Distributed Connected Dominating Set Techniques for Energy-Efficient Topology Control in Wireless Sensor Networks

A thesis submitted for the degree of
Doctor of Philosophy

by

Azrina Abd Aziz

Department of Electrical and Computer Systems Engineering
Monash University
Australia

June 2012

Copyright Notices

Abstract

Despite the considerable research efforts devoted to extending the lifetime of wireless sensor networks (WSNs) by making them more energy efficient, there are still a number of unresolved issues. Among the possible solutions for improving their overall energy efficiency, topology control has significant potential. Distributed topology control is a difficult problem, and optimal solutions are not possible except for very simple topologies. Because of this, heuristic methods are used, but the solutions proposed in the research literature are usually tested with overly simplistic simulation models and consequently they fail to perform satisfactorily in real networks.

The research project reported in this thesis proposes three new topology control methods that are tested on highly realistic simulation models calibrated with data collected on an experimental wireless sensor network. These models accurately handle interference effects, realistic transmission ranges and imperfect communication links. Additionally, the correctness of the proposed methods was verified using theoretical analysis. Two leading algorithms were used as benchmarks.

Based on the outcomes of a thorough literature review and analysis of existing techniques, distributed connected dominating set (CDS) approach was selected as the starting point for the design of the proposed algorithms. The proposed algorithms are not only distributed but also use localized information for computing a CDS. Given that the CDS serves various tasks in a WSN, a fair load distribution strategy was adopted to prolong the network lifetime. This strategy takes into account the remaining energy levels at each node when choosing the eligible CDS nodes.

The first topology control technique called the *three-phase single initiator (TPSI)* was developed to form a small CDS for medium and dense networks (i.e., in deployments when average node degree is relatively high) with minimal communication overhead, computational complexity and energy consumption. The simulation results demonstrate that the TPSI algorithm generates a small CDS for both medium and dense networks but not for sparse networks. These results also prove that the impact of network density on performance of an algorithm is significant and cannot be ignored.

The second technique, *single-phase single initiator (SPSI)* on the other hand was proposed for applications that require fast convergence, and is best suited to WSN applications that have sparse topologies. The simulation results show that SPSI can generate a small CDS for sparse networks using low message overhead and energy consumption, and compute a CDS faster than the TPSI algorithm.

The third one, the *Two-phase multiple initiator (TPMI)* algorithm adapts well to dynamic topology changes, thus it is suitable for applications that require frequent topology updates. Instead of relying on a single initiator to construct the CDS as in the TPSI and SPSI, the TPMI algorithm uses multiple initiators. The simulation results show that although the CDS size of TPMI is larger than the ones generated by TPSI or SPSI, it outperforms them in terms of energy consumption, network lifetime and convergence time in networks with rapidly changing topologies.

Best suited algorithm for a particular installation can be selected manually, or by using some measurement techniques, the structure of a network can be probed to activate the optimal method automatically.

Acknowledgments

This thesis would not have been possible without the help of the following people. First and foremost, I owe my deepest gratitude to my supervisors, Dr Ahmet Şekercioğlu and Dr Milosh Ivanovich for their supervision and guidance throughout my study at the Monash University. Especially Dr Ahmet, he has greatly helped me in so many aspects mainly on setting up my research direction and providing invaluable advice, feedback and technical support for my research.

I would also like to record my utmost appreciation to my husband, Sallehudin Irwan for his endless support, great understanding and persistent encouragement. Without his patience and sacrifices, I could not have gotten this far. I am really grateful to his willingness to take care of our children during my long working hours at the office. To my three wonderful children, Mardhiah, Yusuf and Soffiya your love, smiles and sacrifices have kept me moving forward throughout the duration of my study. Not to forget, my loving parents for their continuous support and care.

I am also indebted to my sponsor, Universiti Teknologi Petronas for giving me the opportunity to pursue my study and providing the financial support for the duration of my study at the Monash University.

A special thank goes to Helen Rahman, Olga Trivailo, Emma Bunton and Martin Firus for their help in proofreading my thesis and papers. I am also indebted to Shakil Bhuiyan and Sarange Dissanayake for their guidance in drawing the 3D graphs in Matlab. Lastly, I offer my regards and blessings to all of those who supported me in any respect during the completion of the project.

Declaration

I declare that, to the best of my knowledge, the research described herein is original except where the work of others is indicated and acknowledged, and that the thesis has not, in whole or in part, been submitted for any other degree at this or any other university.

