 describes our algorithm for domatic partition, its complexity analysis and discusses generalizations. Section 3.6 is on simulation of the protocol. We close the chapter in section 3.7 with a summary of the work.

3.2 Preliminaries

In this section, we define some graph theoretic terminologies and background for use in the rest of this chapter. The degree of a node (denoted as deg(u) for node u) is the number of neighbors of that node, i.e. the number of nodes within its maximum transmission radius. The *maximum degree* $\Delta(G)$ of a graph G is the largest node degree and the *minimum degree* $\delta(G)$=$min\{deg(u), \forall u \in \mathrm{V}(G)\}$, i.e. the smallest.

A graph is said to be *connected*, if for every pair of nodes there exists a path between them. All nodes of a connected network can communicate with each other over one or multiple hops(through forwarding). A graph is k*-connected*, if for each pair there exist at least k mutually independent paths connecting them. The maximum value of k for which graph is k-connected is the *connectivity* (denoted by κ) of G. Thus, the connectivity $\kappa(G)$ of a graph G is the minimum number of vertices needed to be removed so that G is no longer connected. In a graph G, a pair of vertices are *independent* if they are not adjacent. An *independent set* of G, is a subset $(S \subseteq V(G))$ such that no pair of vertices in S is adjacent. A *maximal independent set*(MIS) of G, denoted by I, is an independent set in which any vertex $v \in \{V(G) - I\}$ has a neighbor in I [28]. A $clique$ in a graph is a set of pairwise adjacent vertices. The graph $G_{1/2} = (V, E_{1/2})$ defines its set of edges $E_{1/2}(G_{1/2})$=$\{(u,v)|(\|u,v\|_2 \leq \frac{1}{2}), \text{for } u,v \in V(G_{1/2})\}$, for any pair of vertices $u, v \in V(G)$. The maximal independent set $I_{1/2}$ is an independent set of graph $G_{1/2}$, such that any vertex $v \in \{V(G_{1/2}) - I_{1/2}\}$ has a neighbor in $I_{1/2}$. A global positioning system (GPS) receiver calculates its position using the signals from four or more GPS satellites for very accurate local time, more accurate than any normal clock can provide, so that the receiver internally solves for 4 variables- x, y, z and t i.e for time as well as position.

The domination problems are important in ad hoc and sensor network. *Domatic partition* is partitioning the vertices $V(G)$ into maximum number of disjoint *dominating sets* of graph G. The maximum number of disjoint dominating sets in graph G is called

domatic number $\mathrm{D}\ (G)$. The domatic number of G is at most $(\delta(G)+1)$. According to Ore's theorem [8], graphs without isolated vertices have two disjoint dominating sets, thus $\mathrm{D}(G) \geq 2$. The problem of finding domatic number of size $\mathrm{D}(G) \geq 3$ is NP-complete [8]. A graph G is said to be *domatically full* if $\mathrm{D}\ (G)$=$(\delta(G)+1)$, the maximum possible. Determining if a d-regular graph is domatically full is NP-complete for $d \geq 3$ [8]. A *connected domatic partition* $\mathrm{CDP}(G)$ of a graph G is a partition of the vertex set $V(G)$, into disjoint dominating sets such that the subgraph induced by each dominating set is a connected subgraph of G. The problem of finding the *connected domatic number* of size $|\mathrm{CDP}(G)| \geq 2$ is NP-complete [8]. For the connected domatic partition problem, the connectivity κ of the graph is an upper bound, thus $|\mathrm{CDP}\ (G)| \leq \kappa$.

The k-domatic partition problem deals with partitioning the vertex set $V(G)$ of G into (preferably large) node-disjoint sets of k-dominating sets, where k means length of shortest path measured by counting intermediate nodes (hops) in that path. Thus, for each k-dominating set, the node in $V(G)$ is either in k-dominating set or there exists a shortest path of length at most k between some node in k-dominating set.

3.2.1 Rotation of clusterheads via re-clustering in sensor networks

In homogeneous *clustered* sensor networks, the role of *clusterhead* burdens a few nodes with more duties than others. The rotation of the role of clusterheads equally among all nodes results in load balancing which extends the network lifetime. *Re-clustering* simply identifies new set of nodes for assigning the clusterhead roles. Thus, periodic re-clustering is an essential operation in *clustering algorithms* of sensor networks for enabling role rotation. Role rotation via re-clustering is a global operation which suffers from a significant energy overhead while rotation. Thus, *efficient rotation scheme* aims at *reducing wasteful energy* in re-clustering using local rotation in spite of rotation. This motivates us to design an efficient rotation scheme aiming for energy conservation in sensor networks.

3.2.2 Rotation of clusterheads via domatic partition in sensor networks

Consider network as a graph $G = (V, E)$ where an edge between a pair of nodes indicates that they are in direct communication range. The clustering of network is abstracted by a dominating set $D \subseteq V(G)$ such that, each node $v \in V(G)$ is either in D or has a neighbor in D. The set of maximum possible disjoint dominating sets P=$\{D_1, D_2 \ldots, D_t\}$ of G is called domatic partition. The set of clusterheads can be

abstracted as a dominating set D, therefore *clusterhead rotation via domatic partition* is abstracted as domatic partition (DP) and scheduling the disjoint dominating set D_i through the domatic partition P=$\{D_1, D_2 \ldots, D_t\}$. We assume some scheduling policy [10] and frequency of clusterhead rotation scheme in place. The domatic partition problem is a NP complete problem of graph theory. The problem of rotating clusterhead roles to maximize cluster lifetime can be modeled as domatic partition problem (DP) of G. We observe that activation of dominating sets is a local operation in contrast to global re-clustering. Thus, rotation ensures efficient activation of fresh dominating set through domatic partition set to replace the active dominating set locally.

3.2.3 Clustering and periodic re-clustering setup overheads

There is a cost in terms of time and energy to set up clusters and periodically re-cluster in clustering protocols for sensor networks. Many clustering protocols use an intuitive argument that steady state period should be long enough compared with the setup phase in their assumptions to amortize the overhead of cluster formation [3, 29, 2, 6]. Notwithstanding running the long steady state phase for amortizing setup overhead, the nodes suffer from an early drainout of their energy resources results to reducing lifetime.

3.3 Related Work

This section is divided in two parts to give the related works on node clustering and domatic partition in sensor networks.

3.3.1 Related node clustering techniques

Clustering becomes indispensable when energy is considered as an optimization parameter. Many clustering algorithms for maximum network lifetime problem have been reported in literature [36] aiming at minimizing the energy spent in communication using different heuristics such as: a minimal numbers of clusters such that each node in cluster is atmost k-hops away from the clusterhead[37] and rotating the clusterhead roles to all the nodes based on some criteria [29, 2, 38]. To our knowledge, none of these algorithms aim at reducing the energy spent in clustering or re-clustering setup phase.

An optimal algorithm for clustering sensor nodes uses the idea of balanced clusters such that total distance between sensor nodes and clusterhead is minimized. Balancing the clusters evenly distributes the load on all clusterhead nodes and minimizing the

total distance reduces the communication overhead results in reduced energy dissipation. The well studied problem of clustering is to minimize the maximum intra cluster distance (between the nodes and clusterheads) over all clusters. In addition to minimizing intra cluster distance, minimizing the inter cluster distance (between clusterheads and base station) also becomes an important optimization for energy conservation in lifetime problem. A clustering scheme reported in [39] called as optimal energy aware clustering in sensor networks use these aspects theoretically. For sensor network in underwater sea applications, a clustering scheme based on similar idea called as minimum cost clustering protocol (MCCP) reported in [6] considers a cost-metric based on minimum intra cluster and minimum inter cluster distances for identifying clusters. The main drawback of these minimum intra cluster and inter cluster distances based approaches is the high energy overheads in setup phase of clustering and re-clustering often requires several iterations for convergence of clustering algorithm. Prior works on node clustering used different heuristic to elect a node as clusterhead such as: residual energy, number of neighbors, node ID, number of times a node has become clusterhead etc.

A weight based distributed clustering (WCA) algorithm [40] uses the weighted combination of node parameters such as: the ideal degree, transmission power, mobility and battery power of mobile nodes for electing the clusterheads. This scheme considers the mobility of nodes which perturb the stability of the cluster configuration due to change in network topology, therefore reconfiguration becomes unavoidable. The clusterheads, forming a dominant set in the network, determines the topology, its stability and the time required to identify the clusterheads depends on the diameter of the underlying graph. The drawback of WCA is apparent under energy constraints situations when the algorithm suffer from high energy overhead in configuration and reconfiguration of cluster setup suggests the scope improvement of lifetime. In distributed clustering algorithm (DCA) [4] application nodes use highest weight among 1-hop neighbor to become clusterhead. The drawback of these clustering is high energy overhead in clustering and re-clustering setup.

In LEACH [29, 38], the energy load of being clusterhead is evenly distributed among nodes by randomized rotation of clusterhead role to all the nodes based on remaining energy resource of the nodes to avoid draining the battery of any one sensor in network. In HEED [2] the clusterhead nodes are periodically selected using a hybrid scheme based on residual energy and node degree for load balancing efficiently. In both these approaches and also others assumes a long steady state to offset the setup load due to clustering and re-clustering operations. HEED incurs as many as 12 iterations for each setup phase [2]. Recently, minimum cost clustering protocol [6] has shown better performance than HEED. The advantage of rotating the clusterhead position among all the nodes in adaptive clusters depend on nodes which are affiliated to the closest clusterheads results in minimal intra cluster energy dissipation for communication the

nodes to communicate nodes to its clusterhead.

In LEACH-F [3], a fixed clustering scheme, the clusters are fixed and only clusterhead were rotated, resulting in nodes have to use a large amount of power to communicate with its head when there is another cluster's clusterhead is close by. Therefore, using fixed clusters and rotating clusterhead nodes within cluster, requires more transmit power from nodes, results in increasing non-clusterhead node energy dissipation often arising due to skewed communication in the clusters. However, the advantage of fixed clusters is that once the clusters were formed, there is no setup overhead at the beginning of each round. Depending on the cost of forming adaptive clusters, an approach where the clusters are formed once and fixed and clusterhead position rotates among the nodes in the cluster may be more energy efficient than LEACH.

Unequal size clustering (USC) scheme[38] analyzed that the unbalanced energy consumption problem of clusterhead rotation is often due to equal size clusters in clustering resulting to an unequal load on clusterhead nodes which can be overcome by having clustering to identify unequal size clusters to incur an uniform energy dissipation among the clusterhead nodes, thus increasing network lifetime.

3.3.2 A review of domatic partitioning

Pemmaraju and Pirwani in [1] have given k-domatic partition algorithms ($k \geq 2$) for various ad hoc network models which is summarized here for UDGs:

The *grid partition based k-domatic partition algorithm* given in [1], assumes an infinite grid of small square cell is placed on deployment plane, where each cell is having dimension of $\frac{1}{\sqrt{2}} \times \frac{1}{\sqrt{2}}$. This induces a clique partition P of the deployed nodes $V(G)$ of G. Thus $P = \{V_1, V_2, .., V_q\}$, where the set $V_i \in P$ is a clique. For each clique $V_i \in P$, a coloring scheme χ, based on node locations, assigns a distinct color $r = \{1, 2, .., |V_i|\}$ to each vertex $v \in V_i$, in clique V_i. For many colors r, the set of all vertices colored r form a k-dominating set of G. This scheme may be used for benchmarking of any domatic partition scheme. The disadvantages of grid partition based k-domatic partition [1] are as follows:

- The partitioning of remote deployment area with a fixed grid structure may not be feasible in practice for sensor network applications.
- The adaptation of grid partitioning based k-domatic partition for self organizing ad hoc networks may be non-trivial.

The aim of this work is to identify domatic partition (for $k = 1$) for UDG model whereas the k-domatic partition ($k \geq 2$) given in [1] cannot be adopted for $k = 1$.

In [10] a scheduling scheme based on a randomized algorithm for domatic partition is given for maximizing cluster lifetime problem.

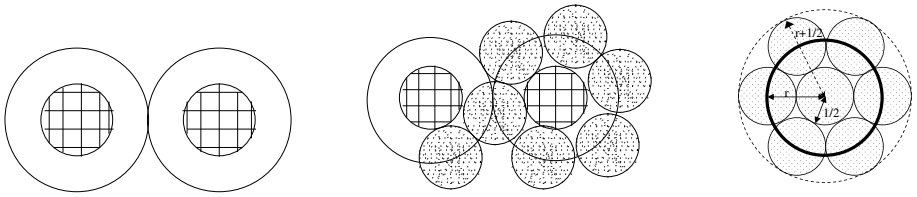
The major drawback of existing domatic partition schemes is the assumption about node's pre-deployed strategy which is based on an external or manual control for placing nodes inside a grid partitions. Due to large number of node deployment and in an inaccessible location, the user control makes its application in ad hoc and sensor network difficult. The technique self organization clearly overcomes from manual or external control using local interactions based on short range wireless communications with the aims to achieve global objectives of particular network configuration.

Our work focuses on efficient clusterhead rotation via self organizing domatic partition for self organizing sensor network. Independent to the clustering, our clusterhead rotation scheme yields energy efficiency by reducing energy overhead in clusterhead rotation setup.

The contribution in this chapter is the domatic partitioning scheme with self organization support in UDGs. From now on, we call our domatic partition scheme as *self organizing domatic partitioning*. To the best of our knowledge no such scheme for construction of *self organizing domatic partition* exists in the literature. Using this, we proposed clustering scheme with efficient clusterhead rotation via self organizing domatic partition which aims at improved network lifetime. The dominating set in domatic partition is not necessarily connected. We also look into makeshift to connected domatic partition problem for *connectedness property* of dominating sets in domatic partition.

3.4 Approach for self organizing Domatic Partition

We describe our self organizing domatic partition scheme in this section which is the basis for efficient rotation of the roles of clusterhead among nodes for achieving load balancing aims at improving network lifetime. We give the big picture of our approach first and then describe the constituent steps in the subsections in detail. The problem of domatic partitioning is achieved in two steps: clique packing and ranking. Clique packing is a network decomposition scheme to identify clique regions having the property that heads form an maximal independent set of $G_{1/2}$ and their neighborhood a clique. The clique neighborhood is defined as, vertices adjacent to v including v itself, enclosed within $\frac{1}{2}$ radius of closed disk around v, denoted by $N_{clique}[v]$, forms an induced subgraph called as clique neighborhood of v. The nodes that are not covered in any of clique packing are called *uncovered nodes*. The property that uncovered nodes should hold is that uncovered node together with clique packing forms an maximal independent set $I_{1/2}$ of $G_{1/2}$. Next step of ranking, use the location information obtained from



(a) Unit disk partitioning

(b) Clique packing:Dark shaded are identified in phase-I and light shaded are identified in phase-II

(c) Dense clique packing in unit disk

Figure 3.1: Clique packing in unit disk

global coordinate system (GPS–global positioning system), which defines the ordering of nodes to evolve a ranking, which assign the ranks to each node in clique. The set of nodes having same rank across each clique in clique packing and uncovered nodes (if not covered) forms a dominating set. The collection of disjoint dominating sets form domatic partition. Finally, rotation of clusterheads roles is to periodically activate dominating set through domatic partition to obtain the clustering with desired properties. Note that the scheduling schemes are beyond the scope of this work, so we have included a simple mechanism for the completeness. For the simplicity of exposition, we present the design for UDG and later show the extension for general graph models. Following five sections present the details of our approach for clique packing, handling uncovered nodes, ranking, domatic partitioning and clustering.

3.4.1 Clique Packing

The goal of clique packing is to decompose network in cliques. In a UDG, the nodes within half radius circle with head at the center forms a clique. Thus, the goal of clique packing is to decompose the network into an independent set of $G_{1/2}$ so that distance between two clique heads is at greater than $\frac{1}{2}$. The following terms as defined below, are associated with clique packing:

- *Clique:* The distance between the head node of a clique and any other node within the clique is at most $\frac{1}{2}$. Therefore, distance between any of two nodes within a clique must be at most 1. Thus the 1-hop clique packing in UDG forms a packing of 1-hop clique. Consider two nodes v, w within clique region in any clique packing. The distance between them and head node (u) is atmost $\frac{1}{2}$. Since, $|uv| < \frac{1}{2}$ and $|uw| < \frac{1}{2}$, therefore, $|vw| < 1$ and all the nodes within partition are within their transmission radius. Thus the partition region forms a clique.
- *Bounded cardinality:* The clique region in clique packing has to be non-empty with cardinality of atleast $\delta(G)/c$ excluding head node for constant c (described

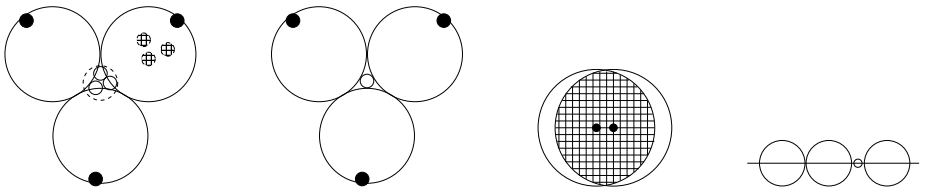
later). Thus, the size of each clique is ensured to be at least $(\delta(G)+1)/c$.

- *Clique packing density:* The number of cliques, which can touch an clique without any intersections, intersecting UDG is defined as clique packing density. Figure-3.1(c) illustrates a proper clique packing. Consider the radius of a 1-hop clique atmost as $\frac{1}{2}$. The distance between two independent nodes (neighboring clique heads) is greater than $\frac{1}{2}$. The density of clique packing is the number of intersecting cliques packed in unit disk. From figure-3.1(c) we calculate the maximum density of clique packing$= \frac{\pi.r^2}{\pi(r/4)^2} = 16 = c$ (details given later in lemma 3.2) to define the lower bound for clique size. Thus, for a unit disk graph, the clique packing density is constant c=16.

No direct method to obtain the clique packing in a network is known to us from literature [41]. However,one can obtain unit disk partitioning of G with the known techniques [3, 29, 2, 38, 42]. In the unit disk partitioning of G, distance between any pair of nodes is atmost 2 in any partition, whereas for clique the property of maximum distance is 1 for any pair of nodes. Therefore, partitioning of unit disks in UDG is not a clique packing. However, from a given unit disk partitioning we could further decompose to obtain the clique packing. Thus we define clique packing as two phase decomposition in our scheme: unit disk partitioning and clique packing.

1. *Unit Disk Partitioning:* The aim is to achieve unit disk Partitioning for bounded size clique in constant rounds. This approach works as follows: Initially a random subset of nodes decides to contest for clusterhead election. Some of them are successful without conflict while others resolves. Using timer based contention resolution, some of them succeed as the clusterheads to form unit disk partitions. We use clustering approach with a timer based atomicity for resolving contention with guaranteed constant round [3, 29, 2, 42, 43]. After $O(1)$ rounds, all the nodes are either within radio range of clusterhead or becomes clusterhead. Unit disk partitions also identify nodes within distance of $\frac{1}{2}$ from clusterheads as a clique. Thus, a partitioning of unit disk with the clique at its center is identified figure-3.1(a), in the first phase and nodes in the unit disk partition that lies between band $\frac{1}{2}$ to 1 has to undergo election in second phase for clique packing as showed in figure-3.1(a). The implementation details of distributed approach are skipped here, for simplicity of exposition. Thus, phase-I forms unit disk partitioning having a clique in each unit disk and only non-cliques nodes participates in second phase for election showed in figure-3.1(a).

2. *Unit disk partitioning to clique packing:* The unit disk partition is further partitioned into cliques in phase-II to identify disjoint cliques. Using the same clustering [3, 29, 2, 38, 42] approach used for unit disk partitioning, only the non-clique nodes undergo election in phase-II to elect nodes as leaders for clique



(a) Uncovered cluster: uncovered nodes inside dotted circle

(b) Uncovered node

(c) Intersection area of uncovered cluster:line marked

(d) Uncovered node prevents re-clustering

Figure 3.2: Uncovered node and uncovered cluster in bounded clique packing of G

partition. For contention resolution, a timer based mechanism is used to guarantee the atomicity for convergence in constant round. Thus, at the end of one or two rounds clique are identified forming clique packing figure-3.1(b). The details of distributed implementation issues are skipped here. Note that size of clique in clique partition is bounded below by $(\delta + 1)/c$ are retained as clique packing for domatic partition whereas the ones not meeting the bounds are termed as uncovered nodes (or uncovered clusters).

3.4.2 Handling uncovered nodes in bounded size clique packing

The nodes, which are unable to form bounded size clique packing become uncovered nodes as shown in figure-3.2(b). We avoid triggering re-clustering in clique packing by handling uncovered nodes separately. For example, a node shown as small circle in figure-3.2(d), becomes an uncovered node by existence of nearby bounded size clique. If the uncovered nodes try to form a bounded size clique then it requires unpacking of nearby clique nodes, thereby triggering re-clustering and leading to several iterations. There can be more than one uncovered nodes in close neighborhood of each other, forming an uncovered cluster (shown in figure-3.2(a)), because of identifying bounded clique in a few iteration rounds. Thus, in bounded size clique packing, uncovered nodes or uncovered clusters may be identified to avoid re-clustering leading to unbounded iterations. It may be noted that for resource constrained sensor networks, the large scale iterations throw substantial overhead on energy resources, therefore keeping some stray nodes as uncovered nodes with a careful mechanism is desirable.

3.4.3 Ranking

The ranking is an alphabetic ordering based on $\langle x, y\rangle$-coordinates of clique nodes in clique packing which helps in identifying disjoint dominating set of G. In the presence of uncovered nodes, the dominating sets also require uncovered nodes as dominators. Therefore ranking of nodes, requires the following mechanisms:

1. *Ranking of nodes in clique packing:* Suppose that each node in network has obtained its physical location from GPS. Each clique in a clique packing assigns a distinct rank to its nodes using alphabetic ordering based on the xy-location of the nodes. The ranking is assigned in following manner: Each node informs its location to its 1-hop neighbors. This requires one round of message exchange msg=$\langle$ID,x,y$\rangle$. After 1-round of message exchange each node collects location information of its neighbors in clique. Now, order the node IDs based on their geographic location at each node in clique. Each node assigns its rank on the basis of it's rank in ordered list within clique. Each node broadcasts message-$\langle$ ID, $\mathrm{clique}_{id}, rank\rangle$ to nodes in its transmission radius.

2. *Acquiring ranks for uncovered nodes:* Each uncovered nodes *acquires* a rank on the basis of its neighboring clique's ranking. Now, we give a simple mechanism to handle the ranking of uncovered nodes to prepare them for domatic partitioning. Each uncovered node keeps track of the ranking of its neighboring clique packing nodes (tracked ones are shown in figure-3.2(a) as grid marked). The uncovered node needs to receive rank announcement of its neighboring clique member's ranks to form an ordered sequence of ranks $\{1, 2, \ldots, max\text{-}range\}$ from any of clique's member nodes irrespective of its clique or node ID. The max-$range$ is defined as:

   max-$range$=min$\{\mathrm{max_rank}\{\mathrm{clique_1}\}, \quad \mathrm{max_rank}\{\mathrm{clique_2}\}, \quad \ldots, \mathrm{max_rank}\{\mathrm{clique_k}\}\}$, where $\mathrm{max\_rank}$ is a function of maximum rank which is assigned to any node of $\mathrm{clique_i}$ and k is the number of neighboring cliques of uncovered node.

   In this way uncovered nodes either acquires the complete ordered sequence of ranks (up to max-$range$) or it can identify the missed sequence numbers.

   (a) *No missed rank in rank-sequence: arbitrary rank* Let an uncovered node receives the complete sequence of ranks from its neighboring clique. In this case, the uncovered node is completely covered by neighboring cliques. Thus, the uncovered node acquires an arbitrary rank from sequence $\{1, \ldots,$max-range$\}$.

   (b) *One missed rank detected in rank-sequence: missed rank* The uncovered node detects one missed rank in the rank-sequence. In this situation, the

uncovered node is covered by all the dominating sets except the missed-rank sequence. Thus, the uncovered node acquires rank as the missed-rank. Therefore, the uncovered node is picked up as dominator in that particular ranked dominating set in domatic partitioning (explained later).

(c) *Multiple missed ranks detected in rank sequence: Either look for uncovered clusters or insufficient nodes* If multiple missed ranks sequence is detected for uncovered nodes then this means an insufficient number of nodes are deployed in network for domatic partition. In this situation at best the uncovered node can look for uncovered clusters to resolve this situation if members of uncovered clusters has similar missed-ranks.

3. *Acquiring ranks for uncovered clusters:* More than one uncovered node in close vicinity forms an uncovered cluster. This brings an optimization for uncovered nodes. The members of uncovered cluster, in addition to forming missed-rank list, exchange the missed-list among each other by a broadcast message and compute the intersection of missed ranks called *missed-intersection-list* (shown in figure-3.2(c)). For a missed-intersection-list, one with lowest node ID of the uncovered node from uncovered cluster acquires that particular rank. Others with common missed-intersection may acquire rank arbitrarily from range $\{1, \ldots,$max-range$\}$.

3.4.4 Domatic Partitioning

In our approach we obtain the domatic partition from clique packing and ranking. The ranking of clique nodes assist to form disjoint dominating sets. The disjoint dominating sets are formed from:-$i)$ clique packing, $ii)$ uncovered clusters and $iii)$ uncovered nodes.

The clique nodes dominates the clique region, therefore they contributes to each of the disjoint dominating sets. All the members of minimum size clique in clique packing necessarily become the dominators in domatic partition and determine the size of domatic number. Since all the members of minimum size clique contribute to the domatic number, therefore nodes which are left over from other cliques are assigned to any disjoint dominating set arbitrarily to satisfy partitioning criteria. It may be noted that a dominating set may not be an independent set.

The uncovered node with an acquired rank either based on missed-rank or arbitrary rank, joins the particular ranked dominating set in disjoint dominating sets. Thus, set of nodes having same rank from each clique of clique packing and same rank of uncovered nodes forms a disjoint dominating set.

It may be noted that in case an uncovered node have multiple missed ranks which are unable to acquire a distinct rank, then it becomes infeasible to construct disjoint

dominating sets of lower bound domatic partition due to insufficient number of nodes in network.

Consider the maximum ranks of minimum size clique as m, which form the disjoint dominating set $P = \{D_1, \ldots, D_m\}$, where D_i is dominating set of G. For each clique in clique packing, members with ranks $\{1, \ldots, m\}$ joins the respective dominating set D_i $(\text{forall } i = 1 : m)$. The members with ranks at least $m+1$ of clique packing are arbitrarily assigned to any of disjoint dominating set P=$\{D_1, \ldots, D_m\}$.

The uncovered nodes with rank joins the respective dominating set in P=$\{D_1, \ldots, D_m\}$. Similarly, in the uncovered cluster, the uncovered nodes with ranks joins the respective dominating set in P=$\{D_1, \ldots, D_m\}$. It may be noted that uncovered nodes contribute to the size of dominating set but does not contributes to domatic number.

3.4.5 Clustering

Consider a given domatic partition of size $(\delta + 1)/c$. Scheduling the dominating sets in round-robin needs a step to construct cluster around the dominating sets. In clustering, nodes in dominating set become clusterhead and allow the neighboring non-clusterhead nodes within transmission range to affiliate with it.

Note that each dominating sets in domatic partition guarantees coverage of target area. Thus not only coverage guarantees but also optimization in reducing the size of dominating set subjected to a guaranteed coverage could be done using our approach. We have included the following lemma-3.1 which establishes the full coverage of network node to function exactly as the existed algorithm reported in [1]. We claim for our self organizing domatic to ensure the full coverage is based on principles of dominating set of graph G. By definition of dominating set of graph, any node of graph not in the dominating set must be the neighbor of some node in dominating set. In our approach the clique packing and un-covered nodes together forms the dominating set. Note that un-covered nodes here only refers to the uncovered in clique packing. Such un-covered nodes when included to the dominating set, gives a full coverage of all the network nodes. Thus, clustering based on dominating sets ensures full coverage. The size of domatic partition is not sacrificed to achieve self organization as observed in comparison with existed algorithm reported in [1], more details are given in the later section-3.6 of the chapter.

Lemma 3.1 *The clusters formed by domatic partition ensures full network coverage.*

Proof: : Consider the domatic partition $D(G) = \{D_1, \ldots, D_{(\delta+1)/c}\}$ of graph G. Each set D_i, for $i = 1 : (\delta + 1)/c$ of domatic partition is dominating set of G by

the definition of domatic partition. The dominating set based clusters of network is induced by the dominating set D_i of G. By the definition of dominating set $S \subseteq V(G)$ of graph G, any node $v \in \{V(G) - S\}$ of graph G not in the dominating set must be the neighbor of some node in dominating set. In our approach the clique packing and un-covered nodes together forms the dominating set. Since, a node from each clique dominates all the elements of cliques. When including un-covered nodes together with clique nodes forms a dominating set of G. Since the dominating set covers the entire network nodes by the definition of dominating set. Therefore set of clusters induced by the dominating set of G ensures full network coverage. The domatic partition $D(G) = \{D_i\}$, for $i = 1 : (\delta+1)/c$, computed by algorithm forms dominating set of G, thus the clustering obtained by disjoint dominating set ensures full network coverage. □

3.5 Algorithms for Domatic Partition and Rotation

The domatic partition problem is NP-complete [44]. Approximation algorithm guarantees $O(\ln \Delta)$ approximation for domatic number which is the best possible approximation for general graphs unless $\mathrm{NP} \subseteq \mathrm{DTIME}(n^{O(\lg \lg n)})$ [7]. Thus, greedy construction of domatic partitions to find large number of disjoint dominating sets in earlier works [10, 1, 7] has not considered an important aspect of self organization. To our knowledge, there is no self organizing approximation algorithm for 1-domatic partition problem, which is needed for efficient and coverage preserving protocols for sensor networks. In this section we describe three algorithms for finding large domatic partitions and scheduling activation of its disjoint dominating sets. We assume that the node of the networks with unique IDs has location information using access to global positioning system (GPS). We assume a UDG model for keeping simplicity of exposition; we show that our scheme works for other complex graph models but with increased time and message complexities.

Algorithm-1 constructs clique partitions using 2-phase approach. The iterative rounds of first phase ensures the partitioning of G in unit disk. By definition a unit disk contains a clusterhead with any pair of nodes within unit disk being at most at distance-2. In the first phase of algorithm-1 a set of nodes tries to become clusterhead so that no two adjacent clusterheads are within the direct transmission range. The contention among the nodes is resolved using a timer within period τ. If all the node are not covered, then iteration continues until either a node is clusterhead or having a clusterhead (one or more) as it neighbor. Therefore, after a few rounds the set of clusterheads form a maximal independent set I of G. This partitioning is called unit disk partition which satisfies the following three properties:

1. Each partition in unit disk partitioning is dominated by clusterhead.
2. Any pair of member nodes of a partition is at most at distance of 2.
3. The set of clusterheads forms maximal independent set I of G.
4. The nodes within $\text{radius } of\ \frac{1}{2}$ forms a clique of partition and rest of nodes are non-clique of partition.

The non-clique members of each partition undergoes further clique partitioning for electing leaders in phase-II to identify cliques which we call as clique packing.