Azrina Abd Aziz
Melbourne
June 2012

Publications Related to the Study Presented in This Thesis

Journal Article

A. A. Aziz, Y. A. Şekercioğlu, P. Fitzpatrick and M. Ivanovich, "A Survey on Distributed Topology Control Techniques for Extending the Lifetime of Battery Powered Wireless Sensor Networks", *IEEE Communications Surveys and Tutorials*, 2012, doi: 10.1109/SURV.2012.031612.00124.

Conference Papers

A. A. Aziz and Y. A. Şekercioğlu, "A Distributed Topology Control Method for Improving Energy Efficiency of Wireless Sensor Networks", in *Proceedings of the 4th International Conference on Intelligent and Advanced Systems (ICIAS 2012)*, Kuala Lumpur, Malaysia, June 2012.

A. A. Aziz and Y. A. Şekercioğlu, "A Distributed Energy Aware Connected Dominating Set Technique for Wireless Sensor Networks", in *Proceedings of the 4th International Conference on Intelligent and Advanced Systems (ICIAS 2012)*, Kuala Lumpur, Malaysia, June 2012.

Contents

List of Figures

List of Tables

List of Acronyms

ASCENT	Adaptive Self-configuring Sensor Network Topologies
CDS	Connected Dominating Set
CLUSTERPOW	Cluster Power
COMPOW	Common Power
CSIRO	Commonwealth Scientific & Industrial Research Organization
DARPA	Defense Advanced Research Projects Agency
DS	Dominating Set
ECDS	Energy-efficient Connected Dominating Set
GAF	Geographical Adaptive Fidelity
GPS	Global Positioning System
ID	Node Identifier
IoT	The Internet of Things
IS	Independent Set
IST	Information Society Technologies
LEACH	Low-energy Adaptive Clustering Hierarchy
MAC	Media Access Control
MCDS	Minimum Connected Dominating Set
MECN	Minimum Energy Communication Network
MIS	Maximal Independent Set
M2M	Machine-to-Machine Communications
ND	Node Degree
PACDS	Power Aware Connected Dominating Set
SMECN	Small Minimum Energy Communication Network
SPAN	Energy-efficient Coordination
SPSI	Single-phase Single Initiator
STEM	Sparse Topology & Energy Management
TC	Topology Control
TMPO	Topology Management by Priority Ordering
TPMI	Two-phase Multiple Initiator
TPSI	Three-phase Single Initiator
UDG	Unit Disk Graph
WSN	Wireless Sensor Network

Chapter 1

Introduction

Wireless sensor networks (WSNs) have been highlighted as one of the most important technologies for the 21st century [bus99], and the Massachusetts Institute of Technology's Technology Review Magazine [eme03] identifies them as one of the top ten emerging technologies that will change the world. Their promise has led to unprecedented growth in research and commercial activities. Among the research funding bodies worldwide, the Defense Advanced Research Projects Agency (DARPA) of the United States, the Commonwealth Scientific and Industrial Research Organization (CSIRO) of the Australia and the Information Society Technologies (IST) Research Framework of the European Union have invested a significant amount of money into WSN research. Based on the forecast conducted by the Harbor Research Consulting [Con], the total revenue of intelligent devices used in WSNs, and ZigBee [All] and other IEEE 802.15.4 [802] related technologies is estimated to exceed US$ 12 billion by 2013.

A WSN is composed of many small, cheap and potentially smart devices equipped with sensing, data processing and storage and communication capabilities. These features allow for the deployment of the network in various applications ranging from military, civil, industrial to health for the purpose of gathering data. However, the miniaturization of the sensor nodes introduces the following challenge. The sensor nodes generally operate on battery power. Therefore, as their size is reduced, the battery size and capacity become smaller. As a result, they have severely limited power sources. In most applications, it is difficult or impossible to retrieve the nodes in order to replace or recharge the batteries. For this reason, energy conservation becomes a major focus in the design of WSNs. Currently, many energy conservation techniques have been proposed to optimize the energy consumption and to extend the network lifetime [MGG10, WLBW01, WWS03, YJY06]. Topology control is one of the leading methods used for minimizing the total energy consumption in deployment. It is a multidimensional problem, allowing control over various network parameters to optimize network performance. One efficient method to accomplish topology control is to form a structured network topology through the use of a connected dominating set (CDS). This thesis focuses on this and aims to design very efficient topology control techniques with minimal computational and communication overheads.

The remaining sections of this chapter are structured as follows: Section 1.1 covers the background information related to WSNs, in particular their features, applications and challenges. In Section 1.2, the concept of topology control is explained and its role as a power management strategy is discussed. Section 1.2.1 gives an overview of the CDS, explains the reasons for choosing the CDS method for the proposed algorithms and discusses the motivations of the thesis. Section 1.4 summarizes the contributions of the research and finally, the structure of the thesis is presented in Section 1.5.

1.1 Wireless Sensor Networks

The main components of a wireless sensor node are radio transceiver, processor, memory unit, and power source [ASSC02b]. The advancement in low power design, wireless technology and miniaturization of the sensors allow these components to be integrated into a single integrated circuit package. This has resulted in a drop of the price and size of the nodes. Figure 1.1 illustrates the estimated price of the future sensor nodes which could reach about 8 cents in 2019. But, miniaturization severely restricts the computation, communication and power supply capabilities which inevitably impose limits on algorithm designs.

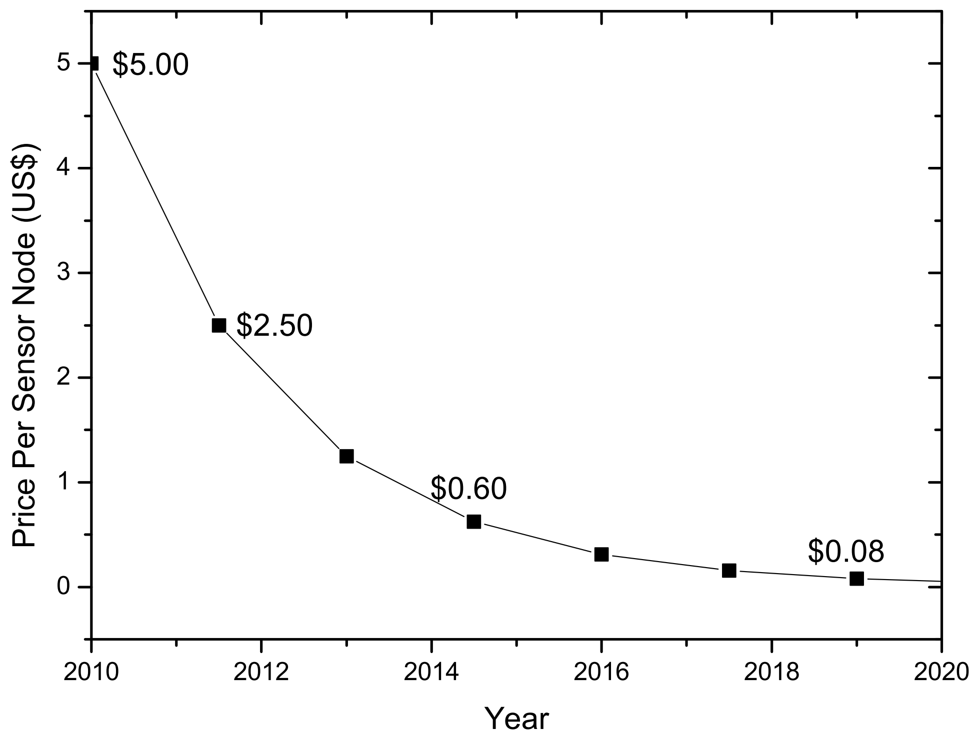


Figure 1.1: Price of sensor nodes is halving every 18 months and could drop to 8 cents in 2019 [Ste05].

Despite the limited capacity of sensor nodes, WSNs have been widely deployed in various applications because:

1. They provide flexible access to information from any physical location without human intervention.

2. They are self-organized and self-configured networks.

3. They form networks on the fly, without requiring any fixed infrastructure, thus resulting in minimum set-up and maintenance costs. For example, it has been reported in [200S8] that the cost of wiring in an industrial installation is estimated to be US$ 130-650 per meter. By using the wireless technology this cost can be reduced up to 20-80%.

1.1.1 Current and Future Applications of WSNs

There are numerous applications of WSNs developed so far. They include military surveillance [HKS+04], industrial quality control [WST], environmental monitoring [LCD03], personal health monitoring [LMFJ+04] and traffic monitoring [ECV+05].