Algorithm 1 Two Phase Clique Packing

Input: UDG, IDs, 2D-location, τ

Output: Bounded Clique packing $U = \{V_1, \ldots, V_s\}$ of V(G), uncovered cluster, uncovered nodes

(* **Phase-I (Unit Disk Partition)** *)
Election of clusterhead so that two nodes of more than 1 units becomes clusterhead.
Contention resolution time τ used for clusterhead election
Affiliate nodes within $\frac{1}{2}$ distances as clique
Nodes greater than $\frac{1}{2}$ distances from clusterhead becomes candidates for phase-II election. *Note that phase-I Identifies set of clusterheads as maximal independent set (MIS) I of G, s.t. for any neighboring $x, y \in I(G)$ ($1 < \|xy\|_2 \leq 2$)*
(* **Phase-II (Clique Partition)** *)
Election for non-overlapping clique so that any two neighboring clusterhead are at least at distance $\frac{1}{2}$
Contention resolution timer with τ time period used for clusterhead election
Affiliate remaining nodes to closest clusterhead of clique
The cliques is bounded with at least the size $(\delta + 1)/c$. The nodes failed to achieve bound becomes *uncovered nodes* or *uncovered cluster* to avoid re-clustering
(* **Handling of uncovered nodes** *)
Nodes failed to form or join bounded clique partition (to avoid violation) are called *uncovered nodes*. More than one uncovered nodes forms an *uncovered cluster*. *Note that phase-II Identifies set of clusterheads for clique and uncovered nodes as maximal independent set (MIS) $I_{1/2}$ of $G_{1/2}$, s.t for any neighboring $x, y \in I_{1/2}(G_{1/2})$ ($\frac{1}{2} < \|xy\|_2 \leq 1$). Thus uncovered node and clique packing together satisfies maximal independent property $I_{1/2}$ of $G_{1/2}$, required for proper domatic partition.*

The second phase starts with non-clique region of every unit disk partition to identify clique packing of G. The election of leaders for clique is also carried out using timer with period τ to resolve the contentions. The contention resolution is based on the idea of allowing a formation non-overlapping of disk of at least $radius = \frac{1}{4}$, whereas resolving the contention in case of overlapped disk formation is detected by member

nodes. The clique partitions thus formed satisfies the lower bound $(\delta + 1)/c$ for cliques in clique packing. The set of nodes not meeting lower bound property for cliques, turns uncovered, such partition or node called as *uncovered cluster or uncovered node*. The properties of clique identified in 2-phase algorithm-1 are the following:

1. The distance between any pair of nodes in a clique of clique packing is at most 1,
2. The distance between heads of any neighboring cliques in clique packing is greater than $\frac{1}{2}$ and
3. The size of clique in clique packing is bounded below by $(\delta + 1)/c$ for some constant c in UDG (defined later in lemma-3.2).

In a few iterative rounds, there may be an uncovered area as a result of non-overlapping area of clique packing. When the *uncovered nodes* is more than one then they are referred to as an *uncovered cluster*. The size of uncovered clusters is bounded above by $(\delta + 1)/c$, which are unable to construct clique in clique packing. The uncovered nodes or uncovered cluster satisfies the following properties:

1. The uncovered nodes together with the clique clusterhead forms a maximal independent set $I_{1/2}$ of $G_{1/2}$ and
2. The uncovered cluster size is bounded above by $(\delta + 1)/c$.

Algorithm-2 has three goals: $i)$ ranking of nodes in clique of clique packing using location information obtained from geographical positioning system (GPS), $ii)$ detect the set of missed ranks in the neighborhood of uncovered nodes for which there are no clique nodes bearing the rank in its neighboring cliques and acquire the rank either as missed-rank or arbitrary rank and $iii)$ optimization of handling missed-ranks for uncovered clusters (already explained in previous section).

To achieve these goals, the algorithm-2 adopts the following simple mechanism. In algorithm-2, the nodes in clique broadcast their IDs to form clique neighborhood list which is then sorted in non-descending order of location (based on global positioning system GPS) so that each node assigns a rank in ordered list. The elements of clique then broadcasts its ranks. In the process of ranking of clique packing and to construct disjoint dominating sets, the uncovered node has to keep track of rank-sequence of its neighboring clique of clique packing and acquires a rank. The uncovered nodes on receiving the neighboring clique-member's ranks arranges to find out complete series of ranks up to the minimum of neighboring clique's maximum ranks. Similarly, an *optimization* is applied for uncovered clusters. They exchange their missed-rank list among themselves so that they computes the intersection of missed-ranked list called

Algorithm 2 Domatic partitioning from clique partition and uncovered nodes

Input: Partition $U = \{V_1, \ldots, V_s\}$ of $V(G)$ where each V_i is a clique, Nodes have its location from GPS

Output: Domatic Partition $D = \{D_1, \ldots, D_d\}$ of $V(G)$

1: Each node in clique broadcasts 3-tuple $\langle x, y, \text{cluster}_{id} \rangle$ to all neighbors
2: Each node receives IDs from neighbors in clique. *Note that neighboring uncovered node also receives it*
3: Each node constructs local list of IDs of clique members
4: Compute locally the sorted list of IDs on the basis of location information. Assign the rank to itself from locally computed ordered list
5: Nodes of same rank forms dominating set say D_i and collection disjoint dominating sets forms domatic partition set D
6: Clique Nodes broadcast information of its ranks and affiliated clique clusterhead (* **Acquiring ranks for uncovered nodes** *)
7: Uncovered nodes needs to receive the complete range of ranks of its neighboring cluster and identify the *missed ranks* in the rank-sequence and broadcast the missed-ranks to neighboring uncovered cluster nodes
8: Uncovered cluster nodes on receiving missed rank-list, they compute its intersection, to get *common-missed ranks*
9: The uncovered-node without any missed-rank, joins arbitrarily any one of disjoint dominating set
10: For a *missed rank* j, the uncovered node becomes dominator in dominating set D_j. For *common-missed* ranks j, any one of *uncovered cluster* k becomes dominator D_j
11: Multiple missed-ranks of an uncovered node to look for uncovered cluster with common missed-ranks to achieve resolution of coverage towards formation of dominating set

as *missed-intersection-rank list*. The uncovered node with lowest node ID becomes the dominator for that particular ranked dominating set. Thus, the nodes of same ranks from each clique of clique packing and uncovered nodes with same acquired ranks together forms dominating set.

Therefore, the node disjoint dominating set partitions the $V(G)$ of G such that each partition becomes the dominating set of G.

In algorithm-3, scheduling of dominating set, assigns the role of clusterhead to nodes. The clusterhead solicits affiliation from the non-clusterhead nodes. In clusterhead rotation, periodically the new dominating set from domatic partition is scheduled periodically.

Algorithm 3 Clustering from Domatic Partition

Input: Domatic Partition $D = D_1, \ldots, D_d$ of $V(G)$
Output: Clustering with rotating clusterhead

1: **repeat**
2: **for all** $i \in 1..d$ **do**
3: Periodically schedule nodes of D_i to be active for Δt
4: Nodes of D_i becomes clusterhead announces neighbors for affiliation
5: On receiving clusterhead announcement, nodes join arbitrary cluster
6: **end for**
7: **until** all dominating sets exhaust their energy

3.5.1 Lower bound approximation factor of domatic partition

Lemma 3.2 *The lower bound approximation factor on size of domatic partition for algorithm-2 is constant* $\frac{1}{16}$ *in UDG.*

Proof: Consider the two phase algorithm-1, for clique packing. In order to identify the lower bound of domatic partition, we first analyze the size of clique obtained in algorithm-1. In the first phase of the algorithm, the unit disk partition of graph G is obtained, where the diameter of each partition is at most two. As a result in the second phase of clique packing, the diameter of a clique is obtained at most one. For clique packing, the set of nodes at the center of each clique should be at a distance greater than $\frac{1}{2}$. For a pair of neighboring cliques with centers at v_i and v_j, $\mathrm{dist}(v_i, v_j) > \frac{1}{2}$. Let $I_{\frac{1}{2}}$ be an MIS of $G_{\frac{1}{2}}$ and let $v_i, v_j \in I_{\frac{1}{2}}$. If we consider a disk of radius $r = \frac{1}{4}$, placed around nodes v_i and v_j, then they are non-overlapping, shown in figure-3.3. This means that for a given node v_i the number of non-overlapping disks in area of $N[v_i]$ is at most of the size $\frac{\pi r^2}{\pi\left(\frac{r}{4}\right)^2}$=$16 = c$ (a constant for UDG).

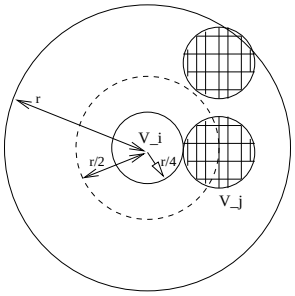


Figure 3.3: Lower bound of domatic partition size for algorithm-2

The size of clique for node v_i is bounded below by $(N(v_i)+1)/c$. Since, $\delta \leq N(v_i)$, therefore minimum clique is bounded below by $|clique_i| \geq \frac{\delta+1}{c}$. The upper bound for optimal domatic partition size is $(\delta + 1)$, therefore approximation factor of size of domatic partition of our algorithm is $\frac{1}{c}$=$\frac{1}{16}$. Thus the approximation factor of domatic partition is 1/16. □

3.5.2 Correctness of algorithm

Lemma 3.3 *The set D_i, for $i = 1 : (\delta + 1)/c$, computed by algorithm-2 forms dominating set of G.*

Proof: : Let us consider the dominating set D_i for a particular value of i. Since, the distance between any two members with *same rank i* of D_i is greater than 1. Therefore, the nodes with the *same rank i* cannot lie within radius 1, thus forming an independent set. Since dominating set D_i, $r \in \{1, 2, \ldots, (\delta+1)/c\}$ contains one member from each clique, i.e. dominator from all the cliques and uncovered nodes. Thus the set of nodes with the same ranks i forms maximal independent set of G. Besides, the nodes of same ranks i, the dominating set also contains nodes with other ranks to adds the redundancy and better coverage. Thus, the set D_i, $r \in \{1, 2, \ldots, (\delta + 1)/c\}$ is a dominating set.

□

Algorithm-2 partitions vertices of G into disjoint dominating sets. Hence it yields domatic partition. After a domatic partition is identified algorithm-3 performs clustering. The scheduling of dominating set for a maximum duration T, invites affiliation of member nodes. Non-clusterhead nodes join any adjacent clusterhead. Thus algorithm-3 yields clustering around disjoint dominating sets.

Lemma 3.4 *Let $D = \{D_1, \ldots, D_k\}$ be a domatic partition of $V(G)$. For each node $v \in V(G)$, either v or its neighbor $N(v)$ is in dominating set D_i for all $i = 1, k$. For each D_i to be dominating set, the following boolean equation holds* true*:*

$$B_1 = \bigwedge_{j=1}^{|V(G)|} [\bigwedge_{i=1}^{k} [(v_j \in D_i) \vee (N(v_j) \cap D_i) \neq \varnothing]] \tag{3.1}$$

Proof:

Consider an arbitrary graph $G = (V, E)$ and a number $k \leq |V|$. For each partition D_i in $D = \{D_1, \ldots, D_k\}$ to be dominating set, then for each node $v \in V(G)$, either node v or its adjacent node set $N(v)$ is in D_i for each $i = 1, k$. Thus, for each $1 \leq i \leq k$, apply dominating set $D_1, \ldots, D_k$ to test for dominating V(G) by each $D_i \in D$. If it is tested true then either $v \in V(G)$ itself is i^{th}-dominating set D or one of its neighbors in D_i. Thus, evaluation of boolean expression equation-3.1 holds true for D computed in algorithm-2 to ensure set D as a set of dominating sets of G.

□

Lemma 3.5 *Let $D = \{D_1, \ldots, D_k\}$ be a domatic partition of $V(G)$. For each node $v \in V(G)$, there exists at most one dominating set D_w s.t $v \in D_w$, for which it is dominator. For D to be node disjoint collection of partition sets D_i, the following boolean equation holds* true*:*

$$B_2 = \bigwedge_{j=1}^{|V(G)|} \exists w[w|(p \geq 1) \wedge (w > p) \wedge (k \geq q > w)]$$
$$[(v_j \in D_w) \wedge (\bigwedge_{p=1}^{w-1} v_j \notin D_p) \wedge (\bigwedge_{q=w+1}^{k} v_j \notin D_q)] \tag{3.2}$$

Proof: The other mandatory property for collection of dominating sets in domatic partition is disjointness of dominating sets i.e all the vertices must be covered in dominating set and each vertex must be member of one and only dominating set. Let i, j, w be the index variable which can take values $\{1, \ldots, k\}$. The formula is divided in three clauses: $i)$ there exists an index w for which node $v \in V(G)$ is in dominating set D_w $ii)$ there must exist a lower index variable $i \leftarrow \{1, \ldots, w-1\}$ the node $v \notin D_i$ and $iii)$ there exists an index variable $j \leftarrow \{w+1, \ldots, k\}$ for which node $v \notin D_j$. Similar to lemma-3.5, for each vertex $v \in V(G)$ and $1 \leq i, j \leq k$, $i \neq j$, apply the test case for its being disjoint. Thus, evaluation of the boolean expression equation-3.2 holds true ensures disjointness criteria. If it is true then sets D must be disjoint. Thus, for given k it is validated to form a node disjoint set in domatic partition of size at least $k \leq |V|$. □

Theorem 3.6 *For any d, the set $D = \{D_1, \ldots, D_d\}$ computed by algorithm-2 forms domatic partition.*

Proof: : Consider partition $D = \{D_1, D_2, \ldots, D_d\}$ identified by algorithm-2. From lemma-3.4, we verify that each partition D_i is dominating set of G. From lemma-3.5, we verify that the set D holds node disjointness property. On the basis of mandatory properties: Collection of dominating sets and node disjointness, we can say that the algorithm-2 computes domatic partition of G. □

3.5.3 Generalizations

The generalization of approach presented is considered in this section. Beyond UDG, the approach also works for general graph models: UBG and Growth bounded graphs. Furthermore, the approach for domatic partitioning can be extendible to connected domatic partitioning.

1. *Generalization to UBG (Unit Ball Graph), Growth Bounded Graph:* The UDG model assumes that network resides in 2-dimensional Euclidean space. Our approach extends in a straightforward manner to UBGs in d-dimensional Euclidean space. Each clique has the diameter 1 and therefore each non-empty d-dimensional hypercube induces a clique in G [25]. In growth bounded graph the maximum independent set is computed in $O(\log \Delta . \log^* n)$ rounds in distributed algorithm using messages $O(\Delta . \log n)$ in [25]. Thus, the algorithm still works but with the complexities of growth bounded graphs dominates.

2. *Connected Domatic Partitioning:* Extending the domatic partitioning scheme to connected domatic partition is difficult than constructing a minimum connected dominating set. The Domatic partition problem find as many disjoint dominating sets of a graph as possible. In the connected version, an additional requirement is that each dominating set induces a connected subgraph in G. Thus, problem is NP-complete for maximum connected domatic partition problem. A simple observation is that a small fraction of dominating sets out of domatic partition may satisfy additional connectedness. The exact relationship of connected domatic partition and domatic partition is kept out of scope of this work. However, we modify the design aspect to convert it for connected domatic partition. For connected domatic partition we modify ranking of clique partition algorithm to extent that ranking of independent nodes needs to include a forwarding node from adjacent clique. Since the distance between two adjacent clique nodes can be at most 1, thus more cliques resulting to a non-overlapping cliques packing having

the radius ranging from $\frac{1}{4}$-to-$\frac{1}{2}$. We observe that the connected dominating size is larger to the size of dominating set, so reducing the size of connected domatic partition than domatic partition.

Lemma 3.7 *The connected domatic partition is bounded by $\frac{16}{25}$ factor of domatic partition.*

Proof: : Consider that packing of $\frac{1}{2}$-radius disk within distance-2 Unit disk. Then, the ratio of packing density of clique is $\frac{1}{25}$ times approx. for connected domatic partition. Hence, connected domatic partition is bounded $\frac{16}{25}$ of domatic partition. Note that to obtain $\frac{1}{25}$ approx. factor for connected domatic partition a careful ranking results to a more substantial computing. □

3.5.4 Distributed Complexity Analysis

In order to collect complete 1-neighborhood, each node broadcasts its status to all neighbors; it thus requires 1-round of message exchange. The remaining steps in algorithm-2 compute the output by locally simulating the computation in distributed algorithm without any communication. Similarly, for clustering round in algorithm-3 also needs a round. Thus the distributed algorithm need constant rounds $O(1)$. The message size is constant $O(M)$ where M is size representation of location information. However, for k-domatic partition the complexity-(time and space) increases to collect k-hop neighborhood information. Thus it is $O(k)$-rounds and $O(M^k)$ message size for k-domatic partition in [1]. The lower bound for the size of domatic partition set is at least of approximation factor $\frac{1}{16}$ to the optimal size. The proposed algorithm gives approximation factor of $\frac{1}{16}$ and approximation factor $\frac{1}{25}$ for connected domatic partition in UDG.

3.6 Simulation of Protocol Behavior

As mentioned earlier, the domatic partition is used for clustering based on dominating set in sensor networks. Therefore, important factors that need to be considered for evaluating our approach are as follows:

1. *size* of domatic partition,
2. *energy* overhead in clusterhead rotation setup,
3. *time* overhead in clusterhead rotation setup and

4. network *lifetime*.

We have conducted simulation to evaluate the performance of our algorithm in the above four areas.

First we compare the size of domatic partition computed by our self organizing domatic partition algorithm with the k-domatic partition scheme reported in [1]. The first objective is to demonstrate through simulation that the benefit of self organizing is achieved without loosing quality of domatic partition. Later, we show that overheads in achieving self organizing feature in domatic partition is amortized by considering a long steady state application in clustering. Therefore, we next examine the effect on the network lifetime by clustering approach based on our self organizing domatic partition and present results obtained by comparing to different clustering protocols [3, 2, 6]. Thus, our objective is to establish through simulations the net gain in network lifetime using clustering technique considering overheads in re-clustering setup (including self organization and re-clustering).

For simulation experiments, we consider a sensor network of nodes $N = 100$ deployed in $100m \times 100m$ region randomly. We assume that all the nodes start with uniform energy. We adopted a linear energy model same as given in [3, 29] to compare the lifetime performance with representative clustering algorithms. The simulation parameters are summarized in the figure-3.6 which are drawn from the experimental setup given in [3, 29, 2, 6]. For simulations, we used Prowler/Matlab [42], event driven simulator for sensor networks. After the domatic partitioning we schedule disjoint dominating sets using some scheduling technique. For clustering based on our self organizing domatic partition, we assume some scheduling scheme in place to rotate clusterhead based on dominating sets.

3.6.1 Performance analysis for domatic partition

We have gathered results to look at the basic issues related to quality of domatic partition in terms of its size. Thus, the main issue considered in this section is: Whether we achieve self organization without sacrificing quality of domatic partition.

We give the performance of self organizing domatic partition in this section. We analyze the trend of clique partition and the quality of dominating set computed by our approach. Next, we compare the size of domatic partition computed by our self organizing domatic partition with the k-domatic partition technique reported in literature [1].

1. *Trend of clique partitioning in domatic partition:* We have used algorithm-1 to show the trends of intermediate stage results in our approach. In the experiments,

Parameters		
E_L	$50nJ/bit$	radio electronics energy
E_{agg}	$5nJ/bit$	energy dissipation for aggregation
Bw	$1Mbps$	bandwidth
ϵ_{friss}	$10pJ/bit/m^2$	radio energy (Friss att. model)
$\epsilon_{2-\text{ray}}$	$0.0013pJ/bit/m^4$	radio energy (two-ray att. model)
$E_{battery}$	$2J$	initial battery
N	100	number of nodes
$M \times M$	$100m \times 100m$	target area
BS	$(110, 110)$	coordinates for base station
l	$1000bit$	data messages size
$l_{control}$	$200bit$	control message size
N_{TDMA}	$5frames/round$	number of frames in TDMA round
R	10	Radio transmission range

Figure 3.4: Simulation parameters

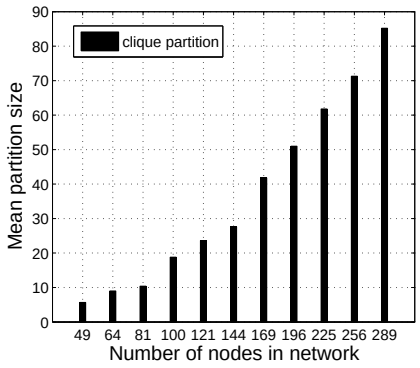


Figure 3.5: Clique partitioning in algorithm-1

we fixed the target area and increased the number of nodes to analyze the clique partitions. Figure-3.5 shows the trend of mean size clique partitions. We observe that average size of clique partition is increases linearly with the number of nodes in network. The linear trend in average size of clique partition shows that self organizing algorithm is able to partition the network properly.

2. *Trend of dominating set in domatic partition* Quality of dominating sets obtained in domatic partition is of special interest to clustering protocol. Here, we obtain the results through algorithm-2 to look into the trend of the size of dominating set computed by our domatic partition algorithm. We measure the mean size of dominating set for different network sizes. The trends in figure-3.6 show that the dominating set size is slowly increasing with the size of the network, on a fixed

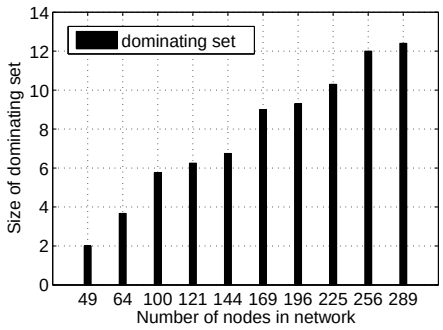


Figure 3.6: Quality of dominating set in algorithm-2

target area. The smaller the size of dominating sets, the larger the domatic set yields. Thus, we observe a constant approximation to the size of dominating set on varying network size to large size. The dominating set covers the entire target area, thus disjoint dominating sets are coverage preserving.

3. *Performance comparison of domatic partition size* We analyze the performance of our self organizing domatic partition algorithm in terms of mean size of domatic partition identified by our algorithm. For the performance comparison, we used k-domatic partition algorithm($k \geq 2$) reported in [1]. In order to compare the effect of our self organizing, we adjust our approach for $k = 2$ where k-domatic partition means disjoint k-hop dominating sets. For, 2-domatic partitioning, we consider 1-hop clustering to be clique partition for $k = 2$, thus it simplifies our algorithm to consider for self organizing 2-domatic partition. Figure-3.6, shows the performance comparison of our self organizing approach with the clique partition using manually placed grid on the region. Inspite of self organizing characteristics, our self organizing domatic partition could achieve a similar performance in terms of sizes of domatic partition. Thus, we observe that self organizing characteristics is not sacrificing to the size of domatic partition. In figure-3.6, we compare the quality of domatic partitioning with algorithm of [1]. The metric of comparison is partition size, we observed that our approach yields the size of domatic set similar compared to [1]. In spite of its being a self organizing protocol, compared to fixed grid topology for clique partitioning of schemes [1]. Thus, our proposed algorithm is distributed and self organizing which discovers their neighbors based on radio communication, has achieved similar approximation factor compared to scheme [1], our algorithm has not sacrificed quality of domatic partition. In figure-3.5 we observe that our approach also gives similar results compared to scheme in [1]. We also state that out of the domatic partition

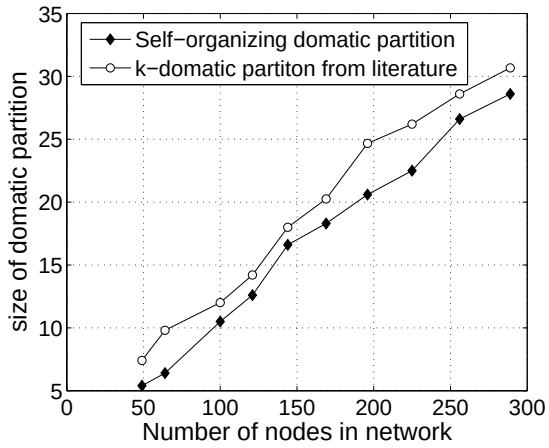


Figure 3.7: Comparison of domatic partition size

some of the dominating sets have connectedness property. The exact relationship between domatic and connected domatic is an open area of research. So we resort to our simulation results for its comparison. In figure-3.5 we observe that for large network sizes, the fraction of connected domatic partitions is small.

3.6.2 Comparison of *time* overhead in clusterhead rotation setup

The temporary unavailability of network services during the setup phase (i.e clustering and re-clustering) is also an important factor for measuring time overhead of the setup phase while comparing the clustering protocols. In order to account the time overhead involved for setting up clusters and re-clustering for its rotation, we define the term *network unavailability* as the time period when the network is busy in setup phase and remain unavailable temporarily to the services. For the time overhead in setup, we assume nodes communicate the control messages in clustering setup and remain unavailable for duration $\frac{l_{control}}{bandwidth}$ seconds. A comparison of time overheads in rotation setup of our domatic partition based clustering approach with different clustering protocols [3, 2, 6] is given in figure-3.9.

We observe that domatic partition based clustering is quick in rotation of clusterheads because the re-clustering involves only a local operation in switching up of clusterhead role to the new nodes becoming clusterheads. On the other hand the clustering protocols [3, 2, 6] require several iterations for identifying new set of clusterheads globally, for example, LEACH [3] involves single iteration for re-clustering compared to HEED [2] (and MCCP[6]) which needs several iterations (upto 12) for re-clustering setup. Thus, for time critical applications, our domatic partition based clustering re-

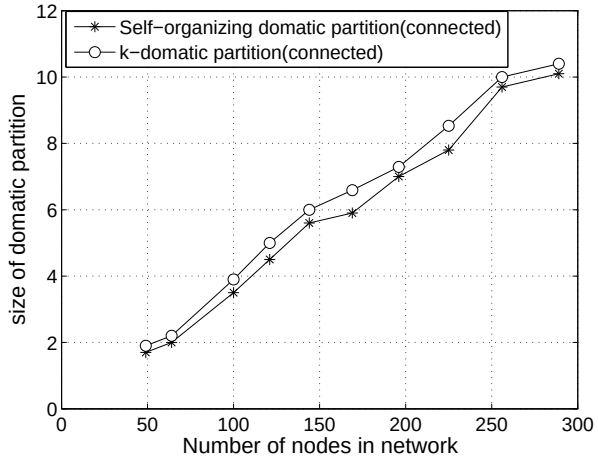


Figure 3.8: Comparison of domatic partition (connected) size

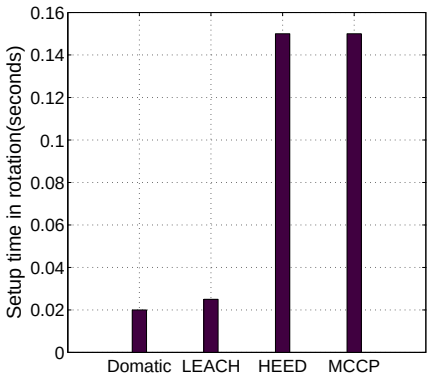


Figure 3.9: Comparison of setup time for rotation

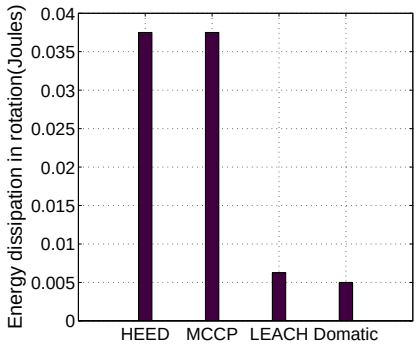


Figure 3.10: Comparison of rotation energy dissipation

duces the non-availability time overhead by about 86% as compared to the HEED[2] protocol. As compared to LEACH [3] our domatic partition based approach improves only marginally in time overhead.

3.6.3 Comparison of *energy* overhead in clusterhead rotation setup

The clusterhead rotation involves setup for re-configuration (also known as re-clustering) to consider the energy overheads involved to identify new set of clusterhead nodes in re-clustering. Besides the re-configuration setup, the periodicity of clusterhead rotation is an another important issue which adds to energy overheads, which is discussed in detail later in section-3.6.4 but used in our simulations. We compare the energy overhead of our domatic partition based clustering technique with those of LEACH, HEED and MCCP. Our domatic partition approach only needs to communicate control messages. Other clustering protocols need to identify a new set of clusterheads by invoking some leader election process. We assume that for re-clustering, node spends energy E_L in cluster setup by communicating control messages $l_{control}$ to its neighbors. The number of setup-iterations to reconfigure the clusters varies differently in clustering protocols. In HEED [2], the iterations varies up to maximum of 12. For LEACH [3, 29] it is of single iteration. A non-overlapping cluster partitioning of network is assumed for experiments. The simulation results of energy overhead for clusterhead rotation is given in figure-3.10.

From comparison of energy overheads in clusterhead rotation given in figure-3.10, we observe that our domatic partition based clustering approach overcomes the energy overhead in clusterhead rotation by about 86% as compared to HEED [2].As compared to LEACH [3] our domatic partition based approach improves only marginally for reducing energy overhead.

3.6.4 *Corrected* network lifetime

The aim of lifetime comparison is put up in the form of following question: Whether amortizing all the overheads in re-organization and self organization setup over the steady state phase results in an improved network lifetime achievable by our algorithm compared to other clustering protocols. The energy consumed by sensor nodes comprises of three components:

1. Energy initially spent in self organization,
2. Energy spent in steady state operation and
3. Energy spent in periodic re-organization.

The network operates in rounds. In each round N_{TDMA} frames are processed by each node. The notion of network lifetime is the number of simulation rounds completed until certain number of nodes die out due to energy exhaustion. Clusterhead rotation is performed after around 20sec of normal operations of rounds which is computed based on initial energy of node $0.08\frac{E_{battery}}{0.009}$ [3]. We assume a simple energy model which depletes its energy linearly used in [3, 29, 2] and all the nodes have uniform energy $2J$ at the start. The normal mode of operation continues until a node dies. After the node starts dying, re-clustering is done and normal mode of operation continues. We do not ensure the network connectivity as the nodes die out in simulation to compare the network lifetime of our technique with which we have compared our results. We assume that a node during its lifetime plays a role of either the clusterhead or non-clusterhead. The nodes in clusterhead role spend energy E_L to receive all the data signals l_{data} from the non-clusterhead nodes, spend energy E_{agg} to aggregate the signals and spend energy to communicate the aggregate data to the base station $d^2_{\mathrm{multihop-to-BS}}$ or $d^4_{\mathrm{CH-to-BS}}$ using pathloss propagation (Friss $\epsilon_{\mathrm{friss}}$ or two-ray $\epsilon_{2-\mathrm{ray}}$). Whereas the non-clusterhead node spends energy E_L to send the sensed data signal l_{data} to the clusterhead. This accounts for energy usage of clusterhead and non-clusterhead nodes in handling a single frame processing. The total energy drained from each node depends on number of rounds completed in a clusterhead role and the rounds completed as non-clusterhead. Thus, a node becomes clusterhead more than once during its lifetime.

While evaluating network lifetime we consider two aspects: $i)$ the total number of rounds completed (in simulation) until certain number of nodes dies of their complete energy exhaustion and $ii)$ the total number of messages communicated to the base station until certain number of node dies. They are called corrected temporal lifetime (described in section-3.6.4) and corrected capacity lifetime (described in section-3.6.4) respectively.

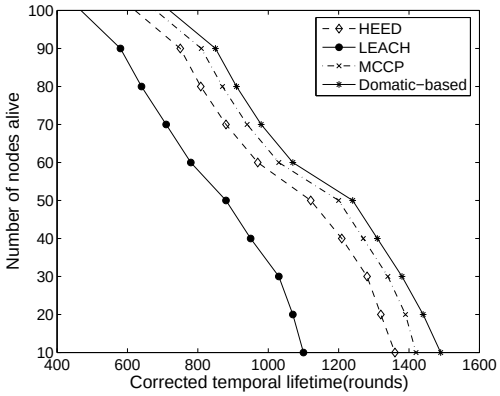


Figure 3.11: Performance comparison with *corrected* temporal lifetime

Corrected *temporal* **lifetime**

The corrected temporal lifetime refers to the number of communication rounds executed until certain number of node dies of its complete energy exhaustion together with an additional energy overhead in rotation of clusterheads at the end of each round.