With the recent technological development of large scale deployments of sensors such as Machine-to-Machine Communications (M2M) [FTS11, GTJ+11, NLP11, YRS+11], Smart Grid [BF54, MM09, SFEB09, ZVS11], The Internet of Things (IoT) [AIM10, MF10, Mul10], and Smart Environment applications [CD05, Pos09], many more exciting applications of WSNs are emerging. These new applications are envisioned to be embedded into everyday objects and eventually to influence all aspects of our lives. A sophisticated home automation network called smart home [DHSR08, EG01] is one example of the future applications of WSNs that will become increasingly ubiquitous in everyday life. WSNs are also envisioned to be integrated into more complex networks such as Cellular networks or the Global Internet to provide information sharing with a wide variety of devices as illustrated in Figure 1.2.

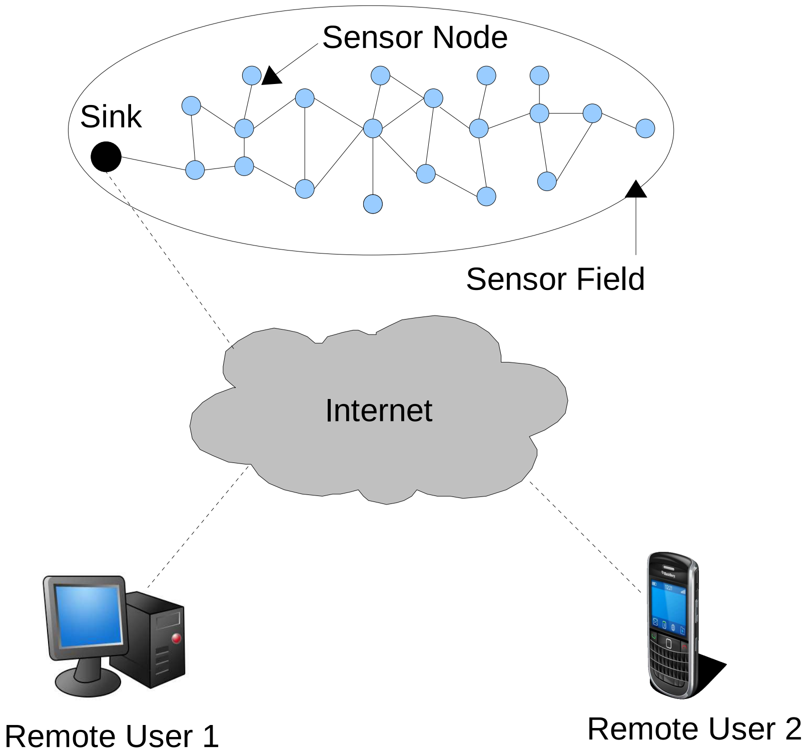


Figure 1.2: A WSN allows information sharing with remote users via Internet.

1.1.2 Challenges in WSNs

WSNs pose significant research challenges due to their unique features mentioned in Section 1.1. They are different from cellular, wired or ad hoc networks in several ways. Table 1.1 provides a comparative overview of them.

The biggest challenge of WSNs is managing the limited energy resources in the network in order to increase the network lifetime [ASSC02a]. These nodes are powered by a battery. Their energy sources are limited and often during the deployment of the networks, they cannot be replaced or recharged. As many applications are demanding a smaller size of sensor nodes, the size of the battery has to be reduced. This however reduces the capacity of the battery. With the slow advancement in battery technology, power conservation will remain an important issue in WSNs.

Cellular/Wired Networks	**WSNs**	**Ad Hoc Networks**
Infrastructured networks	Infrastructureless networks	Infrastructureless networks
Static network topology	Highly dynamic network topology	Less dynamic network topology
Unlimited resources	Limited resources	Unlimited resources
Fixed central coordinators such as cell sites	Lack central coordinators	Lack central coordinators
Pre-planned installation	Ad hoc and automated installation	Ad hoc and automated installation
High installation costs	Low installation costs	Low installation costs
Low failure rates	High failure rates	Low failure rates
Network size is large	Network size is large	Network size is small
One-to-one communication	Many-to-one communications	One-to-one communication

Table 1.1: A comparison of WSN features with cellular, fixed and ad hoc networks.

Due to the unique features of WSNs, many energy management techniques currently proposed for fixed, ad hoc or cellular networks cannot be directly used for WSN applications. Thus, it is important to devise suitable techniques for WSNs that can reduce energy consumption and extend the lifetime of these networks. This thesis will mainly focus on the topology control subject, which is the widely-studied energy conservation technique used in WSNs.

1.2 Topology Control as an Energy Management Solution in WSNs

A network topology defines the placement of nodes and the connectivity among nodes in the network. It is determined by the nodes' physical locations and transmission power. In WSNs, it is highly likely that large number of nodes are deployed in a geographical area of interest. Because of this, usually the topology formed is dense and has a dynamic nature which impacts on the power consumption of the nodes and network performance. For example, a dense topology can cause excessive interference resulting in high packet drop and rapid energy depletion which in turn affects the longevity of the network. Since WSNs are heavily power constrained systems as discussed earlier, topology control is mainly used to eliminate redundant links in a dense topology. While preserving the network connectivity, it constructs a more efficient topology consisting of a smaller number of links to minimize the energy consumption.

There are four general approaches to exercise topology control in WSNs. They are *power adjustment* approaches which control the transmission power of node, *power mode* approaches which manipulate the operating mode of nodes, *clustering* approaches which form clusters in the network and *hybrid* approaches which combine either power mode, clustering or power adjustment in various ways. Among these four techniques, the clustering method has been widely used due to its ability to organize the networks into a hierarchical structure, a property which cannot be achieved by power adjustment or power mode approaches. It simplifies the network management, minimizes the network maintenance and reduces the communication overhead. The most prominent clustering approach is the CDS. A detailed discussion of the approaches is presented in Chapter 2.

The concept of the CDS is to form a subset of sensor nodes which acts as a backbone in a network. Nodes that are not in the subset must have at least one neighbor in the subset. CDS techniques have been used to support routing applications [WGS01, Wu02] or to eliminate the broadcast storm problem [TNCS02, SSZ02, WD03].

1.2.1 Why Distributed CDS Approaches?

In general, CDS algorithms can be classified according to their distributed or centralized approaches. Centralized approaches are efficient in generating a small CDS size [Bou09]. However, these algorithms rely on centralized nodes to collect information from all nodes in the network in order to generate a CDS. Due to the dynamic nature of WSNs and a large number of sensor nodes deployed in practice, the communication and computational costs involved in acquiring this global information are significantly high [BDTC04, Bou09, Sto05]. Therefore, centralized approaches are not suitable for practical implementation.

In contrast, distributed algorithms eliminate the need to learn global network topology. Nodes communicate with neighbors via exchanged messages to gather information and based on this information, they make decisions to compute the CDS. This results in low communication and computational overheads for generating and maintaining the CDS especially in the event of dynamic topology changes. Because of this reason, distributed CDS approaches are preferable over the centralized CDS approaches. Thus, this thesis focuses on distributed CDS approaches.

1.3 Thesis Motivation, Aims and Scope

Earlier CDS techniques [BCDP03, DB97, WL99, WLD06] are mostly designed to reduce the CDS size in order to minimize the communication overhead involved in constructing and maintaining the backbone. However, these techniques fail to consider the energy capacity of each node when forming the CDS. It is a well-known fact that the nodes in the backbone are heavily loaded with various tasks [WWS03]. If the backbone role is not fairly rotated among nodes, their energy will soon be depleted and this will require a CDS reconstruction. Unfortunately, a frequent CDS reconstruction consumes a significant amount of energy and shortens the lifetime of the network. Therefore, it is critical to minimize the energy consumption.

The common practice for conserving energy in CDS is to assign the role of the CDS to nodes with high remaining energy. Thus, the works in [BGLA03, MGG10, WGS01, YJY06] use the residual energy when selecting CDS nodes. However, these techniques have several shortcomings. First, although the energy capacity of nodes is taken into account, these techniques incur a high cost to construct the CDS. The cost here refers to the number of exchanged messages and the computational time of the CDS. The high cost implies a high energy consumption which can shorten the network longevity. Second, these techniques operate based on unrealistic assumptions such as that nodes can always communicate with neighbors without any interruption, nodes in the network are homogeneous devices having a fixed transmission range and/(or) nodes are evenly distributed in the network. Third, many CDS techniques fail to account for the impact of network density (average number of neighbors at each node) on their performance. In general, many of them perform well in dense networks but perform poorly in sparse networks. Hence, various network conditions should be considered when evaluating the performance of this type of algorithm.

Thus, efficient CDS techniques that can quickly compute the backbone, minimize the CDS size, reduce the energy consumption, maintain the CDS longer and use minimal computation and communication overheads are needed. This thesis proposes three new CDS techniques that fulfill these criteria while eliminating the unrealistic assumptions. A thorough performance evaluation is conducted for the new algorithms in the aspects of network density and network size.

Algorithm	Outstanding Features
TPSI	Generates small size CDS in medium and dense networks
SPSI	Generates small size CDS in sparse networks
TPMI	Leads to minimum energy consumption, also communication and computational requirements are low

Table 1.2: A brief overview of the proposed algorithms.