We conducted a simulation experiments for comparing the corrected temporal lifetime using method described in section-3.6.4 and the results are shown in figure-3.11. We compared the *corrected* temporal lifetime of our domatic partition based clustering with the following clustering protocols: MCCP[6], HEED[2] and LEACH[3]. We observe that clustering based on domatic partition improves temporal lifetime compared to all these clustering schemes. The main reason of improving over these clustering scheme is due to identifying clusterhead based on optimized dominating set and non-clusterhead adapting to join to the nearest clusterhead to minimize intra cluster communications. Similarly, since multi-hop communication is used for inter cluster communication to base station, therefore minimum-hop path with in-network aggregation optimizes the inter cluster communication. Most importantly, our domatic partition spend very less energy in clusterhead rotation as compared to clustering schemes which gives the last mile energy gains to improve corrected temporal lifetime of network.

Corrected *capacity* **lifetime**

The corrected capacity lifetime refers to the total number of data messages received at the base station until a certain number of node dies of its complete energy exhaustion together with an additional energy overhead in rotation of clusterheads at the end of each round.

We conducted a simulation experiments for comparing the corrected capacity life-

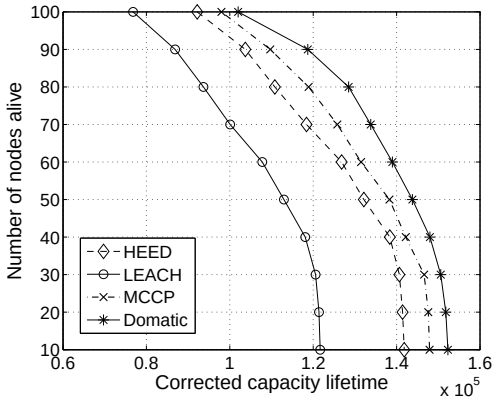


Figure 3.12: Comparison of corrected capacity lifetime

time using method described in 3.6.4 and the results are shown in figure-3.12. We assume a spanning tree rooted at base station to enable a successful inter cluster communication in our simulation. We compared *corrected* capacity lifetime of our technique with the following clustering protocols: MCCP[6], HEED[2] and LEACH[3]. We observe that our technique improves the capacity lifetime compared to these schemes.

3.7 Summary

In this chapter we have developed a new self organizing domatic partition algorithm for sensor networks. When this scheme is used as clustering protocol for sensor networks, it gives time and energy efficiency in clusterhead rotation for maximum lifetime problem. We have implemented the distributed algorithm for self organizing domatic partition and established its correctness for sensor networks. The approximation factor of our domatic partition is atleast 1/16 of the domatic number. The simulation results demonstrate the efficiency of clusterhead rotation measured in terms of energy and time overheads reduction and improving the lifetime of sensor networks. We demonstrated that our self organizing domatic partition has achieved the advantage of prolonging network lifetime even accounting for the overheads and without sacrificing the quality. This work represents a new approach of efficient clusterhead rotation scheme using self organizing domatic partition for sensor networks which may be applicable for other applications.

Chapter 4

Rotation of CDS via Connected Domatic Partition

Wireless ad hoc and sensor networks (WSN) often require connected dominating set (CDS) as the underlying virtual backbone for efficient routing. Nodes in CDS have extra computation and communication load for their role as dominator, subjecting them to an early exhaustion of their battery. A simple mechanism to address this problem is to switch from one CDS to another fresh CDS, rotating the active CDS through a disjoint set of CDSes. This gives rise to the connected domatic partition (CDP) problem which essentially involves partitioning the nodes $V(G)$ of a graph G into node disjoint CDSes. We have developed a distributed algorithm for constructing the CDP using our maximal independent set (MIS) based proximity heuristics which depends only on connectivity information and does not rely on geographic or geometric information. We show that the size of a CDP that is identified by our algorithm is at least $\left\lfloor \frac{\delta+1}{\beta(c+1)} \right\rfloor - f$, where δ is the minimum node degree of G, $\beta \leq 2$ and $c \leq 11$ is a constant for a UDG, the expected value of f is $\epsilon\delta|V|$ where $\epsilon \ll 1$ is a positive constant and $\delta \geq 48$. Results of varied testing of our algorithm are positive even for a network of large number of sensor nodes. Our scheme also performs better than other related techniques, such as the ID based scheme.

4.1 Introduction

A wireless sensor network is an ad hoc network which has no fixed infrastructure. Nodes in a wireless sensor network are battery powered and communicate either through single or multiple hops. These networks are quick to deploy in inaccessible geographic regions. Generally, a large number of sensor nodes are dispersed in the area targeted for monitoring [45]. One possible way of deployment could be that sensor nodes are dropped from an unmanned aircraft forming a deeply dense ad hoc network with much more redundancy over the optimal number of sensor nodes required for area coverage. The problem with wireless sensor network nodes is that they are operated by batteries which generally cannot be replenished, thus limiting their lifetime. Protocol design to extend the battery lifetime and optimize power consumption is, therefore an important objective. In ad hoc networks, a virtual backbone [46, 47] can be formed by constructing connected dominating set (CDS) for efficient routing. In general, a CDS of a graph $G = (V, E)$ is a dominating set $V^{'} \subseteq V$ such that each node in $V - V^{'}$is adjacent to some dominator node in $V^{'}$and the subgraph induced by dominating set $V^{'}$ is connected.

The nodes in a CDS have an extra load of computation and communication, thereby depleting their energy resources faster than other nodes of network. That is why it becomes desirable to switch to a fresh CDS from time to time. It has been shown that battery performance can be greatly improved by using pulsed discharge instead of constant discharge [48, 27]. A simple mechanism to combine load balancing and rest times for lifetime extension would be to schedule the CDS rotation periodically. Besides load balancing, the rotation of CDS breaks the continuous operation of high battery discharge by introducing a rest time to allow recharge recovery effect in electrochemical batteries in extending the battery lifetime. This is a motivation in identifying a high cardinality node disjoint CDS partitioning of sensor networks for energy conservation. This kind of problem has been treated in graph theory as the connected domatic partition (CDP) problem. In line with that, we define the connected domatic partition (CDP) of a graph $G = (V, E)$ as a partition of the vertex set V, into node disjoint connected dominating sets (CDS). The problem of finding connected domatic partition (CDP) of size $\mathrm{CDP}(G) \geq 2$ is NP-complete [33]. Therefore, we shall restrict ourselves to finding an approximate solution to the CDP problem.

A related problem is the domatic partition problem in which disjoint dominating sets of a graph are identified which are used in sensor networks to schedule dominating set for achieving energy conservation in data gathering application [1]. A seminal work of Feige [7] on approximation guarantees for domatic partition algorithm has motivated research in this area leading to work on the domatic partition problem reported in [1, 8, 49]. However none of them consider algorithmic aspects of CDP. One method to identify CDP could be to extend domatic partition to CDP. Alternatively, one could

consider extending CDS to CDP. We analyze these approaches to identify properties required for constructing a distributed CDP algorithm. Some domatic partition schemes are based on clique partitioning of the underlying graph [1, 7, 25] using global coordinates. Energy constraints in sensor networks limit the use of global positioning system (GPS) in the nodes, therefore protocols relying only on connectivity information are more useful. Computing clique partition using only connectivity information is difficult [1] in a distributed framework. As an alternative to clique partition, the domatic partition approach in [1] has defined uniform partition with bounded density property for computing k-domatic partition ($k \geq 2$), where k is the length of shortest path measured by counting the number of edges in the path in G. The approach in [1] has not considered 1-domatic partition (or domatic partition), which is desired for solving CDP problem. The problem of extending domatic partition to CDP is not considered in any of the schemes in [1, 7, 8, 49]. None of the reported works on CDS construction have tried to address the CDP problem. A naive approach would be invoking CDS algorithm multiple times to identify disjoint CDS in a graph. But this approach would loose out on the objective of partitioning graph into a large disjoint CDS, as the main objective of CDS is to identify minimum size CDS. Besides this, the other points against multiple invocation of CDS algorithm is a higher latency to form disjoint CDS which may not be desired for applications.

The acceptability of a CDP construction technique depends much on its construction efficiency. The efficiency of a CDP algorithm is based on collecting network wide information which can be further used in construction of the CDP with reduced communication. The overhead of CDP construction should be much less than invoking the best known CDS algorithm multiple times. The message complexity of CDP is an important performance metric to ascertain the construction efficiency. Latency of identifying disjoint CDS should be reduced using the CDP method as compared of invoking the CDS algorithm multiple times. The time complexity of CDP algorithm gives a measure to judge the effectiveness of CDP method compared to multiple invocation of CDS.

We have addressed here the development of a distributed algorithm for the connected domatic partition (CDP) problem for unit disk graphs (UDG) without relying on geometric or geographic information thereby using connectivity information only. To our knowledge, no such dedicated scheme for CDP is reported in literature.

The rest of the chapter is organized as follows: section 4.2 is devoted to the background from graph theory and related work. Section 4.3 defines the problem for study and the contributions of this work. In section 4.4 we describe our preliminary schemes and results. Our algorithm for connected domatic partition is presented in 4.5. Section 4.6 contains as analysis of our algorithm. Experimental results are given in section 4.7. The chapter is concluded in section 4.8.

4.2 Background and related work

In this section we mainly define terms to be used later. A common model for modeling communication between sensor nodes is the *connectivity model*, where a node u is able to determine that a node v is adjacent to it only if v is within the transmission radius of u. Node u cannot determine its exact distance from v. This is the model we are going to use.

A *dominating set* in a graph $G = (V, E)$ is a subset S of the vertex set V such that every vertex in $V - S$ is adjacent to a vertex in S. A minimal dominating set of a graph is a dominating set which ceases to be a dominating set if any vertex is removed from it. A *minimum dominating set* is a dominating set having minimum cardinality.

The *domatic number* of a graph G, denoted by $\mathrm{dN}(G)$ here, is the maximum number k such that V can be partitioned into at most k dominating sets. The *domatic number problem* is to decide for a graph G and a constant k whether $\mathrm{dN}(G) \geq k$. The *domatic partition problem* is to partition the vertices of G into $\mathrm{dN}(G)$ disjoint dominating sets. In general, domatic partition problem is NP-hard [8], whereas domatic number problem is NP-complete for $\mathrm{dN}(G) \geq 3$. It is known that domatic partition problem is bounded upper by minimum node degree of a graph plus one for general graphs. Graphs, for which domatic number is equal to minimum node degree δ of a graph plus one, i.e. $\mathrm{dN}(G) = \delta + 1$ are called *domatically full*. For example strongly chordal (SC) graphs [8], interval graphs, complete graphs, complement of a complete graph, trees and maximal outer planar graphs are domatically full.

A CDS of a graph induces a spanning tree of that graph where the nodes of the CDS are exactly the internal nodes of the spanning tree. Among all connected dominating sets of a graph, the one with minimum cardinality is called *minimum connected dominating set* (MCDS) [31]. Computing an MCDS in a unit disk graph is NP-hard [50].

A *connected domatic partition* (CDP) of a graph G, is a partition of the vertex set V, into disjoint dominating sets such that the subgraph induced by each dominating set is a connected subgraph of G. If a graph G has connectivity κ, then $|\mathrm{CDP}(G)| \leq \kappa$ [32, 33], giving us an upper bound on the size of the CDP of a graph. A graph G has the *connected domatic fullness* property if the size of its CDP is equal to κ, where κ is the connectivity of G [32, 33]. The *neighbourhood* $\mathrm{N}(v)$ of a vertex v is the set of vertices u ($u \in V(G)$) such that u is adjacent to v. In a given graph G, two vertices are *independent* if they are not adjacent. For a vertex v, the *independent neighbours* of v is $P \subseteq \mathrm{N}(v)$, such that if $v_1, v_2 \in P$, then v_1 and v_2 are independent.

The *distance-2 neighbourhood* of v is denoted as $\mathrm{N}_2(v) = \{u : 0 < \mathrm{dist}(u, v) \leq 2\}$ where $\mathrm{dist}(u, v)$ denotes the length of shortest uv-path in G measured by counting

number of edges in the path. For a vertex v, the *distance-2 independent neighbours* of v is $P_2 \subseteq \{\mathrm{N}_2(v) - \mathrm{N}(v)\}$, such that if $v_1, v_2 \in P_2$, then v_1 and v_2 are independent. An independent set $I \subseteq V(G)$ is a *maximal independent set* (MIS) I of G if each node $v \in (V(G) - I)$ is adjacent to a node $u \in I$. An MIS is also called an *independent dominating set*. A graph $G = (V, E)$ is a *unit disk graph* (UDG) if there exists $\Phi : V \mapsto \mathbb{R}^2$ satisfying $(v_i, v_j) \in E$ if and only if $\parallel \Phi(v_i) - \Phi(v_j) \parallel_2 \leq 1$, where Φ is called a realization of G [24]. The UDG has the following properties [51]:

1. For a vertex v, the size of an MIS induced on $\mathrm{N}(v)$ is at most 5.
2. A UDG G with maximum node degree Δ, contains a clique of size at least $\lceil \Delta/6 \rceil + 1$.

Identification of an MIS and a domatic partition in UDGs are NP-Hard problems [24]. By corollary, identification of a CDP in UDGs is also a NP-Hard problem. Therefore, approximation algorithms for domatic partition is of great interest. The best known ϵ-approximation scheme for domatic partition has been reported in [7] for $\epsilon = \frac{1}{O(\lg \Delta)}$, where Δ is the maximum degree of a node in G. Later Moscibroda et al reported a randomised algorithm for the domatic partition (DP) [8] problem which ensures that the size of DP obtained by their algorithm is within a factor of $O(\log n)$ of the size of maximum size of the DP, with high probability.

Pemmaraju and Pirwani in [1] reported k-domatic partition algorithms for several communication models in ad hoc networks. We summarize their technique for computing k-domatic partition (for $k \geq 2$) in the connectivity model. The k-domatic partition of G is a partition $D = \{D_1, D_2, \ldots, D_p\}$ of $V(G)$ such that each element D_i of D is a k-dominating set. The k-dominating set of G is a subset $D_i \subseteq V(G)$ such that each $v \in V(G)$ is either in D_i or has a k-neighbour in D_i. The k-neighbourhood of v is $\mathrm{N}_k(v) = \{u : 0 < \mathrm{dist}(u, v) \leq k\}$ where $\mathrm{dist}(u, v)$ denote the length of shortest uv-path in G measured by counting number of edges in path. For computing the k-domatic partition ($k \geq 2$) the scheme in [1] computes the bounded size partitions of the underlying graph using MIS and employs ID aware optimization. For an integer q_i, each partition assigns colors $r = \{1, \ldots, q_i\}$ to each of its member of D_i according to ID based ranking. For a particular color in r, the nodes from each partition forms a k-dominating set and the collection of disjoint k-dominating set is k-domatic partition.

A standard mechanism to compute a CDP, is to compute the clique partition first and then from the clique partition obtain a CDP. By clique partition, we mean partitioning of $V(G)$ so that for each partition any pair of nodes is at most at a distance 1 (i.e max transmission radius). Using only connectivity information it is difficult to get a clique partition directly [1]. The alternative is to use the existing k-domatic partition scheme reported in [1]. But the existing k-domatic partition scheme has the following drawbacks:

1. The k-domatic partition (for $k \geq 2$) scheme in [1] is unable to find 1-domatic partition ($k = 1$) for some graph models. Therefore, the problem of domatic partitioning using connectivity information is still open.

2. The clique partitioning using connectivity information is difficult to compute as stated in [1].

3. The ID based optimization used k-domatic partitioning (for $k \geq 2$) scheme in [1] may result to a poor quality due to lack of distance information.

Thus, extending the existing schemes for domatic partition to CDP is non-trivial as the existing k-domatic partition schemes (for $k \geq 2$) [1] do not handle the 1-domatic partition problem. In other words, a 1-domatic partition for $k = 1$, using only connectivity information (or domatic partition) is not computed by the scheme in [1], which is of our interest in this work.

4.3 Formulation of the problem and contributions

We assume an instance of ad hoc network with n nodes settled in ground and use its mounted omni-directional antenna to communicate in a region of maximum radius R. The footprint of the ad hoc sensor network becomes a unit disk graph $G = (V, E)$, where each vertex corresponds to a sensor node and a pair of nodes is connected by an edge if their distance is at most R. A single CDS can operate only for a limited time draining the energy of the CDS nodes earlier than rest of nodes in G which are not in the CDS. Hence, this leads to the maximum CDS lifetime problem which deals with the partitioning of G into maximum number of disjoint CDSes to enable the network to rotate the CDS roles of the nodes by switching from one CDS to another from time to time preventing any single CDS from early exhaustion of its energy. This results in maximizing the CDS lifetime and also the network lifetime.

In the rest of this section, first we give the formal definition of the problem statement and then we summarize the contributions of this chapter.

4.3.1 Problem statement

We now formally define the maximum connected domatic partition problem. For a graph $G = (V, E)$, find a partition $P = \{P_1, P_2, \ldots, P_t\}$ of $V(G)$ of maximum size such that each element $P_i \in P$ in the partition P is a connected dominating set (CDS) of G. It may be noted that a CDS P_i of G is a subset $P_i \subseteq V(G)$ such that each node

$v \in V(G)$ is either in P_i or has a neighbour in P_i and the subgraph induced by P_i is a connected subgraph of G. The problem of finding maximum connected domatic partition is NP-complete, therefore developing a distributed algorithm for connected domatic partition which approximates to maximum size connected domatic partition is the main objective of this work.

4.3.2 Contributions

The contribution of this work is summarized as follows:

1. A new *proximity* heuristics based on identifying the maximal independent set in a unit disk is used to estimate node proximity using only connectivity information without relying on geometric or geographical information. The proximity heuristic has been used for node *ranking* for CDP identification in our CDP algorithm.

2. Another contribution of this work is a new *proximity aware cluster* partitioning. Using an initial CDS, the proximity aware cluster partitioning identifies clusters of a minimum size by affiliating nodes based on their proximity values.

3. The most interesting contribution of this chapter is a new distributed construction algorithm for the maximum connected domatic partition which identifies connected domatic partitions of G. We show that the size of a CDP identified is at least $\left\lfloor \frac{\delta+1}{\beta(c+1)} \right\rfloor - f$, where δ is the minimum node degree of G, $\beta \leq 2$ and $c \leq 11$ is a constant for a UDG, the expected value of f is $\epsilon\delta|V|$ where $\epsilon \ll 1$ is a positive constant and $\delta \geq 48$.

   Our algorithm has time complexity of $O(n\delta)$, message complexity of $O(n\delta^2)$ messages and $O(\delta)$ rounds, where δ is minimum node degree of G.

4. Finally, the distributed CDS switching operation in CDP is shown to be a *local* operation, often involving a single message exchange (at most distance-2) from each node in the current CDS to communicate to the corresponding nodes in the new CDS directly.

Using our CDP algorithm we have shown that network life can be prolonged by way of $i)$ switching between CDS roles among all nodes by CDS rotation through CDP and $ii)$ taking advantage of battery recharge recovery effect by introducing rest times during CDS rotation.

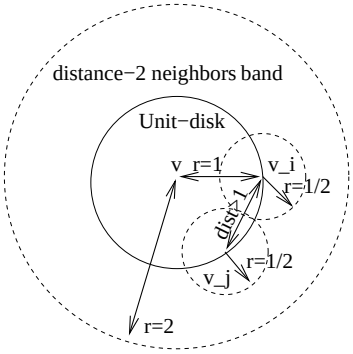


Figure 4.1: Maximum number of distance-2 independent neighbours of any node in UDG

4.4 Preliminary schemes and results

In this section we describe our basic techniques and present lemmas for use in subsequent sections.

4.4.1 Maximum size of an independent set in the halo of a node

The maximum size of independent set (IS) in $\mathrm{N}_2(v) - \mathrm{N}(v)$ for any node v, is a constant c in G. The constant c is used in our CDP algorithm, which is given in the following lemma-4.1.

Lemma 4.1 *For any node v in a UDG, the size of an independent set in $\mathrm{N}_2(v) - \mathrm{N}(v)$ is a constant $c \leq 11$, which is the distance-2 neighbours of a vertex v excluding its distance-one neighbours.*

Proof: Consider a vertex v of a UDG G. The distance-2 independent neighbours V' of vertex v are in the band between {unit disk $(v, 2)$ and unit disk $(v, 1)$}, where unit disk $(v, 2)$ is a disk centered at v of radius 2 and unit disk $(v, 1)$ is a disk centered at v of radius 1. Consider any IS of $\mathrm{N}_2(v) - \mathrm{N}(v)$. Since the vertices are independent, the distance-2 independent neighbours v_i, v_j form a packing of unit disk $(v, 1/2)$ in the region lying between unit disk $(v, 2)$ and unit disk $(v, 1)$ centered around v shown as in figure-4.1. Thus maximum packing density by unit disk $(v, 1/2)$ is given as: $c_2 = \frac{\pi(2)^2}{\pi(1/2)^2} = 16$ in $\mathrm{N}_2(v)$. Any node v can have at most 5 independent neighbours in $\mathrm{N}(v)$ [13, 31]. Therefore, for the region $\mathrm{N}_2(v) - \mathrm{N}(v)$, the number of distance-2 independent neighbours of any vertex v is at most $c = c_2 - 5 = 16 - 5 = 11$. Thus, $c \leq 11$ for UDGs. □

4.4.2 Maximal Independent Set (MIS) based Proximity Heuristics

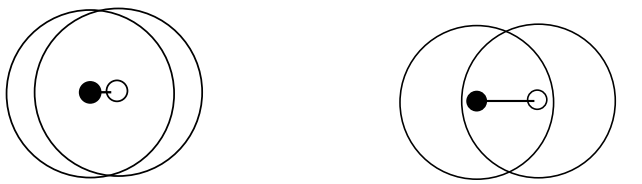
Generally, nodes in a sensor network do not have access to either geographical or geometric information for estimating actual distances. Using only connectivity information nodes have to estimate their relative proximity. We give a simple mechanism based on the notion of maximal independent sets (MIS) to compute the relative proximity of each node v_i in $\mathrm{N}(v)$ to node v, where $\mathrm{N}(v)$ is the set of adjacent nodes of v in G called as the open neighbourhood of v. Consider a node v_i in $\mathrm{N}(v)$.The node v_i computes the intersection of its open neighbourhood $\mathrm{N}(v_i)$ with $\mathrm{N}(v)$. For $1 \leq i \leq |\mathrm{N}(v)|$, each $v_i \in \mathrm{N}(v)$ computes $\mathrm{proximity}(v, v_i)$=$|\mathrm{MIS}(\mathrm{N}(v_i) \cap \mathrm{N}(v))|$. From the property of UDGs, we know that size of the MIS induced on neighbourhood $\mathrm{N}(v)$ of any vertex v is a constant 5 [51]. Therefore, set of independent neighbours in $\{\mathrm{N}(v) \cap \mathrm{N}(v_i)\}$ approximates area of overlapping of unit disk centered at v_i to V_i as shown in figures-4.2(a) and 4.2(b).

Algorithm 4 Proximity ranking

Input: Cluster V_u with clusterhead u.

Output: Ranking of nodes in $\mathrm{N}(u)$ based on relative proximity

1: Let V_u be the cluster dominated by node u.
2: Let $\mathrm{N}(v)$ be the neighbourhood of cluster member $v \in V_u$.
3: Node u sends out message m_1 with the information of V_u to its neighbours.
4: Node v on receiving V_u through m_1 computes the intersection set $S_v = \mathrm{MIS}(V_u \cap \mathrm{N}(v))$. Note that MIS of intersection approximates the area of overlapping.
5: Node v sends S_v though m_2 to dominator u.
6: Node u on receiving m_2 from all its members computes its relative proximity as $proximity = |S_v|$ ($\forall v \in V_u$).
7: Node u ranks each node v in V_u based on non-increasing ordering using proximity $|S_v|$. In case of ties, node IDs are used to resolve for total ordering.



(a) Bigger corona for the close neighbours

(b) Smaller corona for distant neighbours

Figure 4.2: Overlapping of unit disk areas of neighbours in UDG

4.4.3 Computing proximity based ranking

A number of good cluster partitioning schemes have been reported in the literature [52]. However, the problem CDP has remained an open problem. We assume that some clustering technique [52] is used to decompose the network into clusters of cluster partitioning. We now formulate a node ranking scheme among the nodes in the clusters of a cluster partition of G to facilitate the CDP construction. This ranking scheme works using the MIS based proximity heuristics discussed in section-(4.4.2) which uses only connectivity information and does not rely on geographic or geometric information. In this section we present a simple mechanism to obtain the proximity ranking of cluster members. Let V_i be a cluster in the cluster partition and let its dominator node $v_i \in I = (MIS(G))$ dominate all the vertices in V_i.

A vertex $u_i \in V_i$ computes its MIS in $V_i \cap \mathrm{N}(u_i)$. The, set of maximal independent neighbours in $V_i \cap \mathrm{N}(u_i)$ approximates area of overlap of the unit disk centered at v_i to that centered at u_i, as shown in figure-4.2. The quantity $|\mathrm{MIS}((V_i \cap \mathrm{N}(u_i)))|$, clearly approximates relative proximity of node $u_i \in V_i$ with respect to dominator v_i of V_i. The node u_i broadcasts its proximity value to its dominator v_i. The dominator v_i after receiving the proximity values of all nodes in V_i, computes the proximity ranking. In case of ties, nodes can use node IDs to resolve for total ordering. The algorithm for proximity ranking is now given.

4.4.4 Proximity aware cluster partitioning

Assume that we have a CDS $S = \{I \cup C\}$ of G, where I is the set of dominators and C is the set of connectors, which can be formed using some CDS algorithm in $O(n)$ time and $O(n\Delta)$ messages [14]. We now give a simple mechanism to compute the cluster partitioning P of G such that the size of each cluster has a constant lower bound of $\lfloor \frac{\delta}{c+1} \rfloor$.

Step-1 A vertex $v \in V - \{I, C\}$ is affiliated with a cluster dominated by u of I, if v is dominated only by u.

Step-2 If a vertex v is adjacent to multiple vertices of I, then v is affiliated to that $u \in I$ whose cluster size is currently less than $\lfloor \frac{\delta}{c+1} \rfloor$, if such a $u \in I$ exists.

Step-3 Otherwise, v is affiliated to the closest vertex $u \in I$, as determined by the proximity heuristic.

Lemma 4.2 *The size of any cluster obtained by algorithm-5 is at least* $\left\lfloor \frac{\delta}{(c+1)} \right\rfloor$, *where* δ *is minimum vertex degree of graph* G *and* c *is a constant* $c \leq 11$.

Proof: The bound on c ($c \leq 11$) comes from lemma-4.1. Follows from the given construction procedure of the cluster partitions. In particular, steps for rule-2 and rule-3 of algorithm-5 ensure that lower bound of $\left\lfloor \frac{\delta}{(c+1)} \right\rfloor$ is always met. □

Algorithm 5 Proximity aware cluster partitioning

Input: CDS $S = \{I \cup C\}$ of G and lower bound size=$\lfloor \frac{\delta}{c+1} \rfloor$
Output: $P = \{V_1, \ldots, V_{|I|}\}$ as the partition of $V(G)$

1: Let each $u \in I$ send message m_1
2: Nodes $v \in \{V(G) - I\}$ receive m_1 from u and record u in W_v. (* Thus $W_v = \{I \cap \mathrm{N}(v)\}$ is the set of neighbouring dominators. *)
3: Node v sends message m_2 containing W_v to $u \in W_v$.
4: Node $u \in I$ receives m_2 from neighbouring node v.
5: Initialize $V_u = \perp$.
6: **for all** $u \in I$ **do**
7: *Rule-1:Include in cluster V_u nodes covered only by u.*
8: **for all** $(\{v \in \mathrm{N}(u)\}) \wedge (|W_v| = 1)$ **do**
9: $V_u = V_u \cup \{v\}$
10: **end for**
Rule-2: Include v in cluster V_u if closest to u and $|V_u| < \lfloor \frac{\delta}{c+1} \rfloor$
11: **for all** $(\{v \in \mathrm{N}(u)\}) \wedge (|W_v| > 1) \wedge (|V_u| < \lfloor \frac{\delta}{c+1} \rfloor)$ **do**
12: **if** u is the closest dominator of v **then**
13: $V_u = V_u \cup \{v\}$
14: **end if**
15: **end for**
Rule-3: Include v in cluster V_u if $|V_u| < \lfloor \frac{\delta}{c+1} \rfloor$
16: **for all** $(\{v \in \mathrm{N}(u)\}) \wedge (|W_v| > 1) \wedge (|V_u| < \lfloor \frac{\delta}{c+1} \rfloor)$ **do**
17: $V_u = V_u \cup \{v\}$
18: **end for**
Rule-4: Include v in cluster V_u arbitrarily to some of its dominator if $|V_u| \geq \lfloor \frac{\delta}{c+1} \rfloor$
19: **for all** $(\{v \in \mathrm{N}(u)\}) \wedge (|W_v| > 1) \wedge (|V_u| \geq \lfloor \frac{\delta}{c+1} \rfloor)$ **do**
20: $V_u = V_u \cup \{v\}$
21: **end for**
22: **end for**

4.5 Algorithm for Connected Domatic Partition

In this section we describe our distributed construction algorithm to find a large size CDP of a graph G. Let each sensor node have a unique ID. Assume that each node is

aware of its distance-1 neighbours. This can be done in $O(\Delta)$ time. We also assume a fixed node as the leader node. For example a base station can be leader node in sensor networks or leader can be found using some leader election algorithm in $O(n)$ time with $O(n \log n)$ messages [53].

Our scheme is based on proximity aware cluster partitioning, MIS based proximity heuristic and iterative matching to obtain the CDP. Now, we give a big picture of our approach to compute a large size CDP.

Steps in algorithm for CDP

Step 1 Decompose the network in cluster partitions. (using algorithm-5 given in section-4.4.4)

Step 2 Compute the proximity ranking of the cluster nodes using proximity heuristic. (using algorithm-4 given in section-4.4.3)

Step 3 Compute the CDP by growing disjoint CDS trees by iteratively matching cluster nodes in cluster partition to identify node disjoint CDS. (using algorithm-6 given in the following section-4.5.1)

4.5.1 Growing node disjoint CDS tree by iteratively matching nodes rank wise

After identifying the starting CDS and performing cluster partitioning we have a partition of the vertices of G into clusters. Each cluster contains a dominator node of the starting CDS. The cluster that contains the base station is called the leader cluster. We first pick the lowest ranked (with proximity measure) available node in the leader cluster and find matching available nodes for it in all adjacent clusters, which are picked so that it has the lowest rank among all available nodes in the concerned cluster. In this process, the node picked from the leader cluster is to begin with assigned a level of $l = 1$. All the matching nodes that were picked from the adjacent clusters are assigned a level of $l + 2$. Note that seed node in the leader cluster cannot communicate directly with matching nodes as they are distance-2 neighbours. It is, therefore, necessary to have forwarding nodes through which the seed node in the leader cluster can communicate to the matching nodes in the adjacent clusters. A minimal set of such communicating nodes are identified and these are assigned a level of $l + 1$. Through this process a new CDS S' is formed in stages, starting from the seed node in the leader cluster and expanding outwards. The matching nodes in the clusters form an MIS I' of the CDS,

while the forwarding nodes form the set of connectors $C^{'}$. Thus the new $S' = \{I' \cup C'\}$. A S' is then tested to be a CDS. If it is a CDS then it is added to the CDP, otherwise it is not and the nodes continue to remain unavailable until the post processing step.

Connectors to connect level-l and level-($l + 2$) dominators need to be identified. Each potential connector becomes aware of level-($l + 2$) dominators that it can connect and broadcasts the count of the nodes it can connect. Thus, each level-($l + 2$) dominator becomes aware of the number of dominators each adjacent potential level-($l + 1$) connector can connect. Each level-($l + 2$) dominator, then chooses the adjacent node capable of connecting the maximum number of level-($l + 2$) dominators as its connector node. Ties are broken on node IDs. This scheme is essentially an adaptation of the degree aware optimization technique [14]. This process of expansion of the CDS tree is continued until all the clusters have been visited. This process of identifying a CDS tree, starting at the leader cluster and expanding outwards is called a round of CDP formation. Identification of secondary CDSes continues until the construction of a secondary CDS fails. A post processing step is often required to handle the *extra* nodes left *available* after the completion of matching iterations. The properties of *extra* nodes hold are the following:

1. *extra nodes* are the dominatees of each CDS in CDP computed by algorithm-6.

2. *extra nodes* left after termination of matching rounds in algorithm-6 are insufficient to form a separate CDS.

3. *extra nodes* are the available set of nodes which are not in any element of CDP.

A node from the set of extra nodes may be assigned to a CDS say D_j making making that node unavailable. Note that each node $v \in V(G)$ sets B, an array of pointers maintained by each node to record its dominators in each CDS in CDP.

Now we give the outline of algorithm-6. The *outer iteration* beginning at step-6 starts a matching round to deal with finding the disjoint CDSes. Termination occurs when no more available node remains to construct more CDSes. The *inner iteration* beginning from step-9 deals with step by step construction of a CDS tree in a breadth first (BFS) order. Two levels of the CDS tree tree are constructed in the inner iteration. The current iteration terminates when either all the clusters are visited or insufficient *available* nodes remains.

Algorithm 6 Proximity aware connected domatic partition

Input: Initial CDS $D = \{I \cup C\}$

Output: Partition $P = \{D_1, D_2, \ldots\}$ of $V(G)$ as CDP(G) where D_i is a CDS of G. Each node $v \in V(G)$ gets value of i, a $parent$ pointer for its role in i^{th} CDS and $B[1, \ldots, |P|]$ an array of CDP pointers its dominator node-$id \in P$ (for $B[i] = 0$).

1: Compute cluster partitioning $\{V_1, V_2, \ldots, V_{|I|}\}$ of $V(G)$ using *algorithm*-5
2: Compute proximity ranking using *algorithm*-4 (for each cluster $V_i, (1 \leq i \leq |I|)$)
3: Let u_z be the leader node and V_z be its cluster
4: Initialize $available(v) \leftarrow 1, \forall v \in V(G) - CDS$
5: Let $i \leftarrow 0$
6: **repeat**
7: Let $i \leftarrow i + 1$ and $l \leftarrow 0$
8: Let $u_{i,l}$ be the closest available node in V_z; $available(u_{i,l}) \leftarrow 0$; $D \leftarrow \{u_{i,l}\}$
9: **repeat**
10: $l \leftarrow l + 1$
11: $u_{i,l}$ sends a message so that all available nodes in $w_{i,l+1} \in \mathrm{N}(u_{i,l})$ become its *dominatees*
12: Each $w_{i,l+1}$ node in turn sends another message to its (level-$(l+2)$) neighbours, making the *available* nodes as *potential dominators*
13: Let V_p be the cluster, such that $u_{i,l} \in V_p$ and v_p its clusterhead. Note that $v_p \in I$;
14: Let $H = I \cap \mathrm{N}_2(v_p)$ be the set of distance-2 neighbours of v_p in I
15: **for all** *V_h such that $v_h \in H$ and v_h is the clusterhead of V_h* **do**
16: Each *active* node $v_{i,l+2} \in V_h$ sends its proximity value to v_h which then identifies the closest among these (say $u_{i,l+2}$ to become a *dominator*
17: Now $available(u_{i,l+2}) \leftarrow 0$
18: Node $u_{i,l+2}$ broadcasts itself as the new dominator; $D \leftarrow D \cup \{u_{i,l+2}\}$
19: Dominatees $w_{i,l+1}$ on receiving this message keep a count of neighbouring dominators at level-$(l+2)$ and broadcasts the final count
20: **end for**
21: Each level-$(l+2)$ dominator on receiving the counts from the potential connectors, select among them the node with highest count ($w_{i,l+1}$) as its connector and informs it
22: Node $w_{i,l+1}$ then becomes a connector and sets $available(w_{i,l+1})$ to 0 ; $D \leftarrow D \cup \{w_{i,l+1})\}$
23: **until** *no cluster V_k ($1 \leq k \leq |I|$) left unvisited or insufficient available nodes left to carry out matching*
24: Nodes $v \in V(G)$ is verified i.e $v \in N[D]$ to hold the domination property for D and that subgraph of G induced by D is connected
25: If the test fails then D is not included in P, otherwise it is included
26: **until** *no available nodes left in V_z (matching completed) or insufficient available nodes in the cluster to carry out matching*
27: Post processing step distributes extra nodes arbitrarily to some CDS of CDP (still holds to be CDS).
28: Each node set its dominators for each CDS in CDP, in an array B maintained by each node $v \in V(G)$ (except the CDS in CDP where the node itself is the dominator).

Algorithm 7 Adaptive rotation of CDS

Input: CDP partition P=$\{D_1, \ldots, D_{|P|}\}$ of $V(G)$, *dom* pointer array variable of CDS trees of each node (computed in algorithm-6), recharge recovery time τ_{rr} and scheduling time $\tau_s = \tau_{rr}$

Output: Rotation of CDS after every τ_s time.

1: Compute the CDS activation time τ_{active}=τ_s.
2: t =*current-time*
3: **for all** $i \leftarrow 1, |CDP|$ **do**
4: Schedule time for $D_i \leftarrow t + i \times \tau_{active}$
5: **end for**
6: Switching from D_i to D_j takes place through local message based transition from dominators in D_i to D_j
7: **for all** $k \leftarrow 1, |D_i|$ **do**
8: Dominator node $u_k \in D_i$ sends m_1 to its neighbouring dominator $dom_j(u_k)$
9: Nodes $p_k \in D_j$ on receiving m_1 switched to become dominator in j^{th}-CDS tree and sends out m_2 to the dominator node of new CDS.
10: Connectors $c_k \in D_j$ among its neighbouring nodes are activated on receiving m_2 and sends out m_3 message to its independent dominators.
11: Any *remaining dominator nodes* are also switched to dominator in D_j on receiving m_3.
12: **end for**

4.5.2 Rotation of CDS via local switching

We now present a simple algorithm for switching from one CDS to another. This algorithm is given to highlight the simplicity of switching between CDSes locally

We now give our elementary rotation algorithm to highlight two important benefits derived from our CDP construction algorithm, namely: $i)$ load balancing and $ii)$ local switching between CDSes, which may be effected using an exchange of only a single message transfer by each node of current CDS to activate the nodes of new CDS in its close vicinity. The rotation of CDS involves two main issues: $i)$ distributed switching and $ii)$ scheduling time.

Note that to switch from dominators in D_i to D_j, nodes in D_i can reach neighbours in D_j at a distance of 1 through 1-hop messages and the remaining nodes in D_j can be activated by connector nodes in D_j. Thus, rotation becomes an efficient local distributed switching process.

The scheduling time depends on battery parameters of recharge recovery time τ_{rr}, if the size of the CDP is more than one. This allows a pulsed discharge in battery

to prevent from a long continuous battery discharge in any node of network. A rest time is introduced to enable the charge recovery in the electrochemical battery known as recharge recovery effect [48, 27]. Another aspect of scheduling rotation of CDS is the load balancing. The aim is to allow any single CDS only a small fraction of time compared to total time of system operation. Thus, for a period of time T, each CDS in the CDP requires to remain active for a period $\frac{T}{|\mathrm{CDP}(G)|}$ and performs low energy tasks for the rest of the time. Therefore, larger the size of the CDP, smaller is the duration of time when a node is subjected to bear higher energy loads in the role of an active CDS node.

4.6 Analysis of CDP algorithm

In the next three subsections we first analyze our algorithm to provide the size of connected domatic partition, then analyze its complexity for running time and messages exchanged and finally, the correctness of the algorithm.

4.6.1 Size of the CDP obtained

Lemma 4.3 *Let the CDS be $D = \{I \cup C\}$, where I is the set of independent dominators in the CDS and C the set of its connectors. The minimum number of connectors $|C|$ is $\frac{|I|-1}{4}$.*

Proof: Using the construction technique in [13] we proceed as follows. Let the sequence of connectors occurring in any BFS traversal of the CDS tree be $c_1, c_2, \ldots, c_{|C|}$. Let I_1 be the set of nodes in I that are adjacent to c_1. For any $2 \leq i \leq |C|$, let I_i be the set of nodes in I that are adjacent to c_i but adjacent to any of $c_1, c_2, \ldots, c_{i-1}$. This leads to a partition of I as $I_1, I_2, \ldots, I_{|C|}$. As c_1 can be adjacent to at most five independent nodes, $|I_1| \leq 5$. For any $2 \leq i \leq |C|$, at least one node in $I_1, I_2, \ldots, I_{i-1}$ is adjacent to c_i. Thus, nodes in I_i must lie in a sector of at most 240° within the coverage range of node c_i. This implies that $|I_i| \leq 4$. Therefore, $|I| = \sum_{i=1}^{|C|} |I_i| \leq 5 + 4(|C| - 1) = 4|C| + 1$. Thus, $|C| \geq \frac{|I|-1}{4}$ □

Lemma 4.4 *The maximum number of connectors $|C|$ in a CDS $D = \{I \cup C\}$ is $|I| - 1$.*

Proof: Maximum number of connectors is required when in the CDS tree one connector connects exactly two dominator nodes in I. If E is the number of edges in the CDS

tree, then in this case $|E| = 2|C|$. Also, $|E| = |I| + |C| - 1$, so $2|C| = |I| + |C| - 1$, so $|C| = |I| - 1$. □

Theorem 4.5 *For any CDS* $D = \{I \cup C\}$, $\frac{5|I|-1}{4} \leq |D| \leq 2|I| - 1$.

Proof:

From lemma-4.3, $|D| = |I| + |C| \geq |I| + \frac{|I|-1}{4} = \frac{5|I|-1}{4}$

From lemma-4.4, $|D| = |I| + |C| \leq |I| + |I| - 1 = 2|I| - 1$ □

Theorem 4.6 *Using proximity heuristics the CDP obtained is at least of size* $\left\lfloor \frac{\delta+1}{\beta(c+1)} \right\rfloor - f$, *where* δ *is min node degree of* G, $\beta < 2$, c *is a constant such that* $c \leq 11$ *for UDGs and* f *is the number of rounds where the CDS test fails.*

Proof: Let $D = \{I \cup C\}$ be any CDS identified by algorithm-6. Let $\beta = \sup_{j \in J} \left\{ \frac{|D_j|}{|I_j|} \right\}$, where J is the set of CDSes computed by algorithm-6 on all possible problem instances. From theorem-4.5,

$$\beta = \sup_{j \in J} \left\{ x | \frac{5|I_j| - 1}{4|I_j|} \leq x \leq \frac{2|I_j| - 1}{|I_j|} \right\}$$

Clearly, $\beta < 2$. β may be thought of as the number of nodes that goes into the CDS D, for every node that goes into its set of independent dominators I.

Algorithm-6 uses the nodes from the clusters to form CDSes towards developing the CDP. By lemma-4.2, the minimum size of a cluster is $\left\lfloor \frac{\delta+1}{(c+1)} \right\rfloor$. If $f = 0$, the number of CDSes formed from it is at least $\left\lfloor \frac{\delta+1}{\beta(c+1)} \right\rfloor$. One CDS is lost for each failure. Hence, the number of CDSes framed from it is at least $\left\lfloor \frac{\delta+1}{\beta(c+1)} \right\rfloor - f$. □

On termination, algorithm-6 performs a post-processing to distribute nodes in the sets that failed the CDS-test into other CDSes as connectors. We now try to estimate the number of such failed sets. For the analysis we work with all the sets that were identified, prior to the final assimilation.

Lemma 4.7 *The expected number of sets identified by algorithm-6 that fail to be a CDS is* $\epsilon\delta|V|$ *for* $\delta \geq 48$ *and a positive constant* $\epsilon \ll 1$.

Proof: We give an estimate of the number of CDSes lost using the fact that $\left(1+\frac{t}{m}\right)^m \leq e^t, m \neq 0$. Let n be the total number of potential CDS set identified in algorithm-6, $n \geq \left\lfloor \frac{\delta}{\beta(c+1)} \right\rfloor$. Also, assume that $n \geq 2$. This requires $\delta \geq 48$. As a simplifying, but highly pessimistic assumption, let us assume that the vertices are chosen to be placed in the sets identified by algorithm-3, at random. The algorithm actually makes an effort to match up vertices in a more sensible manner using the proximity heuristic. Let $A_{v,d}$ represent the event that there is no node in $\mathrm{N}[\{v\}]$ which is in set number d. This essentially means that node v is not covered by set number d. Let α be a more optimistic constant than $\beta(c+1)$.

$$\begin{aligned}
\Pr[A_{v,d}] &= \prod_{u \in \mathrm{N}[\{v\}]} \left(1-\frac{1}{n}\right) \\
&\leq \left(1-\frac{1}{n}\right)^{\delta_v+1} \\
&\leq \left(1-\frac{\alpha}{\delta}\right)^{\delta_v+1} \\
&\leq e^{-\frac{\alpha(\delta_v+1)}{\delta}} \leq e^{-\alpha}
\end{aligned}$$

Probability that at least one node is not covered by this set is $1-(1-e^{-\alpha})^{|V|} \approx (e^{-\alpha})|V|$, assuming $e^{-\alpha} \ll 1$. This is essentially the probability that this set is not a dominating set. The expected number of sets that fail to be dominating sets is, therefore, $\left\lfloor \frac{\delta}{\beta(c+1)} \right\rfloor e^{-\alpha}|V| = \epsilon\delta|V|$, where $\epsilon \ll 1$ is a positive constant. □

The presence of $|V|$ in the expression is significant. It suggests that as the area of dispersion increases for the same density of nodes, identification of the CDSes becomes more difficult.

4.6.2 Complexity of the CDP algorithm

The complexity analysis of algorithm-6 is dominated by running time of two major nested iterations:

Inner iteration of constructing CDS This loop is dominated by visiting two clusters at a time and constructing two levels of CDS tree in BFS order. The nodes are explored one-by-one to form a CDS. Thus, the time complexity of this phase is at most $O(n)$. The message complexity is dominated by cluster size, since it may be broadcast $O(\delta)$ times by a node. Thus, the message complexity of this phase is $O(n\delta)$.

Table 4.1: Complexity of our CDP Algorithm

Time	Message	Rounds	Size
$O(n\delta)$	$O(n\delta^2)$	$O(\delta)$	$\epsilon\delta\|V\|$

Matching or outer iteration for CDP construction Each matching round of algorithm-6, except the last round, contributes a CDS to the CDP that is finally output. The maximum number of rounds the outer iterations takes is δ.

Now on combining the complexity of nested iterations i.e inner and outer iterations, we give the distributed complexity of the algorithm-6. Thus, algorithm-6 takes $O(n\delta)$ time, $O(n\delta^2)$ messages and $O(\delta)$ rounds. The complexity of algorithm-6 of our algorithm is detailed in table-4.1.

We are unaware of a similar CDP technique reported in the literature. If a single CDS construction technique is used repeatedly to construct new CDSes for rotation, then the complexity of such a technique would be similar to ours, but the important difference is that the number of CDSes obtained may be poor. The strength of our technique is the orderly construction the CDSes, so that the CDP size is maximized.

4.6.3 Correctness of the CDP algorithm

We present here the correctness proof of our CDP algorithm by proving the following three properties: $i)$ each member of the CDP constructed is a CDS, $ii)$ the CDP forms a node disjoint partition of $V(G)$ and $iii)$ the algorithm terminates.

Lemma 4.8 *Each element D_i, $(1 \leq i \leq j)$ of the CDP $P = \{D_1, D_2, \ldots, D_j\}$ computed by algorithm-6, is a CDS of G.*

Proof: Step-24 of the algorithm checks that the set of nodes D identified for the next CDS is indeed a CDS. In step-25, if D is added to P only if the test succeeds. Hence, all each member of P is a CDS. □

It is useful to note that if the available nodes in the partition happen to be placed in such a way that a CDS exists, then the algorithm is likely to find that CDS.

Lemma 4.9 *The partition CDP computed by algorithm-6 satisfies node disjointness property.*

Proof: There exists only two possibilities in matching process of algorithm to decide the membership of node as to which of the partitions in CDP a node has to go into: $i)$ matched nodes and $ii)$ extra nodes. In the matching process of algorithm-6, every round visits each cluster to look for the available nodes to contribute a CDS D_i of CDP. A Boolean flag *available* is associated with every node $v \in V(G)$, which is initialized to 1 at the beginning. As soon as a node v is matched in j^{th}-round for contribution to CDS D_j, it turns off its available flag to 0. Thereafter, in subsequent rounds the unavailable node v is not considered in matching. The Boolean flag of any node enables it to be assigned at most once in at most one set of CDP. Therefore, no matched node exists which is a member of at least two partition of P.

For extra nodes, algorithm-6 takes on a post processing step to distribute the available nodes v which are left unmatched, to some partition in $D_j \in P$ as a redundant connector. After, its allocation to the partition the flag available(v) becomes 0.

Thus, no node exists which is a member of more than one member of P.

□

Lemma 4.10 *Algorithm-6 terminates after finite number of iterations.*

Proof: Let each node $v_i \in V(G)$ initialize a Boolean flag *available* to 1 at the beginning. The running of the CDP algorithm is governed by two major iteration steps: $i)$ inner iteration for step-by-step construction of the CDS tree and $ii)$ matching iterations in the outer loop for identifying disjoint CDS trees.

The inner iteration is controlled by finding available nodes (unexplored so far) considered for matching in some CDS partition for computing CDP. This implies, that an available node which becomes unavailable at a particular inner iteration never becomes available again, until both the loops are exit. Therefore, the inner loop terminates after a finite number of iterations when all the node become unavailable or on being unable to find any available nodes.

The matching or outer iteration is also governed by the Boolean flag *available* for each node. A node is considered for matching only if it is available. The outer loop also terminates after a finite number of iterations when all the node become unavailable or on being unable to find any available nodes.

Hence the algorithm terminates after finite number of steps. □

Corollary 4.11 *Algorithm-6 computes connected domatic partition correctly.*

Proof: Follows from lemmas 4.8 to 4.10. □

4.7 Experimental results

In this section we present the results of simulating our algorithm on various types of graphs. The goals of these simulations are to determine the following: $i)$ performance of algorithm on graphs having connectivity κ with high probability $ii)$ performance of algorithm on graphs with known CDP value $iii)$ performance comparison with two related techniques and $iv)$ battery lifetime simulations.

4.7.1 Simulation of the CDP algorithm on graphs having connectivity κ with high probability

In our simulation environment, a uniform random number generator generates the x and y coordinates of n nodes to be placed on an $100 \times 100\ \mathrm{m}^2$ deployment area A. We have created links between nodes using a parameter *minimum transmission radius* denoted by r_0, so that the resulting graph has connectivity of κ with high probability. To ensure an acceptable confidence of simulation parameter r_0, we use the following results given in [54]:

> The probability for k-connectivity of a homogeneous ad hoc network: $\Pr(G \text{ is } k\text{-connected}) = \left(1 - \sum_{N=0}^{k-1} \frac{(\rho\pi r_0^2)^N}{N!} e^{-\rho\pi r_0^2}\right)^n$ with n nodes $(n \gg 1)$, each with transmission range r_o and homogeneous node density $\rho = n/A$.

In our simulations, to offset the border effect [54] we use a much higher transmission range which is required to achieve the same $\Pr(G \text{ is } k - \text{connected})$. This correction was judiciously carried out as the simulation was done on a bounded area, where as the analytical derivation assumes an infinite area. To model the wireless transmission between the nodes, a radio link model is assumed in which each node has a certain transmission range r_0 and uses omni-directional antennas. Only bi-directional links are considered. This link model corresponds to a propagation model with certain signal attenuation (path loss). Let $P_0 = P(r = 0)$ denote the transmitted signal power at the sending node and $P(r)$ the received power at a distance r from the sender. The received power falls as $P(r) \propto r^{-\gamma} P_0$, where γ is the path loss exponent, which depends on the environment (typically $2 \leq \gamma \leq 5$). The wireless transmission range r_0 can then be

mapped to the equivalent transmission power P_0 using a threshold for receiver sensitivity P_s. A node can receive properly if $P(r = r_0) \geq P_s$. Thus, we calculate transmission range r_0 that is the parameter required to obtain an almost surely k-connected network.

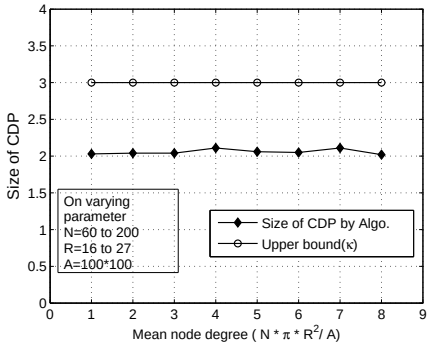


Figure 4.3: Performance of Algorithm on graphs with connectivity κ with high probability for transmission range $r \leq 27$

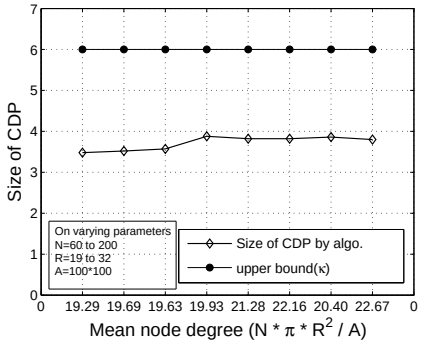


Figure 4.4: Performance of Algorithm on graphs with connectivity κ with high probability for transmission range $r \leq 32$

We have divided mean node degree into three ranges: low degree, average degree and high degree, for providing simulations to observe the effect on the size of connected domatic partition.

We have simulated the algorithm in MATLAB and PROWLER [42] an event driven simulator for sensor networks to simulate the protocol and the behavior of rotation of CDS through disjoint CDS. We considered the parameter transmission range $r \leq 27$ for obtaining low mean node degree for the network sizes ranging 20-200 to measure their effect on size of connected domatic partition. The results in figure-4.3 show a small gap in generating connected domatic partition compared to the upper bound (κ).

On increasing the transmission radius $r \leq 32$, we obtain the average size mean node degree for obtaining graph with higher connectivity κ of graph with high probability for the network. The result shown in figure-4.4 is for average size mean node degree

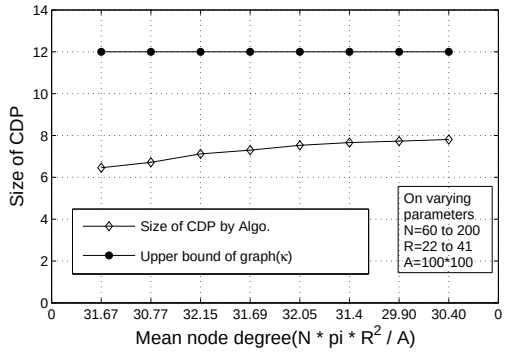


Figure 4.5: Performance of Algorithm on graphs with connectivity κ with high probability for transmission range $r \leq 41$

network density which obtains a desired connectivity of graph which gives the upper bound on the size of CDP to compare with the size of CDP identified by our algorithm.

Now considering the high mean node degree. The figure-4.5 shows the performance of our CDP algorithm on connectivity κ obtained using the mean node degree and network size on the size of CDP. Since, $|\mathrm{CDP}| \leq \kappa$. Thus, mean gap between the computed $|\mathrm{CDP}|$ and the upper bound of $|\mathrm{CDP}|$, i.e. κ is of 36% using uniformly random node distribution in a given deployment area.

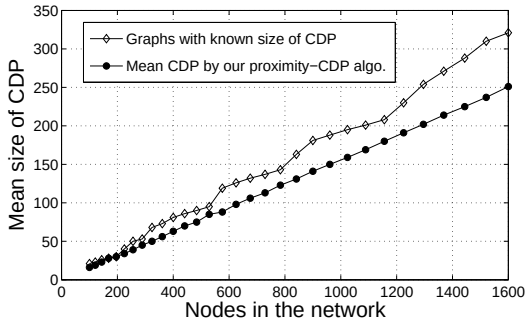


Figure 4.6: Performance of Algorithm on graphs with known CDP

4.7.2 Simulation of the CDP algorithm on graphs with known CDP

We conducted an experiment to adjudge the performance of algorithm on graphs with known sizes of CDP. In order to generate the graphs with known size of CDP, we describe a simple mechanism to generate the graphs with known CDP. Assume the maximum transmission radius r. Consider a minimum spanning tree generated on a given set of nodes with radius r. This gives only one set of CDS. For each node in the tree, identify the clique region as the circular disk of radius $r/2$ with nodes at its center.

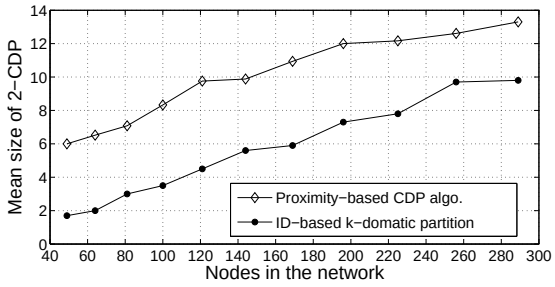


Figure 4.7: Performance comparison: Proposed 2-CDP vs (connected) k-domatic partition scheme of [1]

Place $k-1$ trees in the cliques so that each clique contains a distinct node of tree. If n_t be the set of nodes in first spanning tree and then each tree $t_1, \ldots, t_{k-1}$ of size n_t adds to give out the graph G of size $n = kn_t$. Since each tree in generated graph G is a CDS of G. Thus, the generated graph has the of size kn_t nodes and having the CDP size $|\mathrm{CDP}(G)| = k$. In order to avoid the border of cliques region, we consider placing the nodes little inside clique range, i.e say clique-radius=$r/2.5$, so that algorithm is given a proper data set. The simulation results is shown in figure-4.6 gives the performance of algorithm in identifying CDP. The results reveal that for graphs with known CDP values and the computed CDP value by algorithm gives the mean difference of 19%.

4.7.3 Performance comparison of the CDP algorithm

There is no scheme available to construct CDP using only connectivity. However, in [1] while computing k-domatic partition using GPS, they identified the connected ones among 2-domatic partition (using GPS) called it 2-(connected) domatic partition. Thus, we compare our CDP results with the scheme in [1] 2-(connected) domatic partition. In order to compare the results of our distributed CDP algorithm with 2-(connected) domatic partition scheme ($k \neq 1$) in [1], we need to modify our approach to create a 2-connected domatic partition. We simplified our algorithm to consider proximity aware cluster partition as 2-clique (2 means distance-2), so that any pair of node in partition can be at most at a distance-2 in G. The modification of our algorithm for 2-CDP includes that any pair v_1, v_2 of nodes can be called connected if $\mathrm{dist}(v_1, v_2) \leq 2$, where distance is measured as shortest distance by counting the forwarding hops. Similarly, nodes are independent if $2 < \mathrm{dist}(v_1, v_2) \leq 4$. Using, this simple extension in algorithm to compute 2-CDP, we compared of our proposed distributed algorithm for CDP with the results given in [1]. In figure-4.7, we compare the quality of connected domatic partition with the competitive scheme of [1] using the same simulation parameters. The metric of comparison is the number of disjoint connected dominating set, we observed that our approach yields mean of 20% increase in the size of connected domatic set

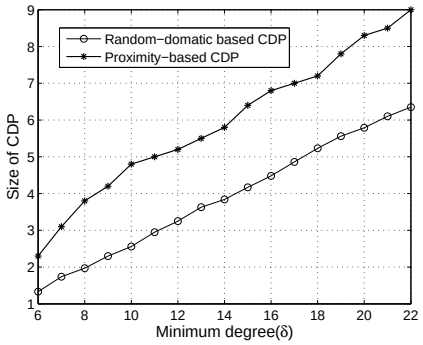


Figure 4.8: Performance comparison: Proximity based CDP vs Random domatic partition based CDP

compared to scheme in [1]. We observe that our proximity heuristics is better than ID based heuristics given in [1], which yields an improved approximation factor of connected domatic partition by 20% obtained through simulations.

4.7.4 Performance comparison with random domatic partition based scheme

We give a comparison with the randomized algorithm for domatic partitioning by Moscibroda [8]. First we present as how we have implemented the randomized algorithm for domatic partitioning by Moscibroda [8] and then describe as how we have converted their domatic partition to CDP so that it can be compared with our proximity based CDP.

A random color is assigned from the range $[1, \ldots, \delta_v/(3 \log n)]$ to the neighbours of a node v, where δ_v is minimum degree of node in neighbourhood $N[v]$. For many colors classes the network is decomposed into the potential domatic partitions of G. Following two rules are applied to decide which of these qualify for domatic partition of G: $i)$ A color class $c \in [1, \ldots, \delta_v/(3 \log n)]$ is discarded if there is no node in the neighbourhood $N[v]$ of any node v with a color c or $ii)$ Any node v does not find all the colors in its neighbourhood needs to be discarded from collection of dominating sets.

In order to ensure connectedness of each color class, we add a spanning tree rooted at a given leader node connecting all the nodes in a particular color class by adding the available nodes to get the disjoint CDSes. In a simple setup, we vary the transmission radius to determine the minimum node degree parameter for some fixed set of nodes 300 and the fixed target area 100×100. The results are shown in figure-4.8. We observe that proximity heuristic upholds a larger size of CDP compared to the randomized domatic partition based scheme.

4.7.5 Battery simulations

We investigated the effect of rotation on battery lifetime using battery property called as recharge recovery effect. For investigation of this battery related aspects arising during energy management via rotation of CDS, we conducted simple experiments to simulate a high load generated by routing traffic on CDS nodes. Assume a cross-layer function to trigger the response for rotation at V_{cutoff}, defined in [48, 27]. In a network of 100 nodes, having CDP of at most 5, a pair of 20 nodes continuously generating CBR data of 256B packets at a rate 5 packets/sec subjected to battery discharge of 1011 mA load. Assume a 2.2 watt-hour lithium-ion battery with lifetime rating 60 min. For constant battery discharge of high load subject to 1011 mA measures the lifetime 30 min against rating of 60 min [48, 27]. If CDS is rotated among CDP, so that each CDS is given a rest period to recharge. For large size CDP, the rest period is larger. The figure-4.9 shows the energy conservation via CDS rotation. We observed that battery lifetime can be extended by rotating CDS through CDP because scheduling afresh CDS introduces the rest period to exploit the battery characteristics.

The observation points out that longer the rest period, the greater the battery recovery effect. Thus, the large sizes of connected domatic partition enables substantial extension of battery lifetime resulting in enhanced network lifetime. It may be noted that for large size of connected domatic partition, the substantial improvement of network lifetime is guaranteed in sensor networks.

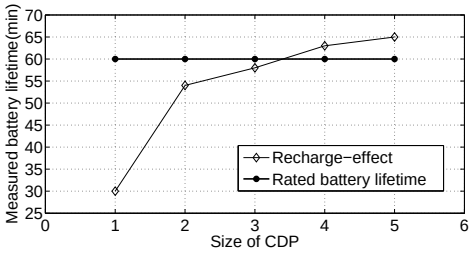


Figure 4.9: Battery energy management using recharge recovery effect via rotation of CDS

4.8 Summary

In ad hoc networks, maintaining virtual backbone via connected dominating (CDS) for efficient routing is well established. The nodes in CDS are subjected to an extra load of communication and computing so they suffer from an early exhaustion of energy resources which gives a scope of improvement in network lifetime. CDS rotation enables activation of a fresh CDS by switching through the node disjoint CDSes of the CDP to avoid overloading any particular CDS.

In this chapter, we have given a distributed technique for the connected domatic partition (CDP) problem. The network model is taken as the unit disk graph with the nodes having only connectivity information. We have given a MIS based proximity heuristics to construct CDP without relying on geometric or geographic information. To our knowledge this is the first algorithm of its kind. The advantage of our construction is lies in maximising the size of the CDP and also the simplicity achievable for rotating between the CDSes of the CDP via a local distributed switching operation. We have provided an analysis of our technique to provide an estimate of the minimum size of the CDS, the time and message complexity and also a proof of its correctness. We have also given simulation results to demonstrate its effectiveness.

Chapter 5

CDS construction using a collaborative cover heuristic

A minimum connected dominating set (MCDS) is used as virtual backbone for efficient routing and broadcasting in ad hoc sensor networks.

Theoretically, the minimum CDS problem is NP-complete even in unit disk graphs. Many heuristic based distributed approximation algorithm for MCDS problems are reported and the best known performance ratio has $(4.8 + \ln 5)$. *We propose a new heuristic called collaborative cover using two principles:* $i)$ *domatic number of a connected graph is at least two and* $ii)$ *optimal substructure defined as subset of independent dominator preferably with a common connector. We obtained a partial Steiner tree during the construction of independent set(dominators) therefore a post processing step identifies the steiner nodes in the formation of Steiner tree for independent set of* G. *We show that our collaborative cover heuristic is better than degree based heuristic in identifying* IS *and steiner tree. Our distributed approximation CDS algorithm achieves the performance ratio of at most* $(4.8 + \ln 5)opt + 1.2$, *where* opt *is the size of any optimal CDS. We show that the message complexity of our algorithm is* $O(n\Delta^2)$, Δ *being the maximum degree of a node in graph and the time complexity is* $O(n)$.

5.1 Introduction

Wireless ad hoc and sensor networks is popularly used for disaster control and geographical monitoring related applications. Such ad hoc networks lack network infrastructure for connectivity and control operations. In remote data gathering applications, the sensor network often uses in-network data aggregation to optimize network communication [45]. In-network aggregation is an intermediate processing of global data gathered often reducing the routing load thereby saving communication energy and results in increasing network lifetime.

Lossless aggregation depends on coverage of aggregating nodes. The set of aggregating nodes forms a dominating set of the network graph. These subset of nodes selected as aggregation nodes is organized in a Steiner tree to form a data aggregation backbone. The effectiveness of the aggregation algorithm is achieved when the underlying CDS tree is minimized. Therefore, constructing an aggregation backbone is modeled as the minimum connected dominating set problem in graph theory. Besides aggregation, the smaller sizes of CDS also simplifies network control operations confines routing operations to a few nodes set leading to advantages such as energy efficiency and low latency. Ad hoc networks use a CDS as a virtual backbone for efficient routing and broadcasting operations. In this work, we report an improved construction of a minimal CDS using effective coverage as a metric in collaborative cover heuristic and Steiner tree achieving the approximation factor $(4.8 + \ln 5)\mathrm{opt} + 1.2$, where opt is the size of any optimal CDS.

A connected dominating set CDS(G) of a graph $G = (V, E)$, is defined as a subset CDS(G)$\subseteq$V(G) of V(G) such that each node in $V(G) - CDS(G)$ is adjacent to at least one node in CDS(G) and the graph induced by CDS(G) is a connected subgraph of G. The problem of finding the CDS with minimum cardinality called Minimum Connected Dominating Set (MCDS) problem which is known to be NP-complete [24]. Therefore polynomial time approximation algorithms for small size CDS construction are of interest. Existing schemes for small size CDS have use degree based heuristic[14] for optimization of independent set and connectors in CDS construction. In this chapter we argue that degree based heuristic looses the coverage information due to overlapping of coverage area which is vital to further improve on the size of the CDS, leading to our new collaborative cover heuristic based on effective coverage. We describe a collaborative coverage heuristic to identify better coverage dominators based on their effective coverage. The effective coverage is ratio of coverage over the size of cover i.e. $\frac{|coverage|}{|cover|}$, where coverage means set of nodes covered by dominators and cover is the set of dominator nodes. A set of nodes having highest effective cover in its 1-hop vicinity are considered greedily for selecting them as dominators, which reduces the size of dominators. We provide a local mechanism to explore the cover with effective coverage in the distance-2 region which is used in our distributed approximation algorithm to

generate smaller size CDS.

Recent works have used a second phase in the MCDS for a Steiner tree construction to optimize the Steiner nodes to tap the independent nodes as terminals obtained in the first phase of construction to achieve an approximation factor of $(4.8 + \log 5)$. We have used the first phase of construction to generate a partial Steiner tree along with the independent set construction, this is achieved by shifting the independent set nodes to a proper placement to identify the Steiner nodes among the neighbouring nodes. Thus, unlike most of the reported schemes which fixe the independent nodes first and take second phase for Steiner tree construction, we shift the independent set (with better coverage) placement to identify most of the Steiner nodes in the first phase itself. The second phase of the algorithm then becomes a post processing step leading to a Steiner tree of no higher cost.

In the energy constrained ad hoc and sensor networks such schemes help to extend the network lifetime due its smaller size CDS compared to other CDS schemes, in terms of: $i)$ A smaller dominating set resulting in larger domatic partition giving better energy conservation and $ii)$ Smaller size dominating set means large coverage giving high degree of data aggregation thereby reducing the network traffic.

The described algorithm has $O(n\Delta^2)$ message complexity, Δ being the max degree of node in graph. The approximation factor of distributed algorithm for finding minimum connected dominating set is $(4.8 + \ln 5)\mathrm{opt} + 1.2$, where opt is the size of any optimal CDS.

The rest of chapter is organized as follows. In section 5.2 we discuss related works on CDS construction algorithms. Section 5.3 is on preliminaries giving definitions and a brief background necessary for our work. Section 5.4 states problem formulation and lists the contributions of this work. Section 5.5 explains the principles behind our collaborative cover heuristic. Steiner tree construction from a given set of dominators is explained in section 5.6. In section 5.7 we present our distributed algorithm for aggregation-CDS based on collaborative cover. Section 5.8 is on analysis of the algorithm. We give simulation results in section 5.9. Finally, we conclude in section 5.10.

5.2 Related Work

In this section we review the literature, which is divided into the following two sections:

5.2.1 In-network aggregation problem

Several reported schemes on routing algorithms such as: Directed Diffusion [55], Pegasis [56] and GAF [57], have used in-network data aggregation where a spanning tree performs aggregation function opportunistically along the internals of the tree, as data flows level by level from leaves to root. The opportunistic aggregation based schemes are neither optimal nor giving approximation guarantees. The aggregation schemes are categorized into two types: $i)$ lossless aggregation and $ii)$ lossy aggregation.

The lossy aggregation schemes are based on exploiting correlated data in tree construction. A connected correlation dominating set scheme reported in [58] constructs CDS for capturing correlation structure to provide lossy aggregation efficiently. We have not come across any significant reported matter on lossless aggregation.

5.2.2 Minimum connected dominating set problem

The use of the connected dominating set (CDS) as a virtual backbone was first proposed by Ephermides in 1987 [11]. Since, then many algorithms that construct CDS have been reported and can be classified into the following four categories based on the network information they use: $i)$ centralized algorithms, $ii)$ distributed algorithms using single leader, $iii)$ distributed algorithm using multiple leaders and $iv)$ localized algorithms.

Guha and Khullar [50] first gave two centralized greedy algorithms for CDS construction in general graphs having approximation ratio $O(ln\Delta)$. Centralized CDS algorithm to be used as virtual backbone for routing application was first reported by Das in [46]. The centralized CDS algorithms requires global information of the complete network. Hence, it is not suited for wireless sensor networks which do not have centralized control. Construction of CDS may be achieved through a distributed algorithm based on either a single leader or multiple leaders.

Distributed algorithms with multiple leader approach does not require a initial node to construct CDS. Alzoubi's technique [59] first constructs an MIS using a distributed approach without a leader or tree construction and then interconnects MIS nodes to get a CDS. Wu and Li in [60] reported a CDS algorithm to identify the CDS using a marking approach to identify dominators with independent nodes and then prune the redundant nodes from the CDS using two set of pruning rules to generate CDS. The multiple leader minimum CDS schemes approximates size of min-CDS to $192\mathrm{opt}+48$, where opt is the size of optimal CDS [59]. Due to its large approximation factor, the multiple leader based distributed CDS construction is not effective for exploiting lossless in-network aggregation. In a localized approach for CDS, construction Adjih [12] presented a approach for constructing small size CDS based on multipoint relays

(MPR) but no approximation analysis of algorithm is known as yet. Based on the MPR approach several extensions have been reported leading to localized MPR based CDS construction. The localized without a approximation guarantees is again not competitive to efficiently exploit aggregation.

A single leader distributed algorithm for CDS assumes an initial leader in place to provide initialization for the construction of distributed algorithm. A base station could be the initiator for construction of CDS in sensor networks. The distributed algorithm uses the idea of identifying an maximal independent set (MIS) and then identifies a set of connectors to connect the MIS is ascertained to form CDS. Alzoubi [13] presented an ID based distributed algorithm to construct a CDS tree rooted at the leader. For UDGs, Alzoubi's [13] approach guarantees approximation factor on size of CDS atmost $8|\text{opt}| + 1$, has $O(n)$ time complexity and having $O(n \log n)$ of message complexity to construct CDS using a single initiator. The approximation factor on the size of CDS was later improved in another work reported by Cardei [14] having an approximation factor of $8|\text{opt}|$ for degree based heuristic and degree aware optimization for identifying Steiner nodes as the connectors in CDS construction. This distributed algorithm grows from a single leader and has $O(n)$ message complexity, $O(\Delta n)$ time complexity, using 1-hop neighbourhood information. Later, Li in [61] reported a better approximation factor of $4.8 + \log 5$ by constructing a Steiner tree when connecting all nodes in I, the independent dominating set.

5.3 Preliminaries

This section is divided into two parts: $i)$ dominating set and $ii)$ network model.

A. Dominating set: Wireless networks generally have omni-directional antennae and nodes use transmission power to establish connection with all nodes in the transmission range. Assume that medium access control layer protocol deals with the intricacies of interference of radio signals, channel regulation, collision handling giving us way to model network as unit disk graph. A graph G=(V,E) is a unit disk graph(UDG) if there exist $\Phi : V \mapsto \mathbb{R}^2$ satisfying $(i, j)\epsilon E$ iff $\parallel \Phi(i) - \Phi(j) \parallel_2$. Φ is called a realization of G. Thus, wireless network is modeled as UDG. In a given graph $G = (V, E), V' \subseteq V$ a subset is a maximal independent set (MIS) if no two vertices in V' are adjacent (independence) and that every $u\epsilon V - V'$ has a neighbour in V' (maximality). A dominating set D is a subset of V such that any node not in D has a neighbour in D. A maximal independent set is also a dominating set in the graph and every dominating set that is independent must be maximal independent, so maximal independent sets are also called independent dominating sets. If the induced subgraph of a dominating set D is connected, then D is connected dominating set (CDS). The relationship between size of a MIS of G and the minimum connected dominating set CDS of G plays an important role in estab-

lishing the approximation factor of approximation algorithm for minimum connected dominating set. Wan[13] showed that in every UDG G, $|\mathrm{MIS}(G)| \leq 4|\mathrm{CDS}(G)| + 1$ which was improved by Wu[31] to $|\mathrm{MIS}(G)| \leq 3.8|\mathrm{CDS}(G)| + 1.2$. We use the improved relationship of MIS and min-CDS for approximation analysis of our proposed algorithm.

B. Ad hoc Network Model: Distances are Unknown We describe the network model used in this work. Assume that nodes do not have any geometric or topological information, thus even the distances to neighbours are unknown to the nodes. The communication overhead due to interference is assumed to be negligible. The computation is partitioned into rounds. Assume that the nodes receive all messages sent in previous round, execute local computations and send messages to neighbours in a round. A wireless ad hoc network is represented as a UDG. Nodes using exchange of hello messages can find its distance-1 neighbour nodes and ascertain its degree. Given $G(V, E)$, G^2 has vertex set $V(G)$ and edge set $E^2 = \{\{u, v\} | u, v \in V(G) \wedge \text{shortest distance}(u, v) \leq 2$.

5.4 Problem formulation and contributions

Consider wireless sensor network consisting of a (large) number (n) of nodes deployed in a geographical region. Each node is mounted by an omni-directional antenna with the transceivers having maximum transmission range of R. The ad hoc network is a unit disk graph $G = (V, E)$ where $|V| = n$ be all the nodes, E be the edges and edge between any pair of node exists if the distances is at most R, taken a a unit radius. The problem is to find a minimum cardinality connected dominating set of G is NP-complete. Therefore, the aim of this work is the development of heuristic based approach to construct a CDS with guaranteed approximation factor to the size of any optimal CDS. When a minimal CDS is used as aggregation backbone for lossless in-network aggregation problem, it saves the network traffic leading to increased lifetime of the energy constrained ad hoc and sensor networks.

5.4.1 Contributions

The contribution of this chapter is summarized as the following:

1. A distributed approximation algorithm for minimum connected dominating set problem with a known initiator.

2. A new collaborative cover heuristic which helps in identifying smaller cardinality MIS of G as compared to ID based or degree based heuristics.

3. A Steiner tree construction process in two phases:

   (a) Steiner nodes identified in the first phase to drive the MIS construction by shifting independent set nodes to locate the connectors in identifying Steiner nodes and

   (b) second phase becomes a post processing step of identifying the Steiner nodes to construct the CDS tree statisfying a standard bound.

   (c) The approximation factor of our algorithm is $(4.8 + \ln 5)\mathrm{opt} + 1.2$, where opt is the size of any optimal CDS. The algorithm has time complexity of $O(n)$ and $O(D)$ rounds, where D is network diameter. The algorithm requires atmost $O(n\Delta^2)$ messages for its construction complexity, where Δ is maximum node degree in G.

We have shown that our CDS approach when used for in-network aggregation application, prolongs the network lifetime.

5.5 Collaborative cover heuristic

Reported work on distributed approximation algorithm for CDS construction using a single leader either use ID based heuristic[13] or degree based heuristic[14]. Cardei[14] has shown that degree based heuristic is better as compared to a pure ID based heuristic in identifying smaller size CDSes greedily. In identifying a MIS using degree based heuristics, nodes with highest degree in their neighbourhood are selected greedily forming an MIS of the underlying graph.

An improvement over the existing degree based heuristic is a new collaborative cover heuristic described in this chapter. The collaborative cover heuristic is based on the idea of using the information of overlapping coverage of the nearby independent set of nodes. On considering the nearby independent nodes, we observe that the effective coverage is less when they are considered in isolation. In a degree based heuristic each node is considered in the isolation thereby loosing important information to further optimize the size of MIS and CDS. The loss of effective coverage is due to overlapping of coverage area of nearby independent nodes. Therefore, instead of effective degrees being considered in isolation, we propose a more encompassing heuristic which considers the coverage of nearby independent nodes while identifying effective coverage (or effective cover of network nodes). Thus, the collaborative cover heuristics is based on effective coverage information which intuitively is better than effective degree. We now provide a formalised definition of the concept of collaborative cover.

Definition 5.1 (Node neighbourhoods) *Consider a node u. Nodes covered by u is represented as $N(u)$, known as neighbours of u. The set $N[u]$ represents nodes covered by u including u. Let the nodes be called independent if they are not neighbours. Independent neighbour of u is a subset of $N(u)$ such that any pair of nodes in this subset are independent. $N_2(u)$ is a set of nodes which are at most at a distance-2 from u known as at most distance-2 neighbours of u. Let the distance-2 neighbours of u is represented as $\{N_2(u) - N(u)\}$.*

For any node, we now define a cover of its distance-2 neighbours such that any pair in the cover are independent.

Definition 5.2 (Distance-2 independent halo) *Let H be the independent cover of the distance-2 neighbour of u. If H is an independent cover then $H \subseteq \{N_2(u) - N(u)\}$ and $\{N_2(u) - N(u)\} \subseteq N[H]$ and any pair of nodes in H are independent.*

Such a cover H of $\{N_2(u) - N(u)\}$ where any pair of nodes in H are independent is obtained using either ID based or a *degree* based heuristic. Note that in either of heuristic, any pair of independent node in H which are distance-2 neighbours has ignored the estimate of coverage loss due to the overlapping in coverage. Further, these independent nodes later requires additional Steiner nodes to form the connected substructure. With this background, we now argue a need of new heuristic which accounts for effective coverage. We propose a collaborative cover heuristic to compute the effective coverage of independent distance-2 neighbour nodes collaboratively.

Definition 5.3 (Independent coversof a node neighbourhood) *Let v_H be node in H and let $R_H = \{N(v_H) \cap \{N_2(u) - N(u)\}\}$ be the coverage of v_H for distance-2 region of u. Then $I(R_H)$ be any independent set of R_H which covers R_H. Thus $R_H \subseteq N[I(R_H)]$. Therefore, node v_H and any independent set in its neighbourhood $I(R_H)$ form the disjoint covers of R_H. Note that there may be multiple such instances of independent sets $I(R_H)$. Let S be the set all instances of independent sets of R_H where each independent set I covers the region $R_H \subseteq N[I_i]$ for $1 \leq i \leq p$, so let $S = MIS(R_H) = \{I_1, I_2, \ldots, I_p\}$.*

Consider any node v_H and a subset of its neighbourhood region R_H. We know that v_H covers the region R_H. There are many possible independent sets (IS) in region R_H each of which covers R_H. Let the set S denote a set of IS which can cover R_H. We have to compute weights for each instance of IS on analyzing its coverage to ascertain its quality. Next we define a measure to compute its effective coverage weight.

Definition 5.4 (Effective coverage) *The effective coverage weight of an independent set(I_i) with respect to a region($\{N_2(u) - N(u)\}$) is the ratio of coverage for the region by the independent set over size of independent set. Thus, effective coverage weight=*$\frac{N[I_i] \cap \{N_2(u) - N(u)\}}{|I_i|}$

The effective coverage weight is computed for each independent set to identify an ordered pair of (I_i, wt_i). We can now identify a weighted independent set to cover a given region R_H.

Definition 5.5 (Weighted independent covers of a node neighbourhood) *The weighted independent set (I_i, wt_i) (for $1 \leq i \leq p$) is an ordered pair of independent set and its effective coverage weight such that each independent set is a cover of the region $R_H = \{N[v_H] \cap \{N_2(u) - N(u)\}\}$. Thus $R_H \subseteq N[I_i]$ for ($1 \leq i \leq p$). Let, the region R_H has p number of covers with the weights represent the ratio of the effective coverage over the cardinality of cover. Thus the weighted independent cover is given by $\{(I_1, wt_1), (I_2, wt_2), \ldots, (I_p, wt_p)\}$.*

In addition to associating the weights for effective coverage with independent sets, we look for those I in S which have a common neighbour node in $N(u)$. Thus, the condition for I which does the check is $\{N[I] \cap N(u)\} \neq 0$. The common neighbour node is called as connector because it can connect the node u and its distance-2 independent neighbours.

Definition 5.6 (Independent set with a common connector to parent dominator) *The independent set I_i with at least a common connector in $N(u)$ is stated as:* $\exists w \in N(u)[I_i|$ *w* $\text{connects at least 2 nodes of I}_\text{i}, \text{ i.e } |N(w) \cap I_i| \geq 2]$.

For any node v_H, the independent set I_i and its effective coverage weight wt_i associated with a connector w together forms a tuple $t_H = (I_i, wt_i, w)$.

The collaborative cover heuristics proposed in this chapter is based on the intuitive argument that degree based heuristic may result to a non-optimal choice locally in the construction of CDS leading to a non-optimal CDS eventually. The collaborative cover heuristic often replaces a non-optimal choice of degree based heuristic with the improved effective coverage using collaborative cover locally. The replacement of degree based selection with collaborative cover based selection suggests the existence of multiple cover locally. Since, the domatic number of any connected graph is at least 2 by Ore's theorem (in lemma-5.1), therefore premise of multiple cover is validated to explore and prune the local best cover.

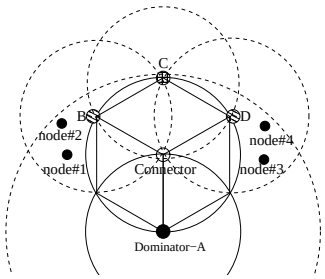


Figure 5.1: Example for comparing collaborative cover and degree based heuristics

Result 5.1 (*By Ore in 1962 [8, 62]) For a connected graph G, the $domatic\ number$ of $G \geq 2$.*

Thus, at every stage of connected graph there exists at least two cover in graph and our approach aims to improve locally with the local best approximation to reduce size of CDS eventually for minimum connected dominating set problem.

Definition 5.7 (Optimal sub-structure) *Let node w be called as connector if it is common neighbour between dominators u and v, where v is the distance-2 neighbour of u.*

An optimal substructure is a tuple (I_i, wt_i, w) in the neighbourhood $N(v)$ of any node v is a highest weight independent set with a common connector w which can connect an IS to some node u and if the weight of the IS is greater than the coverage of the node v for a given region (i.e. effective coverage $wt_i >$ coverage of node $|R_H|$).

Example 5.1 A CDS construction stage of ad hoc network is shown figure-5.1, which consists of a dominator-A, three potential dominators (B, C, D) and six nodes (having two nodes as neighbour to each B, C, D). Let the dominator-A need to select its distance-2 dominators out of the potential choices B, C and D.

According to degree based heuristic, the potential dominator C covers four nodes compared to B and D at three each. Therefore, C becomes a dominator whereas B and D stay as its member nodes. The size of cover for C becomes 1 and coverage of C is 4. Further, in order to cover the nodes $\{1, 2, 3, 4\}$ at least 2 more dominators are needed. Thus the cover size is at least 3 for coverage of 4 nodes (considering only 2-hop cover of A). Thus, dominator C requires two more dominators one from each sets: $\{1, 2\}$ and $\{3, 4\}$, leading to the required three dominators based on degree information. Thus, the weight of the cover is given as: weight=$\frac{|coverage|}{|cover|}$=$\frac{4}{3}$=$1.33$.

Based on the collaborative cover heuristic, the potential dominators B, D are selected as dominators. The size of cover becomes as 2 and the coverage of dominating

set $\{B, D\}$ is 5. The collaborative cover $\{B, D\}$ of size 2 has a coverage of 5. Thus, effective coverage of collaborative cover has the weight=$\frac{|coverage|}{|cover|}$=$\frac{5}{2}$=$2.5$. □

Higher weight indicating more coverage in collaborative cover heuristics as compared to the degree based heuristic leading to smaller size of cover. Furthermore, the number of connector needed in collaborative adds to single number as compared to degree based heuristic of more than one.

Theorem 5.2 (Local identification of optimal sub-structure) *The optimal substructure is computed locally requiring only distance-2 local information.*

Proof: It is evident from example 5.1 that all covers in the neighbourhood of a potential dominator are evaluated and the best is finally chosen. This entire process is carried out locally, around the potential dominator, requiring only distance-2 local information. □

In the next section, we describe the construction of Steiner tree carried out in over two phases of the CDS construction.

5.6 Steiner tree construction

A Steiner tree for a given subset of nodes (called as terminals) I in a graph G, is a tree interconnecting (known as tapping) all the terminals I using a set of Steiner nodes in $\{V(G) - I\}$. We can connect maximal independent set I by using Steiner nodes forming a Steiner tree inter-connecting all the nodes in I. The objective is to find a Steiner tree with minimum number of Steiner nodes to obtain a small size of CDS. We define the Steiner tree with minimal Steiner nodes as:

Definition 5.8 (Minimal Steiner nodes) *Let $I \subseteq V(G)$ be the maximal independent set I of G. Minimal Steiner nodes is subset $V(G) - I$, forming a Steiner tree to interconnect (or tap) the independent nodes I (or terminals).*

For unit disk graphs, the Steiner nodes has a property that any Steiner node can tap at most five independent nodes (or terminals). From the property of unit disk graph given in [51], we know that any node is adjacent to at most five independent nodes. Therefore, any Steiner node can interconnect at most five independent (terminal) nodes. Using this property, we define our scheme to identify the Steiner nodes in the following steps:

Step-1 All the dominatee node with 5 adjacent independent nodes from separate components are chosen become Steiner nodes and the set of adjacent independent nodes forms a connected component. Note that new component thus obtained by an association of Steiner node and its adjacent independent set nodes of different components, reduces the number of components in the network which needs to be updated to dominatee having the adjacent independent set in different components.

Step-2 For each dominatee, recompute the adjacent independent nodes in different components information.

Step-3 Repeat the above steps (1..2) for dominatees having four adjacent independent set nodes in different components.

Step-4 Repeat the above steps (1..2) for dominatees having three adjacent independent set nodes in different components.

Step-5 Repeat the above steps (1..2) for dominatees having two adjacent independent set nodes in different components.

Thus the set of the Steiner nodes forming a single connected component of independent set nodes contributes to CDS. In the next section we describe our CDS algorithm using heuristic based on collaborative cover.

5.7 CDS using the collaborative cover heuristic

Let every node know its distance-1 neighbours and its distance-2 neighbours. Assume that every node also knows its maximal independent set (MIS) in the unit disk around it.

The CDS construction grows the CDS-tree incrementally in a BFS manner. Each node maintains the following state variables: $i)$ The pointer $parent$ is used for the parent link in CDS-tree, $ii)$ The level variable l indicates the level of node from root ($l = 0$) of CDS-tree in BFS construction and $iii)$ The $color$ variable records the current status of node (initially all the nodes are white, dominators and connectors are colored black, potential dominator at distance-2 takes yellow color, whereas dominatees are grey).

Let u be a leader node which initiates the construction of CDS algorithm. The algorithm has three main steps: $i)$ This step is to identify the independent set (cover) of the distance-2 neighbours using degree based heuristic $ii)$ This step computes the collaborative cover for each node of a cover (identified in step $i)$) and a weight based

on effective coverage and $iii)$ This step is to identify a connector, if any, for the highest weight independent set (identified in step $ii)$) with u.

The algorithm starts at the leader node to identify dominators and connectors in CDS-tree constructing two levels at a time (level-l dominator to level-$(l+1)$ connector and level-$(l+1)$ connector to level-$(l+2)$ dominator) of the CDS-tree at each step until no idle nodes are left.

The set of yellow leaders forms an MIS of distance-2 region of u. The yellow leaders perform two tasks: $i)$ identify leaders of yellow leaders in its 2-hop adjacent yellow leaders to form an MIS of yellow leaders induced by graph $G^2[yellow\text{-}leaders]$, and $ii)$ for each yellow leader, compute the MIS of yellow neighbours with common grey nodes.

The yellow leader computes the MIS with common grey neighbour and identifies highest effective coverage MIS among them.

The yellow leader compares its coverage with the highest weight effective coverage of MIS with common adjacent grey nodes. The yellow leader becomes active if its effective-coverage weight has larger coverage than its own coverage. Note that active yellow leader satisfies the following three properties represented by a tuple (I_i, wt_i, w_i) which triggers to explore alternate MIS with better coverage to elect leaders of yellow leaders in the entire yellow leaders of u:

1. size of MIS I_i of node is atleast two,
2. independent nodes of MIS has a common connector w_i and
3. effective coverage weight wt_i of MIS is greater than coverage of a node itself

The active yellow leader sends effective coverage of MIS to its 2-hop neighbouring yellow leaders. $G^2[yellow\ leader]$ is the subgraph of G^2 induced by $yellow\ leaders$. Note that for any given $yellow\ leader$, the subgraph $G^2[yellow\ leader]$ identifies $yellow\ leaders$ in its distance-two neighbourhood. The leaders of yellow leaders are identified based on their effective coverage, which form MIS of graph in $G^2[yellow\ leaders]$ which is a subgraph of G^2 induced by yellow leaders. The yellow leaders are pruned locally to identify an improved MIS based on coverage heuristics in following two phases: $i)$ In the first phase the leaders of yellow leaders grows its highest effective coverage MIS with common grey to become as dominators. $ii)$ In second phase the remaining yellow leaders use the dominators to forms its MIS and then grow them to become dominator. Note that in above two phases, the MIS of distance-2 neighbours of u is identified and updated as dominators. These dominators trigger selection of the adjacent grey nodes which connect highest number of dominators.

At this point node u has identified distance-2 cover preferably as dominators with a connector. The size of cover is reduced heuristically for a larger coverage. Once the dominators (at level-$(l+2)$) and connectors (at level-$(l+1)$) are identified, the (level-$(l+2)$) dominators become leaders to repeat the steps to grow the CDS-tree further until no white nodes are left. After the end of the first phase, the algorithm has identified MIS and the connectors. These connectors which form an initial Steiner tree are discarded to identify new Steiner nodes in second phase. In the second phase, iteratively the Steiner nodes are picked which connects independent set nodes in different components. At the end of second phase the Steiner tree is formed out of Steiner nodes thus identified. It may be noted that the collaborative cover process involves an optimization to reduce the number of dominators. The computation is local therefore it is suitable for computing using a distributed approach.

Algorithm 8 Algorithm for CDS based on collaborative cover heuristic

1: Initialize $\langle parent = \text{nil}\rangle$, level$\langle l = 0\rangle$, $\langle color = \text{white}\rangle$, $count = 0$ for each node.
2: Consider a leader node u initiating construction of the CDS. Leader node u, becomes a dominator and updates its state as $\langle color = \text{black}, parent = \text{ID}, l = 1\rangle$.
3: Node u sends message $m_1 = \langle u, l\rangle$ to its adjacent nodes.
4: Each adjacent node w on receiving $m_1 = \langle u, l\rangle$ from u becomes a dominatee and updates its variables as $\langle color = \text{gray}, parent = \text{u, level } l_w = l_u + 1\rangle$. Node w sends message $m_2 = \langle w, u, l_u + 1\rangle$ to identify the distance-2 nodes of u.
5: A white node v on receiving m_2 from w, becomes a distance-2 neighbour of u and updates its state variables as $\langle color = \text{yellow}, \text{level } l_v = l_u + 2\rangle$ and records its adjacent grey neighbours $N_{grey}(v) = \{w\}$, initialises adjacent yellow neighbours $N_{yellow} = nil$, updates effective degree nodes $N_{\text{eff}}(v) = N(v) - \{w\}$, where $N(v)$ is the nodes adjacent to v.
6: After a lapse of τ time, when all the m_2 messages are delivered to yellow nodes v, the yellow nodes v broadcast message $m_3 = \langle |N_{\text{eff}}(v)|\rangle$ containing its effective degree to its adjacent yellow nodes v.
7: Yellow nodes v of u on receiving m_3 from v' update its adjacent yellow neighbours $N_{yellow} = N_{yellow} \cup \{v'\}$, ranks its adjacent yellow nodes on the basis of their effective degree ($|N_{\text{eff}}|$, ID), where node ID is used for tie breaking. If node v has the highest effective degree node in its distance-1 vicinity, then v becomes a yellow leader. The yellow leader v broadcasts message $m_4 = \langle N_{yellow}(v)\rangle$ containing its coverage of yellow nodes to its adjacent yellow nodes.
8: Each yellow node v (of u) on receiving m_4 from yellow leader v', computes $I_{v'}(v) = N_{yellow}[v'] - N_{yellow}[v]$, the set of yellow nodes in the neighbourhood of v' not adjacent to v and broadcasts message $m_5 = \langle v, I_{v'}(v), N_{grey}(v), N_{\text{eff}}(v)\rangle$ to the yellow leader node v'.
9: Each yellow leader v (of u) on receiving m_4 from v' (of u), computes all $\text{MIS}(\text{yellow neighbours}(v))$ and then selects only those MISes whose $|MIS| > 1$ and have common grey neighbours as $D(v) = \{D_1, \ldots, D_k\}$ (possibly empty).

Node v computes effective coverage of each $D_i, (\forall i \in 1..k)$. The effective coverage weight of $D_i(v)$ is given by:

$$weight_i = \frac{|N[D_i(v)] \cap (N_2(u) - N(u))|}{|D_i(v)|}$$

This forms a tuple $D(v) = \{(D_1, wt_1, w_1), \ldots, (D_k, wt_k, w_k)\}$, where wt_i represents the coverage weight and w_k is common connector node at level-$(l+1)$. Each yellow leader node identifies on the basis of highest effective coverage weight, the MIS set D_h in its neighbourhood (arbitrarily select one in case of tie). If the highest effective coverage weight, of the MIS set D_h is greater than the coverage of v itself, then yellow leader becomes *active*. Each active yellow leader v, sends message $m_5 = \langle \text{eff. coverage}(D_h), \text{ID} \rangle$ to its 2-hop neighbouring yellow leaders of v. (* Note that active yellow leader means it has an MIS which three properties $i)$ $|\text{MIS}| \geq 2$, $ii)$ MIS has at least one common grey node and $iii)$ effective coverage weight indicates that the effective coverage of this MIS is greater than coverage of yellow leader node itself. The active yellow leader triggers the pruning of MIS by activating all yellow leaders to elect a new set of MIS. *)

10: Each active yellow leader v (of u) on receiving m_5 resolves the leaders of (active) yellow leader with highest effective coverage in its 2-hop region. The set of yellow leaders undergoes local pruning to identify local best coverage $MIS(N_2(u))$ (i.e an MIS of $N_2(u)$) in following two phases:

1. In first phase each leader of yellow leaders in ($G^2[\text{yellow leaders}]$) is identified and the nodes its D_h become dominators and update $color = \text{black}$. Their common grey nodes becomes connectors by receipt of a message m_6.

2. In second phase the remaining uncovered yellow nodes identify their MIS to become dominators (updating their colour to black) to cover all the yellow nodes. The dominators of second phase sends message m_7 to select their connectors amongst the grey nodes (preferably which are already connectors of first phase).

11: Particular grey nodes at level $l+1$ on receiving m_6 or m_7 come to know whether they are connectors.

12: Note that the identification of connectors among the grey nodes completes the construction three levels $l, l+1, l+2$ of CDS construction. The connectors at level-$(l+1)$ are identified to connect level-l dominators with level-$(l+2)$ dominators by breadth first expansion of the CDS-tree in a distributed manner.

13: The algorithm phase-I terminates when no white nodes left unexplored.
(* **Phase-II: Identifying Steiner nodes for dominator nodes** *)
(* Phase-II discards the connectors and iteratively identifies Steiner nodes for connecting independent set nodes belonging to different components *)

14: Each node in I broadcasts m_{10} message so that dominatees can know of adjacent independent set nodes in different components.
15: Initially all independent set nodes forms different components and the Steiner nodes list is empty. In the next step, dominatees having required number of adjacent independent set nodes in different components are identified as Steiner nodes iteratively.
16: **for** $i = 5, 4, 3, 2$ **do**
17: **while** a grey node v exists having i-adjacent independent nodes of I in different components **do**
18: Add node v into Steiner nodes list
19: **end while**
20: **end for**(* The identified Steiner nodes connect the dominator nodes to form a Steiner tree. Thus, independent set nodes and Steiner nodes forms the CDS of G *)

5.8 Algorithm analysis

In analysis of algorithm-8, we provide the approximation factor of size of CDS and complexity analysis in following sub-sections.

5.8.1 Approximation analysis of CDS algorithm

Lemma 5.3 *For the algorithm-8, the size of every maximal independent set computed in phase-I is at most* $3.8\mathrm{opt} + 1.2$ *where* opt *is the size of a minimum connected dominating set in the unit disk graph.*

Proof: From the result reported in [31].

□

Lemma 5.4 *The size of Steiner nodes obtained from algorithm-8 is at most* $(1 + \ln 5)\mathrm{opt}$*, where* opt *be size of any optimal CDS.*

Proof: The proof follows directly from theorem-2 of [61] because at step-15 of algorithm-8, the set of connector nodes originally identifed are discarded and a new set of Steiner nodes are identified in steps 16 to 20, also based on the Steiner node identification scheme reported in [61]. □

It may be noted that steps steps 16 to 20 for algorithm-8 may optionally be skipped and the original set of connectors used, in which case lemma-5.4 will no longer apply. However, in the section 5.9 we show that original set of connectors that are identifed compare well the connectors identified in steps 16 to 20.

Theorem 5.5 *For algorithm-8, the size of CDS is at most* $(4.8 + \ln 5)\mathrm{opt} + 1.2$*, where* opt *is the size of any optimal CDS.*

Proof: From lemma-5.3 and lemma-5.4, we have:

$$\begin{aligned} |\mathrm{CDS}| &= |I| + |\text{Steiner nodes}| \\ &= 3.8\mathrm{opt} + 1.2 + (1 + \ln 5)\mathrm{opt} \\ &= (4.8 + \ln 5)\mathrm{opt} + 1.2 \end{aligned}$$

□

5.8.2 Complexity Analysis

Theorem 5.6 *The algorithm for Connected dominating set has time complexity* $O(n)$ *time and* $O(D)$ *rounds, where* D *is the network diameter and message complexity of* $O(n\Delta^2)$*, where*Δ *is max degree of node in* G*.*

Proof: Assume that in a given unit disk the size of an MIS is always less than maximum degree of a node in G, therefore $|\mathrm{MIS}| \leq \Delta$. Each node sends at most two messages to become grey (dominatee) and at most Δ messages per degree to update neighbour's information and Δ^2 to get neighbours of neighbour, to become dominator. Thus, message complexity is $O(n\Delta^2)$, where Δ is the maximum node degree.

While establishing the relationship between connectors and dominators the message complexity is only size of CDS which is at most $O(n)$. Thus the message complexity of algorithm $O(n\Delta^2)$. Each node is explored one by one, so the time complexity $O(n)$. The number of synchronous rounds is $O(D)$, where D is network diameter, which is bounded by shortest distance of farthest node from a given leader. □

5.9 Simulation results

In this section we present the simulation results of algorithm-8. The goals of these simulations is the following: $i)$ performance comparison of Steiner nodes with independent set nodes $ii)$ performance comparison of Steiner nodes against ignored connectors $iii)$ performance comparison with the related techniques, and (iv) energy analysis of network for exploiting aggregation.

We model wireless ad hoc sensor network as a set of nodes deployed in a predetermined rectangular area of dimension 100×100 square units called as deployment area A. Each node has a unique ID. We use a uniform random number generator that chooses the x and y coordinates in deployment area A for sensor nodes. For simulation experiments, we consider the network of varying sizes. We assume that each wireless sensor node has the same transmission range r. Unless stated otherwise, we assume maximum transmission range $r = 25$. The edge between any pair of nodes indicates that distance between them is at most radius r. Since the maximum transmission r is fixed parameter of our experiments in the given network, thus the induced graph is unit disk graph. For a given r, the number of nodes per unit area called as network density (d) increases as the network size (n) increase. The approximate governing relation for the transmission radius is given by $r^2 = (d*A)/(\pi*n)$ [45]. The simulation is carried out by varying the network size(n), so that impact of network size can be observed on size of connected dominating set. The parameters nodes (n), transmission range (r) thus generated, are used in our experimental setup of simulation. The deployment area in our experimental setup is assumed to be of rectangular shape which effects the nodes located at border as low degrees called as border effect. In our simulations, to offset border effect, we use a correction of higher transmission radius judiciously to nullify the border effect. The Simulation is carried out in PROWLER/MATLAB, an event driven simulator for Ad hoc Networks.

We first compare the performance of the Steiner nodes required to connect the independent set nodes using a metric which is ratio of number of Steiner nodes is to number of independent set nodes. The results shown in figure-5.2 for large size networks it comes out to be less than 0.3, which indicates the Steiner nodes often connects more than three independent sets to achieve the results.

Next we analyze through simulation the performance of Steiner nodes as compared to connectors identified while identifying independent set which are ignored to identify optimal Steiner nodes as a post -processing step. We give an account of how far we achieved in partial Steiner tree in our collaborative cover CDS algorithm.

The performance comparison are shown in figure-5.3. The results show that our collaborative cover is quite close in identifying partial Steiner tree in its first phase of construction and therefore, requires post processing step only to identify some of the

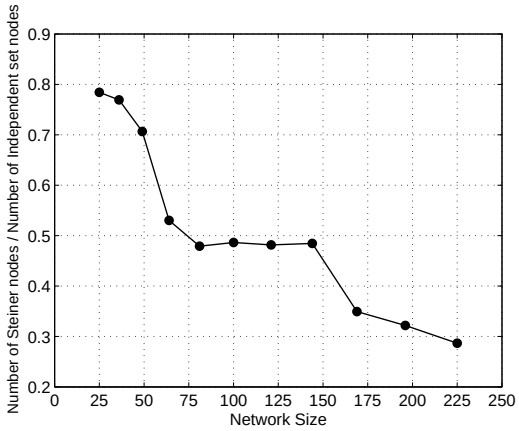


Figure 5.2: Performance comparison of number of Steiner nodes and number of independent nodes

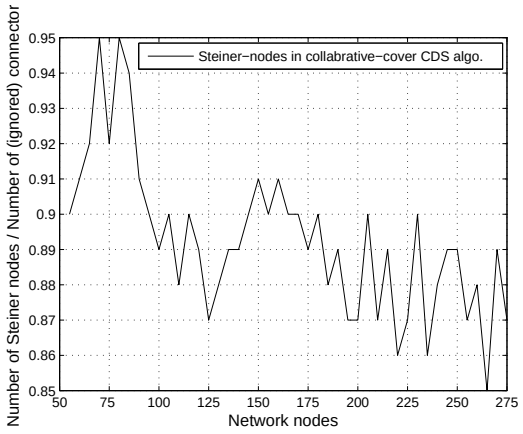


Figure 5.3: Performance comparison of Steiner nodes with (ignored) connectors

optimal Steiner nodes to achieve Steiner tree.

Note that besides this our collaborative cover also gains in reducing independent set which is discussed in later part of this section.

We also analyze the message exchanges for CDS construction in our algorithm. The comparison shows that number of messages in our CDS construction are closer to that of degree-CDS approach. Thus, our collaborative-cover CDS is not sacrificing on the message overheads. The message complexity analysis of $O(n\Delta^2)$, where Δ is max degree of G, is also validated by comparing the simulation results (shown in figure-5.4) with degree-CDS scheme.

Finally,we compare the performance of our collaborative cover based CDS algorithm with the CDS algorithm reported by Cardei in [14], by Alzoubi in [13] and by Li in [61] . Assume the maximum transmission range values to be varying between(25,50)

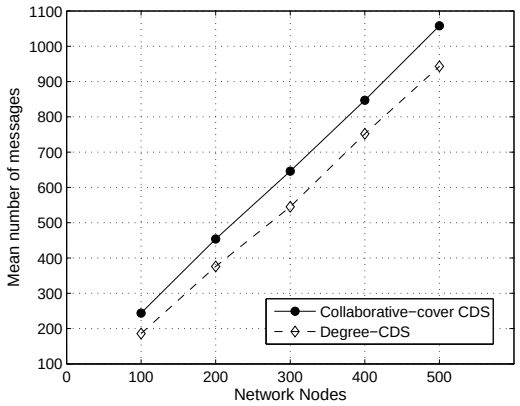


Figure 5.4: Comparison of message exchanges in CDS construction

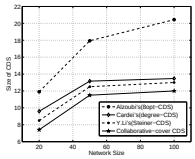


Figure 5.5: Performance comparison with CDS algorithms (R=25)

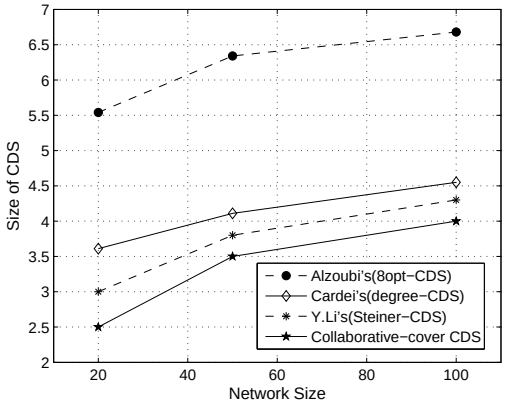


Figure 5.6: Performance comparison with CDS algorithms (R=50)

units for the network with varying the node sizes as (20, 50 and 100). We considered only the connected graph for our result analysis.

The performance comparison shown in figure5.5,for max.transmission range r=25 whereas for r=50 is shown in figure5.6. The simulation results reveal that our collaborative cover based CDS algorithm reduces the size of CDS by 15% compared to Cardei et al.'s [14] approach whereas reduction of CDS size is 10% in Li's CDS [61] approach. From both the results, we observe that our proposed is better than Alzoubi's [13], Cardei's [14] and Li's [61] approach in identifying a smaller size of CDS.

Table 5.1: Description of parameters

Parameter	Value	Summary
E_l	50nJ/bit	Energy dissipated in transceiver for per bit operation.
E_{agg}	5nJ/bit	Energy dissipated in data aggregation per bit
α_{friss}	10pJ/bit/m^2	radio transmitter coefficient for short distances.
α_{2-ray}	0.0013pJ/bit/m^4	radio transmitter coefficient for longer distances.
M	100m^2	target area of 100x100m^2.
m	1000bit	frame size in bit per round of data gathering.

Aggregation based energy model

In order to evaluate the energy profile for data aggregation in our aggregation-CDS algorithm, we considered an aggregation based energy model. Let the energy dissipation for aggregation to be 5nJ/bit. This value is drawn from realistic experimentation reported in literature as energy dissipation for performing beamforming computations to aggregate data is 5nJ/bit/ signal [3]. The table-1 summarizes the system parameter used for energy modeling in our simulation.

In order to evaluate the role of number of dominators in energy dissipation, we need to compare energy dissipation in the entire network in aggregation-CDS with degree CDS. Consider the energy dissipation of nodes in network represented as E_{dom} for nodes having dominator's role and $E_{non-dom}$ for the non-dominators. The non-dominators nodes spend energy $E_{non-dom}$ to communicate the sensed data to nearest dominator at distance d within direct transmission radius r_{max} and therefore obeys Friss free space propagation model having attenuation d^2 with coefficient (α_{friss}). Let E_l be the per bit energy dissipation of transceiver electronics. In order to transmit a message of m-bits at a distance d, the non-dominator expends energy:

$$E_{non-dom} = m.E_l + m.\alpha_{\text{friss}}.d^2 \tag{5.1}$$

Let the dominators dissipate energy E_{dom} in $i)$ receiving information from dominatees (E_l), $ii)$ performing aggregation (E_{agg}) and $iii)$ transmitting aggregate data to base station ($\alpha_{2-ray}.d^4$). It may be noted that the average distances d between dominator and base station is much greater than maximum transmission radius r_{max}. Thus, the network nodes have two modes of communication i.e higher range communication (beyond $d > r_{max}$) and multi-hop communication. Using opportunistic routing if multi-hop energy dissipiation greater than higher range direct transmission energy then higher range transmission is used which follows 2-ray propagation model with attenuation d^4. Thus, the multi-hop communication energy is upper bounded by energy dissipation of 2-ray propagation model with attenuation d^4. Thus, to transmit m-bit message after

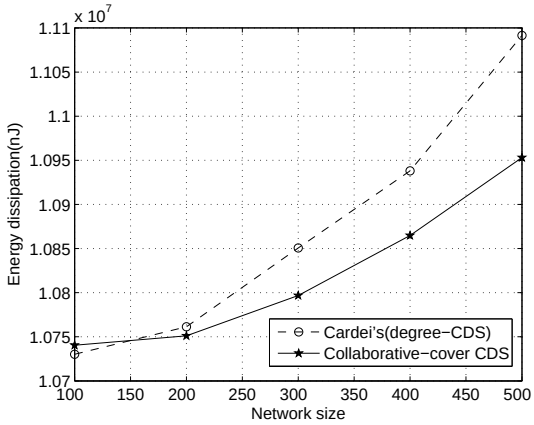


Figure 5.7: Performance comparison of aggregation energy dissipation with degree-CDS algorithm

aggregating data from its dominatees in its neighbourhood say $|\mathrm{Nbd}|$, the radio energy E_{dom} expends:

$$E_{dom} = m.E_l.|\mathrm{Nbd}| + m.E_{agg}.|\mathrm{Nbd}| + m.\alpha_{2-ray}.d^4 \tag{5.2}$$

Thus, energy dissipation of a dominator and its dominatee is given by:

$$E_{total-dom} = E_{dom} + |\mathrm{Nbd}|.E_{non-dom} \tag{5.3}$$

Therefore, total energy dissipation of network with $|\mathrm{CDS}| = k$ dominators is given by

$$E_{total} = k.E_{total-dom} \tag{5.4}$$

The equation-5.4 provides the total energy dissipation of network in communicating the sensed data to base station while performing aggregation at the dominators of CDS. Using equation-5.4,we conducted an experiment to simulate our CDS algorithm for computing the network wide energy dissipation and analyze the effect of smaller size of CDS on in-network aggregation in energy dissipation of network. We have taken a frame m of size 1000 of sensing data generated from all nodes, which is communicated by our CDS based aggregation backbone to the base station located centrally inside target area. The simulation results are captured for single round of data gathering application. We then compare the energy dissipation for single round data communication for degree based CDS[14]. The results in figure-5.7 show the crossover at the early network size of 100 nodes and beyond network size 200 onwards in our aggregation-CDS reduces the dissipation energy substantially of sensed data communication even for a single round. The reduction in the network wide energy dissipation using our aggregation-CDS results in increase of the network lifetime.

5.10 Summary

In this chapter we have described a distributed approximation algorithm for identifying a minimal size connected dominating set using the collaborative cover heuristic for which the approximation factor is at most $(4.8 + \ln 5)\mathrm{opt} + 1.2$, where opt is the size of any optimal CDS.A post-processing step identifies the Steiner nodes leading to a Steiner tree for independent set nodes. This improves upon the existing approximation for reported CDS algorithms. When our proposed CDS scheme is used for lossless in-network aggregation function shows a substantial improvement in reducing energy dissipation of network compared to degree based CDS. The message complexity of our algorithm is at most $O(n\Delta^2)$, where being the maximum degree of a node in graph and time complexity is $O(n)$.

Chapter 6

Node mobility transparent CDS construction algorithm

A connected dominating set (CDS) provides a virtual backbone in an ad-hoc network. Such networks can even have mobile nodes. We define a self organizing CDS which reconfigures itself to adapt to node mobility. Local self configuration without any manual or external control is desirable. Only nodes that are currently stationary participate in the CDS construction and reconfiguration. Node mobility is handled using three major operations: $i)$ adapting CDS to the changes in topology due to node mobility by detaching mobile nodes and self reconfiguring CDS for stationary nodes, $ii)$ maintaining coverage of mobile nodes with the CDS backbone of stationary nodes by tracking their locations and $iii)$ self reconfiguring the CDS when a mobile node becomes stationary. We have modified the local source independent multipoint relay (MPR) based CDS contruction technique for adapting to node mobility. For optimizations, we developed a Markov model for a weighted CDS to reduce location updates in tracking of mobile node. The complexity of our mobile node tracking algorithm is at most $O(d \log d)$, where d is the number of boundary crossings while a single node moves. Simulation results indicate that our mobile node tracking algorithm achieves 40% reduction in location updates using weighted CDS compared to shortest hop tracking path to CDS.

6.1 Introduction

Wireless sensor networks open up new applications in remote monitoring of environment, habitat, etc. Issues such as energy constraints, coverage preservation and topology control play an important role in the design of protocols for sensor networks. Robotic mobility (or mobility) is the ability of the nodes in sensor networks to move under electronic control without human assistance. Node mobility opens up possibilities to overcome some constraints, such as coverage preservation and energy replenishment. Using controlled deployment based on node mobility sensor nodes can be parked at optimal locations for remote geographical monitoring after their initial deployment. Similarly, for controlled network maintenance, sensor nodes can move locally to merge multiple connected components in network to a single connected component. In controlled energy harvesting, nodes move to recharge their energy resource from nearby resource center.

These networks lack the network infrastructure for their connectivity and control operations. A connected dominating set [11] is used to provide a virtual backbone to a sensor network for efficient routing and broadcasting. A dominating set D of a graph $G = (V, E)$, is defined as a subset of V such that each node in $V - D$ is adjacent to at least one node in D. Dominating sets are often chosen such their members are pairwise independent or not within range of direct wireless communication. For this reason a additional set of connector nodes (C) are also taken to ensure that the subgraph induced by the dominators and connectors is connected. Such a set of dominators and connectors ($D \cup C$) is called a connected dominating set (CDS). A CDS of small size often simplifies network control operations which confines network backbone operations to the few CDS nodes, leading to advantages such as energy efficiency and low latency. It also supports mobility, as we shall see through this work. A multipoint relay (MPR) [63] of a node is defined as its forwarding node set which covers its 2-hop neighbours. Recently, Adjih, Jacquet and Viennot [12] introduced a CDS construction based on source independent MPR which is localized and generates a small CDS [12, 21].

Mobility of nodes that are in the CDS can disturb the CDS. The connectivity of a mobile node with the rest of network changes due to its movement. It is, therefore, desirable to have an efficient mechanism to handle node mobility. So long as a node is not moving, it behaves like any other fixed node. A node is treated as a mobile node only when it is moving. When a member of the CDS becomes mobile, it becomes necessary to reconfigure the CDS to sustain coverage. Similarly, when a mobile nodes halts, it may be necessary to extend the CDS to cover it. While a node is on the move, it has to be tracked if its connectivity is to be ensured. It is assumed that any mobile node is attached to some dominator in the CDS at most 2-hops away. If this condition is violated, then the mobile node becomes unreachable from the network. We show that it is possible to utilise redundant coverage of the mobile node by CDS nodes to avoid

frequent reconfiguration of the CDS as a mobile nodes moves around. In this context we show that our adaptive approach has some advantages over the general self stabilization paradigm, such as quick reconfiguration, contained reconfiguration and reduced state updates.

Several algorithms have been reported for finding CDS in ad hoc networks and a few of them are self stabilizing. Many tracking techniques based on hierarchical structures are reported in the literature for tracking of mobile objects [20]. However, to our knowledge, there is no reported technique that considers the aspect of coverage preservation during node mobility. In this work we describe a CDS construction technique featuring efficient and transparent (to the extent that no centralised intervention is required) self configuration and adaptation to node mobility.

The contributions of this work are as follows.

1. We have developed a self organizing MPR based CDS for the sensor network. When any node becomes mobile, the rest of the network self reconfigures the CDS locally. Similarly, when a mobile node halts, it is allowed to join the fixed network. Again, the network self reconfigures the CDS locally. Reconfiguration can be in $O(n\Delta^3)$ time. We are unaware of any self organizing MPR based CDS scheme being reported in the literature.

2. We have adapted a reported technique for tracking mobile nodes. Our scheme enables making location updates of mobile nodes by some dominator in the CDS. Location updates by our algorithm is done in $O(d \log d)$ time, where d is the number of boundary crossings while a single node is moving. We have developed an optimization technique based on the Markov chain model to assign weights to the CDS tree to reduce the time taken for self reconfiguration and making location updates using our adapted technique for tracking of mobile nodes. The weighted CDS is shown to reduce the number of location updates for the highest weight path compared to the tracking scheme based on the shortest hop path. Simulation results indicate a reduction of 40% in location updates using our Markov chain heuristic.

The rest of chapter is organized as follows. We present preliminary concepts and a survey of related work in section 6.2. Formulation of the problem is described in section 6.3. In section 6.4 we describe our self configuring MPR based CDS construction algorithm. In section 6.5 we present an analyse our algorithm. Our scheme for tracking of mobile nodes and making shortest path based location updates is described in section 6.6. Section 6.7 presents our optimisation scheme for making location updates using weighted CDS based on the Markov model. Simulation results are discussed in section 6.8. We close the chapter in section 6.9 with a summary of the work.

6.2 Background and related work

The problem of transparent node mobility in networks with a CDS based backbone mainly deals with self configuration of the CDS and tracking mobile nodes to maintain their location information and ensure their connectivity with the network. We survey reported work on CDS construction and tracking of objects to motivate the need of our approach in this work. The related works on tracking of mobile objects can be classified as: $i)$ mobility profile based object tracking and $ii)$ online object tracking, based on history information of mobile object or current information for its location updates.

In the next few subsections we do a review of techniques related to basic CDS construction, multipoint relays, MPR based CDS algorithms and mobile object tracking schemes.

6.2.1 Basic CDS construction

Use of a CDS as a virtual backbone was first proposed by Ephremides in 1987 [11]. Since, then many algorithms for CDS construction have been reported. These can be classified as $i)$ centralized, $ii)$ distributed and $iii)$ localized algorithms, based on network wide information or local information for its construction. Guha and Khullar [64] first reported a two 2-phase centralized greedy algorithm for general graphs having approximation ratio $O(\log \Delta)$, Δ being the highest degree of a node in the graph. Ruan designed a 1-phase greedy with performance ratio at most $3 + \log \Delta$. Cheng proposed 4-phase greedy algorithm for minimal connection dominating set (MCDS) construction for UDGs.

A distributed algorithm is more suited in adhoc networks as that depends on local information only. Das [46] reported two distributed algorithm of greedy approach. Wan [13] described a single initiator, MIS based distributed algorithm for UDGs of at most $8\mathrm{opt} + 1$, $O(n)$ time complexity and $O(n \log n)$ message complexity. Here opt is the cost of any optimal CDS and n is the number of nodes in the graph. Cardei [14] improved it using degree based MIS and Steiner tree based connectors algorithm to identify a CDS of size at most $8\mathrm{opt}$ for growing from single leader having $O(n)$ message complexity and $O(n\Delta)$ time complexity using only 1-hop neighbourhood information. Li [16] has reported technique to construct a CDS of size at most $(4.8 + \ln 5)\mathrm{opt} + 1.2$ using Steiner connectors. Wu and Li [21] first proposed fully distributed using 2-hop information for pruning.

Recently Adjih [12] and Wu [21] reported a local approach for small size CDS construction based on multipoint relays. Extended MPR k-hop ($k \leq 3$) local information

based small size connected dominating set construction has been proposed [21]. We now report a localised CDS construction technique to support mobility with the properties of quick convergence and self configuration.

6.2.2 Multipoint relays (MPR)

A multipoint relay is a local dominating set of a node to cover its 2-hop neighbours. A multipoint relay set has the property that each 2-hop neighbour of the node has a neighbour in the multipoint relay set. The multipoint relay set plus node forms a dominating set of 2-hop neighbour of the node. In order to define it formally in graph theory, we give the following definitions. Let $\mathrm{N}[V]$ be the set of all nodes in a given set V or have a neighbour in V.

Definition 6.1 (Cover) *V covers a set W when $W \subset \mathrm{N}(V)$.*

Let $\mathrm{N}_1(V)$ be the nodes at distance-1 from V, then $\mathrm{N}_1(V) = \mathrm{N}[V] - V$. Let $\mathrm{N}_2(V)$ be the nodes at distance-2 from V, then $\mathrm{N}_2(V) = \mathrm{N}[\mathrm{N}[V]] - \mathrm{N}[V]$.

Definition 6.2 (MPR) *multipoint relay (MPR) set is defined in any of following ways:*

- *A MPR set M is a dominating set of the subgraph induced by $\mathrm{N}[\mathrm{N}[V]]$*
- *A MPR set M is any subset $M \subseteq \mathrm{N}_1(V)$ such that $\mathrm{N}_2(V) \subset \mathrm{N}(M)$*
- *A MPR set M is a subset of neighbours that covers the 2-hop neighbourhood of V.*

The problem of computing a multipoint relay set (MPR) for a given graph $G(V, E)$ with minimum size is NP-Hard [12].

6.2.3 MPR based CDS Algorithms

In a pioneering work, Adjih [12] redefined source dependent MPR to source independent MPR and reported a novel localized algorithm for constructing MPR based CDS which is source independent. This approach is later modified by Wu [21] and referred to as enhanced approach for CDS based on MPR.

Both MPR based CDS construction approaches are now described.

Approach by Adjih, Jacquet and Viennot The following greedy algorithm can be run on each node to construct locally its MPR sets, which is also called as local dominating set of a node.

Algorithm 9 Greedy algorithm for MPR

1: **repeat**
2: Add $u \in \mathrm{N}_1(v)$ to $M(v)$, if there is a node in $\mathrm{N}_2(v)$ covered only by u.
3: Add $u \in \mathrm{N}_1(v)$ to $M(v)$, if u covers the largest number of nodes in $\mathrm{N}_2(v)$ that have not been covered.
4: **until** all $\mathrm{N}_2(v)$ nodes are covered

The source independent MPR based algorithm-9 is used for CDS construction. The CDS construction scheme uses the following two rules to determine whether a node belongs to a CDS.

Rule-1 the node has a smaller ID than all its neighbours or

Rule-2 it is multipoint relay of its neighbour with the smallest ID

The set of nodes selected by rule-1 and rule-2 forms a CDS. Applying rule-1 and rule-2 to the greedy algorithm generates a smaller CDS.

Enhanced approach by Wu Wu [21] pointed out two drawbacks in the source independent MPR: $i)$ the nodes selected by rule-1 are not essential for a CDS and $ii)$ the greedy algorithm does not take advantage of rule-2. Wu's extended approach comprises of enhanced rule-1 and the extended greedy algorithm, which are given below.

> *Enhanced rule-1:* the node has a smaller ID than all its neighbours and it has two unconnected neighbours.

Wu [21] proved that enhanced rule-1 together with the original rule-2 generates a CDS except when the graph is complete. The extended greedy algorithm is used by each node v for identifying multipoint relays. Here node u is a free neighbour of v if v is not the smallest node ID neighbour of u.

6.2.4 Mobile object tracking schemes

Online object tracking using hierarchy of regional directories was given by Awerbuch and Peleg [20], which limit the location updates work after an object moves. Since,

Algorithm 10 Extended greedy algorithm for MPR

1: **repeat**
2: Add all free neighbours to $M(v)$
3: Add $u \in \mathrm{N}_1(v)$ to $M(v)$, if there is an uncovered node in $\mathrm{N}_2(v)$ covered only by u
4: Add $u \in \mathrm{N}_1(v)$ to $M(v)$, if u covers the largest number of uncovered nodes in $\mathrm{N}_2(v)$ that have not been covered by the current $M(v)$. Use node IDs to break a tie when two nodes cover the same number of uncovered nodes
5: **until** all $\mathrm{N}_2(v)$ nodes are covered

then several techniques on object tracking that use a variants of hierarchical structure for tracking of mobile objects in sensor networks have been reported in the literature. The aim of tracking mobile objects is to handle queries related to their location using a distributed indexing structure. A scheme reported in [65] uses self stabilizing hierarchical tracking service for tracking of mobile objects. The drawbacks of tree based indexing algorithms for object tracking in sensor networks is they have not considered the aspects of node mobility. We have considered a CDS backbone for our tracking scheme and also considered adapting CDS to the node mobility.

Algorithms for location tracking of mobile objects work better with weighted CDS trees. Mobility *profile based* object tracking schemes are based on mobility profile history to derive weights for constructing the CDS tree. The resultant tree holds property of deviation free paths from every node to sink having minimum hops. Techniques reported in [17, 18] are used to assign weights to sensor nodes with minimum object crossing rate. The minimum weight heuristics represents *deviation free* paths.

In a mobility *profile independent* object tracking scheme the weights to sensor nodes are determined without simulating object movements or using history of mobility profile. Heuristics based on Markov model [17] using geometric information determined by Voronoi diagram are used to assign the weights to sensor nodes. A tree construction algorithm based on a maximum spanning tree [19], registers saving of message transmissions in the object tracking based on crossing rates between sensor nodes. Mobility models which are independent of object mobility profile are of interest to our work. We use a Markov model that is independent to mobility profile. However, the heuristics we use for our Markov model are different to those used in [17]. We have used Markov chain using neighbourhood information to reduce the location updates for node mobility.

6.3 Problem Formulation

Consider a wireless sensor network with fixed number of nodes in which only a few nodes are mobile at a time. Assume that total number of nodes are fixed and a mobile node is allowed to move within a region so that full connectivity of the network, including mobile nodes is always possible. We assume that a CDS of the stationary nodes is available and let the location of mobile node be maintained by some node in the CDS. When a sensor node that is part of the CDS backbone moves, then the backbone may get disturbed. It then becomes necessary to detach that moving node and reconfigure the CDS to regain proper coverage. This enables the CDS topology to adapt to changes arising due to movement of some of the nodes. The mobile node needs to be tracked to ensure its connectivity to the CDS. For a mobile node to get connected to the network, it beacons messages from time to time which eventually reach some node in CDS at most at a distance-2 from the mobile node. A stationary node detects the mobile node when a node crosses into its transmission range boundary for the first time. This event is referred to as boundary crossing, it then notifies this event to its dominator of CDS for making location updates. The dominator also informs the location information to neighbouring dominators so that they know of the how the node is moving. Nodes in a wireless sensor network have energy constraints, therefore efficiency of self reconfiguration and location updates in networks for adaptation of node mobility becomes an important issue.

Let $G = (V_{\text{stat}} \cup V_{\text{mob}}, E)$ be a connected graph, where $(V_{\text{stat}} \cup V_{\text{mob}})$ are set of static and mobile nodes, respectively and E is a set of edges.

Definition 6.3 (Adaptive CDS (ACDS)) $\text{ACDS}(G)$ *is a subset of nodes of* V_{stat} *satisfying these three properties:*

1. *It dominates* V_{stat}
2. *The subgraph induced by it is connected*
3. *A mobile node* V_{mob} *is at most at a distance-2 from some node in it.*

Thus, ACDS(G) is 2-dominating set of mobile nodes, meaning that any mobile node is at most at a distance of two from some node in the ACDS, the distance being measured as the number of edges in any shortest path.

We consider some important differences between ACDS and a self stabilizing CDS. In ACDS, when a node starts moving it may be removed from the fixed network. When a node becomes stationary, it may become a dominator. In ACDS, location updates are triggered when a mobile node crosses a node boundary. Reconfiguration can be

done in $O(n\Delta^3)$ time. In case of self stabilisation, convergence to legitimate state is the main goal. Moving node triggers the self stabilization procedures to kick in restore to legitimate. Stabilisation is done in $O(n^2)$ time for every movement of mobile nodes [66].

In the next two sections we describe algorithms for adaptive CDS construction and mobile node tracking.

6.4 Self configuring MPR based CDS construction

The main algorithm for mobility adaptive CDS reconstruction is given as algorithm-13. It relies on algorithm-11 for identifying potential MPRs and on algorithm-12 for MPR based CDS construction. Both these secondary algorithms are adapted from related reported techniques in the literature[12, 21] and can be applied to a set of nodes so that they work only in a particular region.

We first describe the construction mechanism for the self configuring MPR of a node x. After that we describe the self configuring CDS algorithm using MPR and finally we describe the main algorithm for mobility adaptive CDS reconstruction.

6.4.1 Construction of the self configuring MPR of a node

Let us consider only one node x, then $\mathrm{N}_1(\{x\})$ is the distance-1 neighbourhood of x and $\mathrm{N}_2(\{x\})$ is its the distance-2 neighbourhood. A MPR set M of a node x is any subset of $\mathrm{N}_1(\{x\})$ such that $\mathrm{N}_2(\{x\}) \subset \mathrm{N}(M)$. Thus, M is a dominating set of the subgraph induced by $\mathrm{N}(\mathrm{N}(\{x\}))$. The multipoint relay selector of a node y is a node which selected node y as a multipoint relay. For a given node x, let v be a distance-2 neighbour of x. We give a simple mechanism to compute MPR set of x and its selector.

For each pair of nodes x and $v \in \mathrm{N}_2(\{x\})$, compute the set of nodes that are 1-hop neighbours of both x and v, by finding the intersection of the neighbourhoods of x and v to give a set $S = (\mathrm{N}_1(\{x\}) \cap \mathrm{N}_1(\{v\}))$. If node $u \in S$ has the smallest identifier among them, then u becomes an element of MPR set of x, whereas nodes x and v are both added to the MPR selector set of u. It may be noted that a node may have multiple selectors. Thus, node x selects nodes u as its MPR if the selector of u is x and other selector, say v is its 2-hop neighbour. Thus, every MPR node u has one or more pairs of MPR selectors $\{\ x, v \mid u$ is the minimum ID node of the 1-distance neighbourhood intersection x and $v\}$. Given node x, we can find its MPR nodes which are its distance-1 neighbours such that they form the cover of distance-2 neighbourhood of x.

Algorithm 11 Self configuring MPR

1: Node u broadcasts message to its distance-1 neighbours.
2: Each distance-1 neighbour say w forwards it to distance-2 neighbours v of u along with node IDs of forwarding nodes.
3: The distance-2 neighbours v of u after receiving all of them selects the lowest node ID forwarding node w as the MPR node.
4: Selected MPR node w keeps the list of selectors(u, v) for MPR node.
5: Nodes u and v record w as its MPR.

6.4.2 Self configuring CDS algorithm using MPR

The self configuring algorithm for generating a CDS of G using MPR is based on the following principles:

1. Node x is in CDS as a lowest ID dominator, if it has smaller ID than all its neighbours in $\mathrm{N}(x)$

2. Node u is in CDS as a connector, if it is an MPR of two or more dominators, As a result of becoming a connector u may connect two or more dominators belonging to different components. Distinct components thus getting connected need to resolve their identities – by taking on the ID of the lowest component

3. Node u also takes up the role of a CDS connector if it later detects that it is an MPR for two or more distinct components

6.4.3 States of a node and information to be stored

Each node of the network can be in one of several states, which can be captured in a local variable called **STATE** which can assume one of the following values:

`IDL` The node is in an idle state.

`DOM` The node is a lowest ID dominator. In this case the node needs to record the following:

- its relays, which may be maintained in a list **MPRList_ofDom**
- its dominatees, which may be maintained in a list **NDMList_ofDom**

`CON` The node is an MPR based connector. In this case the node needs to record the following:

Algorithm 12 CDS construction using MPR

1: Nodes in $V(G)$ broadcast their node ID so that each node in its distance-1 neighbourhood can know the *lowest* ID node.
2: **if** *node finds that its node ID is the lowest* **then**
3: Node becomes dominator sets STATE=DOM and broadcasts its dominator announcement.
4: Sets its ID as its component ID
5: **end if**
6: Distance-1 neighbours on receiving *dominator* message first time becomes *dominatee*, sets STATE←non_dominator.
7: **if** *dominatee* w *is an MPR of dominator* u *(** $u \in$ *selector(*w*) *)* **then**
8: Add u to DomList_ofMPR of w
9: **if** $|$*DomList_ofMPR of* $w| > 1$ (* w is a non-trivial relay *) **then**
10: Node w sets STATE←CON
11: Update component ID as a result of adding the new MPR connector
12: **end if**
13: **end if**
14: **if** *dominatee* w *is an MPR of* u *(** $u \in$ *selector(*w*) *)* **then**
15: Any dominatee w finding itself connecting distinct components also becomes a non-trivial relay and set STATE←CON
16: **end if**

- the dominators it connects, which may be maintained in a list **DomList_ofMPR**
- its own dominator in **myDom**

`NDM` The node is a dominatee and its dominator is recorded in **myDom**

`MOB` The node is mobile.

Thus the variables that are required are **STATE**, **MPRList_ofDom**, **NDMList_ofDom**, **DomList_ofMPR** and **myDom**. Each node has its unique **ID**. In addition, some working variables such are **lowest_ID**, **selectorList** are also be required. A node also needs to record the ID of the component of which it is a member. By hypothesis complete connectivity of the network is possible. However, during the process of CDS construction distinct components could be present which are eventually merged to a single component.

6.4.4 Adapting the CDS to accommodate mobility

Initially, the network configures itself using the MPR based CDS algorithm. As a result a set of independent nodes are identified as dominators and a set of MPR based connec-

tor nodes are identified to serve as connectors for these dominators. This constitutes a stable state of the network.

The dominators and connectors together constitute the CDS backbone of the network. When a node starts moving, the CDS could get disturbed, in which case there is a period of metastability of the network, as the CDS reorganises itself. In order to handle reconfiguration of the CDS in the wake of a node exhibiting mobility, we consider the cases listed below and actions necessiated thence. As a result of the actions, the network returns to a stable state with a reconfigured CDS.

Case: A stationary node starts moving:

- *Mobile node was a dominator:* The mobile node u is aware that it is a dominator node. As a result of its movement the CDS in its region will be severely disturbed and will require reconfiguration. A new dominator will have to be identified and MPR nodes for this new dominator will also have to be identified. Each MPR connector (`CON`) node $w \in \mathrm{N}(u)$ will remove u from its DomList_ofMPR if $u \in$ DomList_ofMPR of w. If, as a result, for w, |DomList_ofMPR| ≤ 1, then w will cease to be a non-trivial relay.

  All nodes in $\mathrm{N}(u)$ will be triggered to configure themselves as MPRs for new dominators that will be identified in $\mathrm{N}(u)$. However, non-trivial MPR relays will continue in their earlier roles, as the departure of u will not disturb their status. This will be followed the MPR based CDS construction in the region of $\mathrm{N}(u)$. Thereafter, the coverage and connectivity will be restored in $\mathrm{N}(u)$ and the overall CDS will once again become stable.

- *Mobile node was an MPR connector:* The mobile node w is aware that it is a non-trivial MPR based connector for two or more dominator nodes. Each dominator nodes u, where $u \in$ DomList_ofMPR of w, which was connected by w will have to identify new MPR based connectors for themselves. Note that each node u of these dominators will continue to be then minimum ID node in $\mathrm{N}(u)$. Our initial assumption is the underlying graph will not get disconnected. Therefore, for each such node u, all the nodes in $\mathrm{N}[u]$, will have to go through the process of forming MPR connectors with adjacent dominator nodes. However, non-trivial MPR relays in these regions will continue in their earlier roles, as the departure of w will not disturb their status. Thereafter, the coverage and connectivity will be restored in $\mathrm{N}(u)$, for each such u and the overall CDS will once again become stable.

- *Mobile node was a regular node:* This is the simplest case where the CDS is not affected and nothing needs to be done to the CDS.

Case: A mobile node halts:

- *Outside coverage of a dominator:* Let the mobile node be u. Since there is no dominator node in $\mathrm{N}(u)$, u, itself will have to become a dominator. Before that it will first initiate identification of MPR based connector nodes to adjacent distance-2 dominators and then announce itself as a dominator.

- *Within coverage of a dominator:* Let the mobile node be v and let dominator of the area where it halts be u. If v has a higher ID than u, then obviously nothing needs to be done. However, even if v has a lower ID than u, a round of MPR identification followed by dominator identification in $\mathrm{N}[u]$ can be skipped and v can simply join as a dominatee of u. However, in any future event where dominator selection is required in $\mathrm{N}[u]$, u will participate in the usual ID based resolution mechanism for dominator identification.

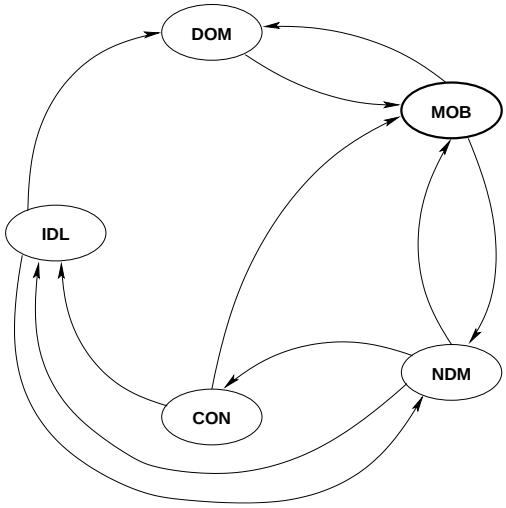


Figure 6.1: State transition diagram for self configuration

6.5 Analysis of technique

The following subsections deal with the correctness of the technique and its complexity analysis.

6.5.1 Correctness of MPR based self configuring CDS construction

In the MPR based CDS construction described here, nodes with minimum ID become dominators in their neighhourhood. If nodes u and v are such that $\mathrm{N}(u) \cap \mathrm{N}(v) \neq \emptyset$, then algorithm-11 will find MPR based connectors for u and v and u and v (and their connector and dominatee nodes) will then become members of the same component.

Algorithm 13 Self reconfiguration of CDS

```
1: if stationary node is about to become mobile then
2:    if it is a dominator (* STATE=DOM *) then
3:       broadcast message "DOM node with ID going MOB"
4:    else if it is a connector (* STATE=CON *) then
5:       broadcast message "CON node with ID going MOB"
6:    else if it is a regular node (* STATE=NDM *) then
7:       broadcast message "NDM node with ID going MOB"
8:    end if
9:    STATE ← MOB
10: else if mobile node is about to become stationary then
11:   broadcast message "MOB node with ID going NDM"
12:   if it is within coverage of a dominator (* determined by receipt of appropriate message
      *) then
13:      STATE ← NDM
14:   else
15:      STATE ← DOM (* no reply received from a dominator, it is must itself become one *)
16:      initiate identification of MPR connectors
17:   end if
18: else if node is a stable dominator node (* STATE=DOM *) then
19:   if message received is "CON node with ID going MOB" then
20:      if ID ∈ MPRList_ofDom then
21:         remove ID from MPRList_ofDom
22:         if ID ∈ NDMList_ofDom then
23:            remove ID from NDMList_ofDom
24:         end if
25:         initiate MPR identification by algorithm-11 and non-trivial relay identification by
            algorithm-12
26:      end if
27:   else if message received is "NDM node with ID going MOB" and ID ∈ NDMList_ofDom
      then
28:      remove ID from NDMList_ofDom
29:   end if
30: else if node is a stable connector node (* STATE=CON *) then
31:   if message received is "DOM node with ID going MOB" then
32:      if ID ∈ DomList_ofMPR then
33:         remove ID from DomList_ofMPR
34:         if |DomList_ofMPR| ≤ 1 then
35:            STATE ← NDM (* DomList_ofMPR dropped, to participate in algorithms 11 &
               12 *)
36:         end if
37:      end if
38:   end if
39:   node responds to messages to participate in algorithm-11 to become a potential connector
40: else if node is a stable idle node (* STATE=IDL *) then
41:   if message received is "DOM node with ID going MOB" and ID = myDom then
42:      STATE ← IDL
43:   end if
44: end if
```

In this process individual components will grow until no further joining of dominator nodes of the components by MPR based connectors is possible.

After this process let there be distinct components $C_1, C_2, \ldots, C_p$. At this stage line-14 of algorithm-12 will kick in. By the hypothesis that connectivity of the nodes is possible, there must exist components C_i and C_j, each having a set of dominatee nodes $V_i \subset C_i$ and $V_j \subset C_j$, such that for each $v_i \in V_i, \exists v_j \in V_j$, such that $v_i \in \mathrm{N}(v_j)$ and vice versa. Thus the minimum ID nodes of V_i and V_j will be chosen as an MPR based connector by algorithm-11 to connect components C_i and C_j, leading to their merger. In addition to the minimum ID nodes, locally minimal ID nodes will also be chosen. This process will continue until there is only one component left.

6.5.2 Correctness of self reconfiguration

The correctness of our technique for CDS self reconfiguration in the presence of node mobility is essentially based on the correctness of our basic MPR based CDS construction technique. When a dominator node u becomes mobile, a new set of dominators D is chosen in $\mathrm{N}(u)$. Similarly, when a connector node w becomes mobile we consider the set of dominator nodes D for which w was a connector. In each case a subset of nodes of D may get isolated from the network and form new components. By the arguments given in the previous subsection, these dominators will be able to identify MPR based connectors to connect them to the main component. Otherwise, MPR based connectors will be identified to connect the other dominators directly to distance-2 dominators in the main component.

6.5.3 Complexity Analysis of Algorithm

The time complexity of the MPR computation is $O(\Delta^3)$, where Δ is the maximum degree of a node in the graph. The time complexity of CDS computation is at most $O(n\Delta^3)$, where n is the number of nodes in the network. The size of the CDS computed by the algorithm is not optimal, but the algorithm is self configuring.

6.6 Tracking of mobile node and location update using CDS

Our tracking structure is a CDS. Each node in the tracking path has at most one non-dominator. Thus, the tracking path has two variants: $\mathrm{Path} : \{\mathrm{mobile\_node} \rightarrow$

$\mathrm{non\_dominator} \rightarrow \mathrm{dominator}\}$ or $\mathrm{Path} : \{\mathrm{mobile\_node} \rightarrow \mathrm{dominator}\}$. Nodes in the path generally point to nodes that are closer to the mobile node. Assume that the network always maintains a valid CDS (except during the transitory phase of reconfiguration). Thus, nodes in network can be classified in two types as: non_dominators or dominators. Each non-dominator node also records its dominator. A mobile node obtains its neighbourhood information either through hello messages or through idle listening messages. The mobile node then finds out all the 2-hop paths referred to as tracking paths. When the tracking path has a direct connection to the dominator, the location of the mobile node with its timestamp is registered in the dominator whereas the mobile node records the ID of the dominator. Alternatively, when the tracking path has an intermediate node to the dominator, the mobile node stores the information of intermediate node which acts as relay to dominator as a virtual dominator. Thus, the CDS along with *virtual dominators* forms the connected dominating set of network in the presence of mobile nodes. Note that when a mobile node moves out of the region of current virtual dominator to the region of another virtual dominator under the same dominator, the mobile node updates only its relay node information. Whenever the mobile node moves the region of a different dominator, it updates not only its dominator and virtual dominator, but also registers its *location* and *timestamp* with its neighbouring dominators at most at a 2-hop distance.

We describe two ways to achieve the node mobility adaptive CDS.

1. *Shortest tracking path in CDS where distance means number of edges in path between mobile node and some node in CDS*
2. *Tracking path in weighted CDS having highest weight dominator which is at most distance-2 away from a mobile node*

In the next subsection we describe the tracking of mobile node by the CDS of the stationary network. In section 6.7.1 we describe a more efficient scheme based on Markov chain modelling.

6.6.1 Shortest tracking path in CDS

We define shortest tracking path as the shortest path between mobile node and some dominator node in the CDS (at most distance-2 away), where distance is measured by number of edges involved in the path. The mobile node can receive beacons from either a non-dominator or a dominator node in the CDS. The non-dominator beacons also gives the information of its distance-1 dominator node in CDS. Therefore, a tracking path consists of at most one non-dominator and a dominator. As the mobile node moves to a new position, it often crosses the communication range boundary of the current

tracking node and also enters in the range of new node. This triggers in a change of tracking path as the following way.

1. *Non-dominator is changed for a given dominator:* For a given dominator in tracking path, if only the non-dominator node is changed then the mobile node only updates its non-dominator on the path to the dominator.

2. *Dominator is changed:* When the dominator in the tracking path changes, it updates its neighbouring dominators of the current tracking path. Therefore, CDS dominator nodes need to keep the location information of the mobile node. Algorithm-14 for tracking path gives the details of creating tracking path to compute the location updates of mobile node.

6.7 Tracking of mobile node using weighted CDS

In this section first we describe Markov chain based scheme for assigning weights to the CDS leading to a weighted CDS. We then use the weighted CDS, for determining highest weight tracking path to optimize the reconfiguration by reducing the number of location updates of the mobile node.

6.7.1 Markov Model for weighted CDS

In this section we will present an improvement on the shortest path tracking technique discussed in the previous subsection, to track a mobile node while making fewer location updates. We assume a discrete time stochastic process defined over a set of states in terms of a matrix of transition probabilities. Thus, the time values are considered discrete which advance only there are state changes. We assume a CDS forms a backbone of the fixed network. The fixed network (with a stationary backbone) tracks the mobile sensor node and reports back to the backbone nodes. In order to avoid redundant reporting, we assume that only the sensor closest to the moving node is supposed to report. We assume that deployment area is covered and the underlying graph $G(V, E)$ is connected, where V is the set of sensor nodes and edge $\langle i, j\rangle \in E \forall i, j \in V$, iff $||i, j||_2 \leq \rho$, for maximum transmission radius ρ. ρ may be treated as the unit distance. G is also assumed to include the mobile nodes. It can be used not only to track the mobile node but also compute the mobility profile by labeling backbone nodes with a weight $w_{i,j}$ that represents the crossing rate of moving nodes between backbone nodes d_i and d_j.

Consider a CDS of N nodes of a (large) wireless sensor network. The dominators separated by at most 2-hop distances and having common connectors form neighbour-

Algorithm 14 Shortest tracking path using CDS

1: find shortest path between mobile node m_i and any dominator d_j in the CDS (at most distance-2 away).
2: **if** $dominator_{mobile} = \perp$ and shortest_path(m_i, d_j) < 2 **then**
3: mobile node updates the $dominator_{mobile} = d_j$ and informs to dominator.
4: Dominator updates the new location of the mobile node in $mobile\_loc \leftarrow mobile\ node\text{-}ID$. It also informs to its neighbouring dominators.
5: **else if** shortest-path(m_i,d_j)=2 (* a non-dominator nd_{ij} node exists on the path between the mobile node and the dominator *) **then**
6: Set $on\_path \leftarrow nd_{ij}$ at m_j, where nd_{ij}=on-path(m_i,d_j)
7: **else if** shortest-path(m_i,d_j)=1 (* no intermediate node *) **then**
8: Set $on\_path \leftarrow \perp$
9: **end if**
10: **if** on_path node is changed between m_i, d_j and $dominator_{mobile} = d_j$ is unchanged **then**
11: Set $on\_path \leftarrow nd_{ij}$, where nd_{ij}=on-path(m_i,d_j)
12: **end if**
13: **if** the shortest path is changed to different dominator (* $dominator_{mobile} \neq d_j \wedge$ shortest_path(m_i, d_j)=1 *) **then**
14: Update state $dominator_{mobile} = d_j$ and $on\_path \leftarrow \perp$ and informs to the dominator for its location update at dominator.
15: Dominator updates the new location of the mobile node as $mobile\_loc \leftarrow mobile\ node\text{-}ID$.
16: Dominator sends location update to its neighbouring dominator.
17: **else if** the shortest path is changed to different dominator (* $dominator_{mobile} \neq d_j \wedge$ shortest_path(m_i, d_j)=2 *) **then**
18: Update state $dominator_{mobile} = d_j$ and $on\_path \leftarrow nd_{ij}$, where nd_{ij}=on-path(m_i,d_j) and informs to the dominator for its location update at dominator.
19: Dominator updates the new location of the mobile node in $mobile\_loc \leftarrow mobile\ node\text{-}ID$.
20: Dominator sends location update to its neighbouring dominator.
21: **end if**

hoods in the CDS. Suppose a dominator d_i has k neighbouring dominators. By neighbouring dominators, we mean that dominators which are at most at a distance of 2-hop and having a common connector.

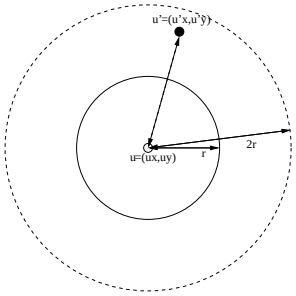


Figure 6.2: Uniform distribution region of node mobility

Let $X(t)$ denote the stochastic process to model the mobility of mobile nodes. The random variable X is uniformly distributed over the sub-interval of $(0, 1)$ such that probability density function is computed by considering the radial uniform random movement of mobile node between dominator d_i and its neighbouring dominators d_j ($1 \leq j \leq k$), where k is constant represents the maximum number of (at most distance-2) neighbouring dominators in UDG. Let each dominator d_i know the set of its neighbouring non-dominator nodes Q_j which it dominates and the sum of non-dominators of its neighbouring dominators p represented by $\sum_{1 \leq p \leq k} |Q_p|$.

The probability $p_{i,j}$ that a mobile sensor node moves out of the duty area of a backbone node d_i and moves into the duty area of backbone node d_j ($1 \leq j \leq k$), given the fact that mobile node is currently under the surveillance of sensor node d_i is computed as

$$p_{i,j} = \begin{cases} \frac{|Q_j|}{|Q_i| + \sum_{p=1}^{k} |Q_p|} & \text{if } (j_x - i_x)^2 + (j_y - i_y)^2 \leq (\pi.\rho)^2 \\ \\ 0 & \text{otherwise} \end{cases}$$

Note that $\sum_j p_{i,j} = 1$. We have no knowledge of the next move of the mobile node knowing its past history and current position of the node. Thus, for any $t \geq 0$ yields the following transition probability.

$$p_{i,j} = \Pr[X_{n+1} = j | X_n = i] \tag{6.1}$$

The state transition probabilities for this process may be obtained as a matrix called

as *transition probability matrix.*

$$M = \begin{Vmatrix} p_{1,1} & p_{1,2} & \dots & p_{1,N} \\ p_{2,1} & p_{2,2} & \dots & p_{2,N} \\ \vdots & \vdots & \dots & \vdots \\ p_{N,1} & p_{N,2} & \dots & p_{N,N} \end{Vmatrix} \tag{6.2}$$

Thus, for a Markov chain, M represents the probability that tracking system in state i will enter state j at the next transition.

Let $p_i^n = \Pr[X(n) = i]$ be probability that the process is in state i at time n, for any integer $n \geq 0$ and connected dominating set $i \in \{1, \dots, N\}$. Suppose this process satisfies the Markov property that $X(m) = j$ given that $X(n) = i$, $X(n+1) = i_{n+1}, \dots, X(m-1) = i_{m-1}$ depends on current state only and not on the history. Thus, $X(m-1) = i_{m-1}$. The state transition probabilities after m moves can be defined by m-stage transition probabilities denoted as $p_{i,j}^{(m)}$. The m-stage transition probabilities $p_{i,j}^{(m)}$ can be computed by using Chapman-Kolmogorov equation.

$$p_{i,j}^{(m)} = \sum_{k=0}^{M} (p_{i,k}^{(r)} . p_{k,j}^{(m-r)}), \forall (0 < r < m) \tag{6.3}$$

The properties of the *finite* state Markov chain are summarized below:

1. *Markov chain is irreducible:* this means that, it is possible to get to any state from any state in an *irreducible* Markov chain.
2. *Markov chain is aperiodic:* this means that, the return time is not fixed for any state.
3. *Markov chain is positive recurrent:* since M^m converges as $m \to \infty$, the number of times any state is entered is infinite for $m \to \infty$.

From the above properties, we observe that there exists a unique stationary distribution π (a (row) vector in M) which satisfies the equation $\pi = \pi \mathbf{M}$.

In addition, M^m converges to a rank one matrix in which each row is the stationary distribution π, i.e. $\lim_{k \to \infty} \mathbf{M}^k = \mathbf{1}\pi$, where $\mathbf{1}$ is the column vector with all entries equal to 1. Therefore, using $\mathbf{M}$ and π, the weights $w_{i,j}$ for the connected dominating set can be ascertained as $w_{i,j} = (\pi_i \times p_{i,j})$.

Each node in the CDS is assigned a weight $w_{i,j}$, which represents the mobility profile of nodes around its domination region. The higher the weight, the bigger is the

region of domination. Thus higher weights represent lower rate of crossing of mobile nodes across dominators resulting in lower degree of location updates in tracking. In the next section, we use the weighted CDS to track the mobile nodes.

Example 6.1 [Construction of Markov chain]

Consider a sensor network shown in figure-6.3 with $CDS = \{2, 4, 6\}$ and a mobile node x. Let the transition probability matrix $A = [p_1 p_2 p_1; p_2 p_6 p_3; p_5 p_3 p_4]$, where $p_i's$ are the transition probability. For a random walk of n-steps on the Markov chain converges. Using equation-6.3 we obtain the weights which are assigned to CDS. □

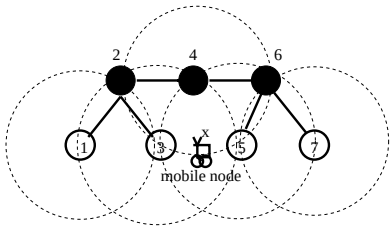


Figure 6.3: Connected dominating set with mobile nodes

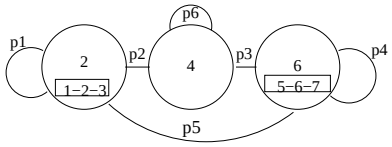


Figure 6.4: Markov chain for connected dominating set

6.7.2 Mobility tracking using weighted CDS

Algorithm-15 is meant for tracking mobile node by the CDS backbone. Let the mobile node maintain two variables: $dominator_{mobile}$ and on_path. When a mobile node moves from its current location, it maintains its connection with the backbone. The mobile node finds the highest weight distance-2 path to the highest weight dominator node in CDS and records it in as $dominator_{mobile}$. As soon as mobile node moves out from the domination range of current state $dominator_{mobile}$, it needs to self configure. A variable on_path maintains the intermediate node, if any, in distance-2 path to reach some node in CDS according to its weight. The mobile node connects to the CDS backbone via the intermediate node maintained in its on_path called as virtual dominator. At any point of time, the backbone nodes along with the virtual dominator in on_path forms the connected dominating set of the network. There are steps in algorithm-15 to correctly maintain the variables $dominator_{mobile}$ and on_path in the mobile node.

The dominator in the tracking path also records the current position and ID of a mobile node via $mobile\_loc$. When the mobile node moves to a new dominator in the tracking path, its information needs to be refreshed to remain consistent. Therefore, its new dominator sends the updates to its neighbourhood dominators.

Algorithm 15 Location update of mobile node by weighted CDS based tracking

1: **if** $dominator_{mobile} = \perp$ and exists a highest_wt_path(m_i, d_j)≤ 2 **then**
2: Set $dominator_{mobile} = d_j$ and informs to dominator.
3: Dominator updates the new location of the mobile node in $mobile\_loc \leftarrow mobile\ node\text{-}ID$. It also informs to its neighbouring dominators.
4: **end if**
5: **if** mobile m_i moved away from dominator d_j (* dist$(m_i, d_j) > 2$ *) and $dominator_{mobile} = d_j$ **then**
6: Update $dominator_{mobile} = \perp$ and informs to dominator.
7: Dominator updates the new location of the mobile node in $mobile\_loc \leftarrow mobile\ node\text{-}ID$. It also informs to its neighbouring dominators.
8: **end if**
9: **if** dominator is at distance-2 and nd_{ij}=on-path(m_i, d_j) and $on\_path \neq nd_{ij}$ **then**
10: Set $on\_path \leftarrow nd_{ij}$ and informs to dominator.
11: Dominator updates the new location of the mobile node in $mobile\_loc \leftarrow mobile\ node\text{-}ID$. It also informs to its neighbouring dominators.
12: **end if**

Consider a mobile node m_i trying to connect to some dominator $d_j \in D$. Notations used in algorithm-15 are defined as follows.

1. $highest\_weight\_2hop\_path(m_i, d_j)$: Let each dominator node $d_j \in D$ be assigned a weight w_j by the Markov Model. Let S be the set of all paths at most of 2-hop lengths (using hop count as a distance metric) between mobile node m_i and some dominator node $d_j \in D$, where D is the dominating set of G. From the set S of $2hop(m_i, d_j)$ identify a path with the highest weight w_j dominator d_j using lexicographic order $(w_j, ID(j))$.

   Lexicographic order: $(w_j, ID(j)) \geq (w_k, ID(k))$ for all $k, j \in S$ [$k \neq j$] (iff $w_j > w_k$ or ($w_j = w_k$ and $ID(j) < ID(k)$)).

2. $on\_2hop\_path(m_i, d_j, n_m)$: Assume that each node in network has a unique node ID. Let $ID(node)$ be a function returns the node ID. Let $S' \subseteq S$ be subset of all highest_weight_2hop_paths between mobile node m_i and highest weight dominator d_j, when $highest\_weight\_2hop\_path(m_i, d_j) = 2$. Among all the paths in S', let there be highest weight path with intermediate node $n_m \in \{V(G) - \mathrm{CDS}(G)\}$ having the lowest ID.

   $\exists n_m [ID(n_m) < ID(n_l), forall(n_l \neq n_m) \in S']$

Example 6.2 [Tracking of mobile node] Consider example-6.1, which assume a sensor network shown in figure-6.3 with CDS $D = \{2, 4, 6\}$ and a mobile node x and the transition probability matrix $A^n = [p_1^n p_2^n p_1^n; p_2^n p_6^n p_3^n; p_5^n p_3^n p_4^n]$, where p_i^n's are the n-stage transition probability and using the weighted CDS. Let a mobile node-x is in the radio range of nodes $\{3, 4, 5\}$. When the mobile node x leaves the boundary of node-3, it has two options for handing over its locations $\{4, 5\}$. In weighted CDS tracking path if the $\text{weight}(5) > \text{weight}(4)$ then transition to node-5 saves the location update at CDS node-4 bypassing the transition to CDS node-6. In the shortest hop path the node-4 is selected which may yield more number of location updates if the mobile node continues the crossing in the range of node-5. This example illustrates the usefulness of weighted heuristics for reducing the location updates of a mobile node. □

6.7.3 Complexity of single node tracking algorithm

The location of mobile node is maintained at CDS backbone. Any node of fixed when observes a mobile node in its region it updates its location to its dominator. The dominator on receiving the location for the first time not only records the location of moving node but also updates to its 2-hop neighbours in CDS. By update at 2-hop CDS neighbour, we mean that the previous trace of its location need to be erased in location updating. We may use the timestamp of mobile node to ensure the correct updating of current location maintained by only one CDS node at any point of time locally. Thus, for every 2^d moves the number of location updates required in d as shown in figure-6.5. The amount of work by non-dominator is at most d for node mobility of d boundary crossing. Therefore, tracking algorithm observes location updating complexity of at most $O(d. \log d)$ for tracking of single node.

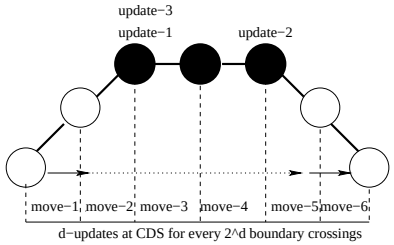


Figure 6.5: Complexity of location update for tracking single node mobility

6.8 Simulation results

We divide this section into four parts with the aim of bringing out the performance analysis of major issues in the proposed node mobility transparent CDS algorithm: $i)$

size of CDS identified by algorithm-12 for self configuration is compared to a well known CDS construction algorithm reported by Alzoubi [59], $ii)$ messages required by algorithm-12 for self configuration compared to messages required by Alzoubi's CDS construction [59] $iii)$ performance of node mobility measured using location update metric and $iv)$ comparison of updates required by Markov chain based model vs shortest tracking path

Consider a sensor network with nodes having maximum transmission range 25 units dispersed randomly in a fixed target area of 100×100. The number of nodes is varied in this experiment from 25 to 400. We assume that the network is connected. In the experiments we use an event driven simulator. Messages required to achieve self configuration in course of the simulation are monitored and counted. The size of the CDS is recorded. These experiments are conducted for algorithm-12 and Alzoubi's technique [59] to get the required data to compare CDS sizes and message counts.

Once a valid CDS is obtained, behaviour of the network nodes in response to mobility of its nodes, in order to track the mobile nodes is again monitored and necessary statistics on location updates is collected. These experiments are conducted for algorithm-14 (shortest tracking path using CDS) and algorithm-15 (Markov chain based tracking) to get the required data for making a comparison on the number of location updates made by the two techniques.

We assume a mobility model for simulating the node mobility where the mobile node makes a straight run from one end of the rectangular area to the other end. Sensor nodes placed in the linear trajectory of the mobile node are triggered when the mobile node for the first time enters their transmission range. Resulting location updates on the crossings are monitored for making the performance comparison.

The specific experiments are described below.

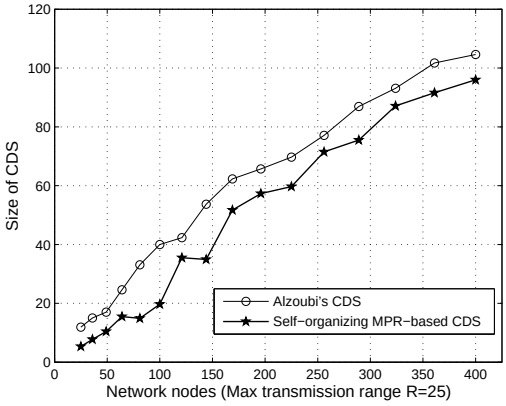


Figure 6.6: Performance comparison of CDS algorithm (transmission range is 25)

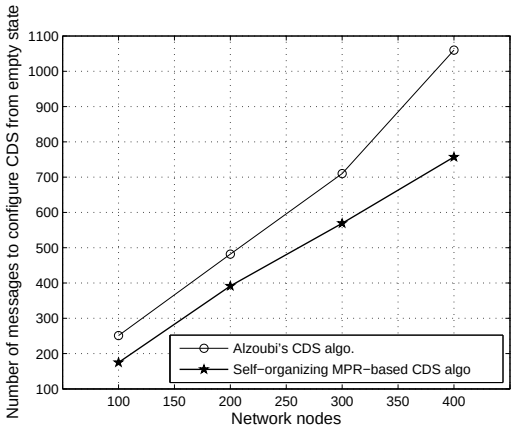


Figure 6.7: Performance comparison of messages to configure the CDS starting from initial state in algorithm-12 vs Alzoubi's CDS algorithm

1. *Size of CDS:* We observe the size of the CDS in the simulation experiments using self configuring CDS as the performance metric to compare with Alzoubi's CDS [13]. The simulation results is shown in figure 6.6 compares the CDS size identified by the proposed CDS with the Alzoubi's CDS method.

   We observe from the simulation study that CDS identified by our proposed CDS algorithm is of smaller size compared to that of Alzoubi's CDS approach.

2. We also analyze the number of messages required to configure the CDS starting from initial state when none of nodes are dominators. The performance of algorithm-12 is compared to Alzoubi's CDS in terms of messages required to configure a CDS of G. Results are shown in figure-6.7. We observe a substantial reduction in messages required.

3. *Description of mobility model:* Consider a straight run movement of mobile node which retraced back and forth between the two boundary of rectangular region. We assume a single node becoming mobile whereas rest of the network remains fixed. The nodes in fixed network observes tracking when the mobile nodes crossing inside its transmission region boundary for the first time. In order to observe effectively adapting the node mobility, we describe a simple mobility model. Let each node assumes Euclidean coordinates and are placed in linear grid such that its incrementing the x-coordinate node moves to forward node whereas decrementing x-coordinate node moves to backward direction. When the mobile node is within the range of node with coordinates x, y, the mobile nodes can move to left node $(x-1, y)$ or right node $(x+1, y)$ linearly. When the node reaches the right boundary Rx it rebounds back $Rx > (x+1)?(x+1) : (x-1)$. Similarly, mobile node reaching left boundary Lx it rebounds back $Lx < (x-1)?(x-1) : (x+1)$. Such a node which observes the mobile node maintains the location. The location updates is

compared when the nodes maintain location with the location when maintained by CDS nodes and is shown in figure-6.8. We observe that number of updates is quite reduced when maintained by CDS nodes. In order to update the location of mobile node efficiently we, assume that CDS nodes only maintains location of mobile node.

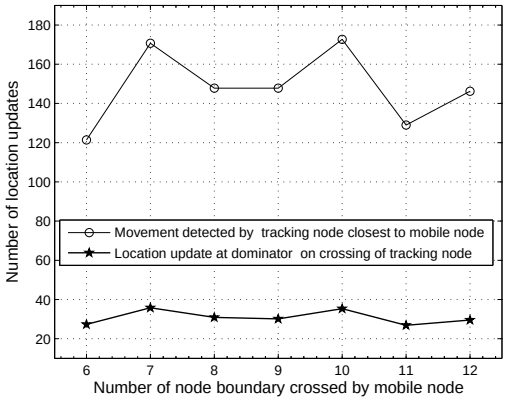


Figure 6.8: Performance comparison of location updates in nodes vs CDS nodes

The path from mobile node x to intermediate node a to dominator d is called tracking path. The location is updated only when the node moves out from one dominator region to another dominator region based on time stamp of mobile node. The tracking path is ascertained using highest weight 2-hop tracking path from x to some dominator d. We compare our weighted CDS approach with the shortest path based tracking approach in terms of location updates for mobile node. The comparison is shown in figure-6.9. We observe that weighted CDS brings in substantial reduction in location updates for tracking of mobile node.

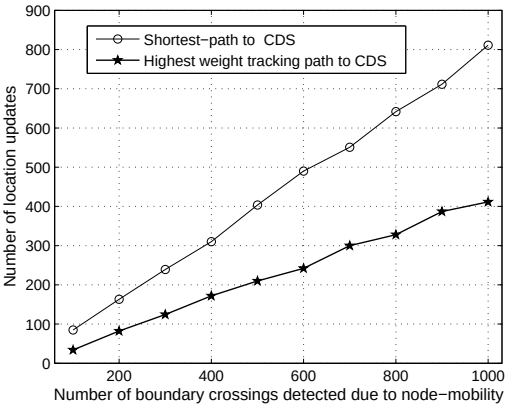


Figure 6.9: Performance comparison location updates in highest weight 2-hop tracking path using weighted CDS vs shortest path(hops)-tracking path method

6.9 Summary

We have reported an efficient node mobility transparent CDS construction technique for sensor networks. The scope of application of this algorithm ranges from energy recharging of nodes to strategic military applications which self configures a small CDS based backbone and using self reconfiguration and adaptation when a few node in CDS are mobile. This self organizing CDS construction algorithm integrates three main approaches: $i)$ self configuring CDS construction and self reconfiguration when node becomes mobile or becomes stationary, $ii)$ adaptation of mobile nodes using tracking technique by CDS backbone for its location updates and $iii)$ optimization using Markov model based weighted CDS to reduce self reconfiguration and tracking updates. The self configuring CDS algorithm has time complexity of $O(n\Delta^3)$, where n is number of nodes in network and Δ is the maximum node degree of G. The location updates of mobile node by weighted CDS achieves 40% reduction compared to shortest hop tracking path approach. The complexity of node tracking approach is $O(d \log d)$, where d is the number of boundary crossings.

Chapter 7

Conclusions

The work presented in this thesis comprises of domination techniques for *lifetime problems in sensor networks*. In the perspective of sensor networks, the following graph domination problems on unit disk graphs are addressed in the thesis: minimum connected dominating set, maximum domatic partition and maximum connected domatic partition. These problems are NP-complete, therefore we considered approximation solutions guided by assessment of the quality of solutions for the NP-complete domination problems. We have also addressed the problem of self organisation of the virtual backbone in the presence of node mobility. All the solutions that have been presented here are essentially distributed algorithms in view of the distributed nature of the target platform. Results and findings of this research work for the specific problems considered in this thesis are summarised in the next section.

7.1 Contributions

Efficient clusterhead rotation via domatic partition In this work, we considered the problem of constructing a domatic partition of nodes in a sensor network when the nodes are equipped to determine their location information. It helps in maximizing the lifetime of clusters induced by disjoint dominating sets, thereby prolonging the lifetime of the network. This first work of the thesis presents a self organization technique for the maximum domatic partition problem. Each node is equipped with geographical positioning system (GPS) or assumes some localization service in place which assigns location using a few nodes equipped with GPS. The domatic partition algorithm provides an energy efficient solution to the hierarchical topology control problem in sensor networks by means of energy efficient load balancing, thereby prolonging the network lifetime. Our simulation studies indicate that our technique is competitive in performance with other available schemes.

Rotation of CDS via Connected Domatic Partition In this work, we considered the problem of constructing a connected domatic partition for the network graphs with connectivity information only. The underlying aim again was to prolong the network lifetime in providing a CDS based backbone by enabling load balancing across all network nodes. This second work of the thesis is a heuristic driven distributed algorithm to find an approximate solution to the maximum connected domatic partition problem which is NP-Hard. We assume a general model for ad hoc networks with connectivity information only. By connectivity, we mean that nodes in sensor network neither have location information nor can they sense the distances. Therefore, nodes can reach their direct neighbors within their maximum transmission range through messages, but can reach non-neighboring nodes only through multi-hop communication. We have introduced a proximity heuristics for partitioning the network graph. A matching scheme then produces an approximate solution to the maximum connected domatic partition. The connected domatic partition of the network ensure not only CDS based general backbone but also prolongs network lifetime by load balancing.

CDS construction using a collaborative cover heuristic In this work, we considered the fundamental problem of constructing a connected dominating set from a single leader. This third work of the thesis is a distributed algorithm for finding an approximate solution to the minimum connected dominating set problem, which is also NP-complete. Our algorithm assumes the existence of a single leader which initiates the construction of connected dominating set. We have introduced a collaborative cover heuristic to find the local best cover which helps to optimize the CDS construction process. The approximation factor of our approach is at most $(4.8 + \ln 5)\mathrm{opt} + 1.2$, where opt is size of any optimal CDS. This approximation scheme is well suited for lossless aggregation backbone function in sensor networks. The simulation study reveals a substantial improvement in reducing the energy dissipation results in prolonging the life of node and the network.

Node mobility transparent CDS construction algorithm In this work, we considered the problem of constructing MPR based self organizing connected dominating set with the objective of handling node mobility efficiently and transparently. We have extended the basic MPR based dominating set constructing technique to support self organisation in the presence of node mobility. Our MPR based CDS construction algorithm has three main functions: $i)$ self configuring MPR based CDS, $ii)$ self reconfiguring CDS when a node in CDS becomes mobile or halts after completing mobility operation and $iii)$ adapting the CDS to keep the mobile node connected to the CDS backbone by tracking its location. Two tracking schemes have been developed. The first one uses the shortest-hop path to the mobile node from the nearest CDS dominator. The second is a Markov model based scheme to predict the movement of the mobile

node. This scheme has the advantage of making fewer location updates over the simpler shortest-hop path based tracking scheme, but has a higher computational overhead. Our tracking approach has the complexity $O(d \log d)$, where d is the number of boundary crossing by a mobile node.

7.2 Directions for further work

There are ample topics to explore within this research area. Domination algorithms and protocols developed for sensor networks could also be applied for other kinds of networks as well. Application of the techniques developed here for different types of networks, such a peer-to-peer networks can also be considered. Improvement of approximation bounds for various graph theoretic problems considred remains an important challenge. Development of a scheduling technique based on local information so that each node coordinates with a local coordinator to generate a local schedule would also be of interest.

Bibliography

[1] S. V. Pemmaraju and I. A. Pirwani, "Energy conservation via domatic partitions," in *MobiHoc '06: Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, (Florence, Italy), pp. 143–154, ACM, 2006.

[2] "Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004. Ossama Younis and Sonia Fahmy.

[3] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–70, 2002.

[4] S. Basagni, "Distributed clustering for ad hoc networks," in *Proceedings of the International Symposium on Parallel Architectures, Algorithms and Networks (I-SPAN '99)*, (Fremantle, Australia), IEEE Computer Society, 1999.

[5] W. Heinzelman, *Application-Specific Protocol Architectures for Wireless Networks*. PhD thesis, Massachusetts Institute of Technology, 2000.

[6] P. Wang, C. Li, and J. Zheng, "Distributed minimum-cost clustering protocol for underwater sensor networks (uwsns)," in *Proceedings of the IEEE International Conference on Communications (ICC07)*, (Glasgow, Scotland), pp. 3510–3515, IEEE, 2007.

[7] U. Feige, M. M. Halldórsson, G. Kortsarz, and A. Srinivasan, "Approximating the domatic number," *SIAM J. Comput.*, vol. 32, no. 1, pp. 172–195, 2003.

[8] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater, *Fundamentals of Domination in Graphs*. CRC Press, 1998.

[9] S. V. Pemmaraju and I. A. Pirwani, "Energy conservation via domatic partitions," *Proceedings of the seventh ACM international symposium on Mobile ad hoc networking and computing MobiHoc 2006*, pp. 143–154, 2006.

[10] T. Moscibroda and R. Wattenhofer, "Maximizing the lifetime of dominating sets," in *IPDPS '05: Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05) - Workshop 12*, (Denver, Colorado), p. 242.2, IEEE Computer Society, 2005.

[11] A. Ephremides, J. Wieselthier, and D. Baker, "A design concept for reliable mobile radio networks with frequency hopping signaling," *Proc. of the IEEE*, vol. 75, pp. 56–73, 1987.

[12] C. Adjih, P. Jacquet, and L. Viennot, "Computing connected dominated sets with multipoint relays," *Ad Hoc & Sensor Wireless Networks*, vol. 1, no. 1-2, p. 2739, 2005.

[13] P.-J. Wan, K. M. Alzoubi, and O. Frieder, "Distributed construction of connected dominating set in wireless ad hoc networks," *Mobile Networks and Applications*, vol. 9, no. 2, pp. 141–149, 2004.

[14] M. Cardei, M. Cheng, X. Cheng, and D. Du, "Connected dominating set in ad hoc wireless networks," in *Proc. Sixth International Conference Computer Science and Informatics(CS&I'2002)*, 2002.

[15] M. Min, H. Du, X. Jia, C. X. Huang, S. C.-H. Huang, and W. Wu, "Improving construction for connected dominating set with steiner tree in wireless sensor networks," *Journal of Global Optimization*, vol. 35, no. 1, pp. 111–119, 2006.

[16] Y. Li, M. T. Thai, F. Wang, C.-W. Yi, P.-J. Wan, and D.-Z. Du, "On greedy construction of connected dominating sets in wireless networks: Research articles," *Wireless Communication Mobile Computing*, vol. 5, no. 8, pp. 927–932, 2005.

[17] H. Kung and D. Vlah, "Efficient location tracking using sensor networks," in *IEEE Wireless Communications and Networking(WCNC 2003)*, pp. 1954–1961, IEEE Computer Society, 2003.

[18] C.-Y. Lin, W.-C. Peng, and Y.-C. Tseng, "Efficient in-network moving object tracking in wireless sensor networks," *IEEE Trans. on Mobile Computing*, vol. 5, no. 8, pp. 1044–1056, 2006.

[19] L.-H. Yen and C.-C. Yang, "Mobility profiling using markov chains for tree-based object tracking in wireless sensor networks," in *SUTC '06: Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing - Vol 2 - Workshops*, pp. 220–225, IEEE Computer Society, 2006.

[20] B. Awerbuch and D. Peleg, "Online tracking of mobile users," *JACM*, vol. 42, no. 5, pp. 1021–1058, 1995.

[21] J. Wu, W. Lou, and F. Dai, "Extended multipoint relays to determine connected dominating sets in manets," *IEEE Trans. on Computers*, vol. 55, no. 3, pp. 334–347, 2006.

[22] E. J. Cockayne and S. T. Hedetniemi, "Towards a theory of domination in graphs," *Networks*, vol. 7, no. 3, pp. 247–261, 1977.

[23] D. Estrin, R. Govindan, J. S. Heidemann, and S. Kumar, "Next century challenges: Scalable coordination in sensor networks," in *Mobile Computing and Networking*, pp. 263–270, 1999.

[24] B. N. Clark, C. J. Colbourn, and D. S. Johnson, "Unit disk graphs," *Discrete Mathematics*, vol. 86, pp. 165–177, January 1990.

[25] F. Kuhn, T. Moscibroda, and R. Wattenhofer, "On the locality of bounded growth," in *PODC '05: Proceedings of the twenty-fourth annual ACM symposium on Principles of distributed computing*, (Las Vegas, NV, USA), pp. 60–68, ACM, 2005.

[26] N. Bulusu, D. Estrin, L. Girod, and J. Heidemann, "Scalable coordination for wireless sensor networks: Self-conguring localization systems," in *International Symposium on Communication Theory and Applications (ISCTA)*, (Lake Ambleside, UK), July 15-20 2001.

[27] D. Rakhmatov and S. Vrudhula, "Energy management for battery-powered embedded systems," *IEEE Trans. on Embedded Computing Sys.*, vol. 2, no. 3, pp. 277–324, 2003.

[28] F. Kuhn, T. Moscibroda, T. Nieberg, and R. Wattenhofer, "Fast deterministic distributed maximal independent set computation on growth-bounded graphs," in *Proceedings of 19th International Conference on Distributed Computing (DISC-2005)* (P. Fraigniaud, ed.), vol. 3724 of *Lecture Notes in Computer Science*, pp. 273–287, Springer, 2005.

[29] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *HICSS '00: Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8*, (Washington, DC, USA), p. 8020, IEEE Computer Society, 2000.

[30] F. Dressler, "Self-Organization in Ad Hoc Networks: Overview and Classification," Technical Report 02/06, University of Erlangen, Dept. of Computer Science 7, March 2006.

[31] W. Wu, H. Du, X. Jia, Y. Li, and S. C. H. Huang, "Minimum connected dominating sets and maximal independent sets in unit disk graphs," *Theoretical Computer Science*, vol. 352, no. 1, pp. 1–7, 2006.

[32] B. Zelinka, "Connected domatic number of a graph," *Math. Slovaca*, vol. 36, pp. 387–392, 1986.

[33] B. L. Hartnell and D. F. Rall, "Connected domatic number in planar graph," *Czechoslovak Math.Journal*, vol. 126, no. 51, pp. 173–179, 2001.

[34] D. M. Blough and P. Santi, "Investigating upper bounds on network lifetime extension for cell-based energy conservation techniques in stationary ad hoc networks,"

in *MobiCom '02: Proceedings of the 8th annual international conference on Mobile computing and networking*, (Atlanta, Georgia, USA), pp. 183–192, ACM, 2002.

[35] M. Cardei and D.-Z. Du, "Improving wireless sensor network lifetime through power aware organization," *Wirel. Netw.*, vol. 11, no. 3, pp. 333–340, 2005.

[36] O. Younis, M. Krunz, and S. Ramasubramanian, "Node clustering in wireless sensor networks: recent developments and deployment challenges," *IEEE Network*, vol. 20, no. 3, pp. 20–25, 2006.

[37] S. Bandyopadhyay and E. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks," in *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom 2003)*, 2003.

[38] S. Soro and W. B. Heinzelman, "Prolonging the lifetime of wireless sensor networks via prolonging the lifetime of wireless sensor networks via," in *IPDPS '05: Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05) - Workshop 12*, (Washington, DC, USA), p. 236.2, IEEE Computer Society, 2005.

[39] S. Ghiasi, A. Srivastava, X. Yang, , and M. Sarrafzadeh, "Optimal energy aware clustering in sensor networks," *Sensors*, vol. 2, pp. 258–269, 2002.

[40] M. Chatterjee, S. K. Das, and D. Turgut, "Wca: A weighted clustering algorithm for mobile ad hoc networks," *Cluster Computing*, vol. 5, pp. 193–204, April 2002.

[41] J.Panch and P.K.Agrawal, *Combinatorial Geometry*. John Wiley, 1995.

[42] G. Simon, P. Volgyesi, M. Maroti, and A. Ledeczi, "Simulation-based optimization of communication protocols for large-scale wireless sensor neworks," in *In Proceedings of 2005 IEEE Aerospace Conference*, (Big Sky, MT), March 2003.

[43] S. Schmid and R. Wattenhofer, "Algorithmic models for sensor networks," in *20th International Parallel and Distributed Processing Symposium (IPDPS 2006)*, (Rhodes Island, Greece), April 2006.

[44] M. R. Garey and D. S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co., 1990.

[45] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, pp. 102–114, August 2002.

[46] B. Das, R. Sivakumar, and V. Bharghavan, "Routing in ad hoc networks using a spine," in *IC3N '97: Proceedings of the 6th International Conference on Computer Communications and Networks*, (Washington, DC, USA), p. 34, IEEE Computer Society, 1997.

[47] V. Bharghavan and B. Das, "Some routing in ad hoc networks using minimum connected dominating sets," *Proc. of International Conference on Communications'97*, pp. 376–380, January 1997.

[48] D. Rakhmatov, S. Vrudhula, and D. A. Wallach, "Battery lifetime prediction for energy-aware computing," in *ISLPED '02: Proceedings of the 2002 international symposium on Low power electronics and design*, (Monterey, California, USA), ACM, 2002.

[49] M. Cardei, D. Maccallum, M. X. Cheng, M. Min, X. Jia, D. Li, and D. Du., "Wireless sensor networks with energy efficient organization," *Journal of Interconnection Networks*, vol. 3, pp. 213–229, 2002.

[50] J. Blum, M. Ding, A. Thaeler, and X. Cheng, *Connected Dominating Set in Sensor Networks and MANETs*, pp. 329–369. Kluwer Academic, 2004.

[51] M. V. Marathe, H. Breu, H. B. Hunt III, S. S. Ravi, and D. J. Rosenkrantz, "Simple heuristics for unit disk graphs," *Networks*, vol. 25, pp. 59–68, 1995.

[52] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communications*, vol. 30, no. 14-15, pp. 2826–2841, 2007.

[53] N. A. Lynch, *Distributed Algorithms*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1996.

[54] C. Bettstetter, "On the minimum node degree and connectivity of a wireless multihop network," *Proceedings of the seventh ACM international symposium on Mobile ad hoc networking and computing MobiHoc 2002*, pp. 80–91, 2002.

[55] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *Mobile Computing and Networking*, pp. 56–67, 2000.

[56] S. Lindsey, C. Raghavendra, and K. M. Sivalingam, "Data gathering algorithms in sensor networks using energy metrics," *IEEE Trans. on Parallel and Distributed Systems*, vol. 13, pp. 924– 935, September 2002.

[57] Y. Xu, J. S. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *MOBICOM*, pp. 70–84, 2001.

[58] H. Gupta, V. Navda, S. R. Das, and V. Chowdhary, "Efficient gathering of correlated data in sensor networks," in *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, (Urbana-Champaign, IL, USA), pp. 402–413, ACM, 2005.

[59] K. M. Alzoubi, P.-J. Wan, and O. Frieder, "Message-optimal connected dominating sets in mobile ad hoc networks," in *MobiHoc '02: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, (Lausanne, Switzerland), pp. 157–164, ACM, 2002.

[60] J. Wu and H. Li, "On calculating connected dominating set for efficient routing in ad hoc wireless networks," in *DIALM '99: Proceedings of the 3rd international workshop on Discrete algorithms and methods for mobile computing and communications*, (Seattle, Washington, United States), pp. 7–14, ACM, 1999.

[61] Y. Li, M. T. Thai, F. Wang, C. W. Yi, P. J. Wang, and D. Z. Du, "On greedy construction of connected dominating sets in wireless networks," *Special issue of Wireless Communications and Mobile Computing (WCMC)*, vol. 5, pp. 927–932, January 2005.

[62] O. Ore, *Theory of Graphs*, vol. 38. American Math. Soc. Colloq. Publ., 1962.

[63] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint relaying for flooding broadcast message in mobile wireless networks," in *Proceedings of the 35th Hawaii International Conference on System Sciences (HICSS-35)*, pp. 3898–3907, January 2002.

[64] S. Guha and S. Khuller, "Approximation algorithms for connected dominating sets," *Algorithmica*, vol. 20, no. 4, pp. 374–387, 1998.

[65] V. Kulathumani, A. Arora, M. Demirbas, and M. Sridharan, "Trail: A distance sensitive wsn service for distributed object tracking," *Lecture Notes in Computer Science*, vol. 4373, pp. 83–100, 2007.

[66] S. Kamei and H. Kakugawa, "A self-stabilizing distributed approximation algorithm for the minimum connected dominating set," in *Proceedings of 21th International Parallel and Distributed Processing Symposium (IPDPS 2007)*, (Long Beach, California, USA), pp. 1–8, IEEE Press, March 2007.

Publications from this work

Published articles

1. Rajiv Misra and Chittaranjan Mandal. An Improved Energy Efficient Distributed Clustering Algorithm for Large Wireless Sensor Networks. In *Proc. of 4th Asian International Mobile Computing Conference (AMOC 2006)*, Kolkata, January, 2006. Tata-McGraw-Hill Press.

2. Rajiv Misra and Chittaranjan Mandal. Self-Rotating Clusterheads for Self-Organizing Cluster Sensor Networks. In *Proc. of ADCOM 2006*, Surathkal, December 2006. IEEE Press.

3. Rajiv Misra and Chittaranjan Mandal, Self-Healing for Self-Organizing Cluster Sensor Networks. In *IEEE INDICON 2006, New Delhi.*

4. Rajiv Misra and Chittaranjan Mandal. A Localized Self-configuration Scheme for Controlled Mobility in Self-Organizing Sensor Network. In *Proc. of 5th Asian International Mobile Computing Conference (AMOC 2007)*, Kolkata, January, 2007. Tata-McGraw-Hill Press.

5. Rajiv Misra and Chittaranjan Mandal. Clusterhead Rotation via Domatic Partition in Self-Organizing Sensor Networks. In *Proc. of IEEE/ACM COMSWARE 2007*, Bangalore, Janurary, 2007 (IEEE Press).

6. Rajiv Misra and Chittaranjan Mandal. Self-Stabilizing MPR based Connected Dominating Set for Ad hoc Networks. (Accepted for publication in *Proc of International Wireless Communications and Mobile Computing Conference 2008 (IWCMC 2008)*, August 2008, Crete, Greece (IEEE Press).

Articles under revision

1. Rajiv Misra and Chittaranjan Mandal. Efficient Clusterhead Rotation via Domatic Partition in Self-Organizing Sensor Networks using GPS . *Wireless Communication and Mobile Computing*. *Wiley* (Minor Revision Submitted)

2. Rajiv Misra and Chittaranjan Mandal. Rotation of CDS via Connected Domatic Partition in Ad hoc Sensor Networks. *IEEE Transactions on Mobile Computing*. (Revision submitted)

Communicated articles

1. Rajiv Misra and Chittaranjan Mandal. Minimum Connected Dominating Set using Collaborative cover heuristic and Steiner-tree for Ad hoc Sensor Networks. *under review*.

2. Rajiv Misra and Chittaranjan Mandal. Efficient Node Mobility Transparent Connected Dominating Set in Sensor Networks. *under review*.

Biodata of Author

Mr Rajiv Misra, son of Shri Raj Kumar Misra, was born on 18th July, 1966. He graduated with a **B. E.** in *Computer Science* in 1987 from MN National Institute of Technology, Allahabad. He received his **M. Tech.** in *Computer Science* in 1989, from Indian Institute of Technology, Bombay. His area of research is *Mobile Computing*. His is working for a PhD in this area from the School of Information Technology, Indian Institute of Technology, Kharagpur.

His permanent address is 104A/325 Rambagh, Kanpur, Uttar Pradesh - 208012, India. He is an Indian national.