1.4 Thesis Contributions

The contributions presented in this thesis are as follows:

1. **A detailed and rigorous review of topology control techniques** was conducted to understand the state-of-the-art. The techniques were classified and their strengths and weaknesses were presented. This chapter was extended and published as a journal paper in [ASFI12].

2. **Three novel distributed CDS construction algorithms** *Three-phase Single Initiator (TPSI)*, *Single-phase Single Initiator (SPSI)* and *Two-phase Multiple Initiator (TPMI)* were designed to support various WSN applications. TPSI is efficient for minimizing the CDS size of medium and dense networks, whereas SPSI is efficient for reducing the CDS size of sparse networks. As small-size CDS is demanded in routing and broadcasting applications, both TPSI and SPSI can be used interchangeably to suit the network types during the applications. TPSI and SPSI algorithms were respectively published as conference papers in [AS12b] and [AS12a]. On the other hand, TPMI converges fast and is an energy-efficient algorithm that requires low message overhead to build a CDS. It is suitable for applications that trade-off the CDS size with energy-efficiency. Table 1.2 summarizes the outstanding features of each proposed algorithm.

3. **TPSI algorithm** generates a small CDS with low computational and communication overheads. It computes the CDS in a distributed approach using localized information gathered within two hops vicinity so that the cost of acquiring this information is minimized. To achieve a small CDS size, it first generates a dominating set, then finds connectors to connect the set and finally prunes the redundant nodes in the set. Based on the simulation results, it produces a reduced topology of small size for medium and dense networks and provides minimal computational and message complexities. This contributes to its low energy consumption and extended network operational time. This algorithm thus offers an energy-efficient solution for computing a CDS in most WSN applications.

4. **SPSI algorithm** forms a small CDS for sparse networks. SPSI uses only a single phase to construct a CDS. To retain the advantages of TPSI, SPSI also uses the localized information and communication with neighbors using exchanged messages. The performance evaluations of SPSI show its superior performance in creating a small backbone for sparse networks while keeping the communication overheads low.

5. **TPMI algorithm** has fast convergence time and it can cope with the dynamic nature of networks. The convergence time here refers to the computational time of finding the CDS. In TPMI, multiple initiators distributed across the network are selected to initiate the CDS construction. The major advantage of this algorithm is that it quickly finds the CDS and uses the lowest message overhead and energy consumption to build a CDS compared to the TPSI and SPSI but at the expense of generating a larger CDS size than the one formed in TPSI and SPSI.

1.5 Thesis Outline

The structure of the thesis is as follows:

- Chapter 2 presents the survey on topology control algorithms in WSNs. It provides the reader with the necessary background on the topology control aspect. It starts with the definition and classification of topology control, then discusses the state-of-the-art of current topology control techniques and finally critically analyzes the advantages and disadvantages of each topology control technique.

- Chapter 3 introduces the two single initiator algorithms, TPSI and SPSI, which build a small size backbone using a CDS approach.

- Chapter 4 describes the multiple initiator algorithm, TPMI. This algorithm not only adapts well to the dynamic topology changes but also minimizes the cost of the CDS construction overhead.

- Chapter 5 provides the theoretical analysis of the three algorithms. A series of proofs and the performance analysis of the algorithms are discussed.

- Chapter 6 presents and discusses the results of the performance evaluation of the algorithms conducted in a simulation-based experiment. In order to see their improvements over the existing methods, their performance is compared against the leading CDS algorithms.

- Chapter 7 covers the conclusions of the thesis and includes suggestions for future work.

Chapter 2

Topology Control Algorithms for Energy Conservation: A Survey

2.1 Introduction

This chapter presents a comprehensive review of topology control techniques in WSNs. As wireless sensor nodes are generally battery-powered devices, conserving energy for the purpose of extending the network lifetime is a critical requirement. This requirement motivates the development of various energy management techniques, including topology control [WPMR07, YHE01, HCB00, THH02, HDB04, RM99, JSAC01, WX06]. The aims of this chapter are twofold:

1. to introduce different classes of topology control techniques that have been proposed for WSNs, and
2. to see why topology control is poised to play an important part in energy management.

Special attention has been devoted to promising energy-efficient techniques that have gained wide attention in the literature of WSNs. These techniques are systematically classified, and detailed discussion including qualitative comparisons is presented and elaborated upon.

This chapter is organized as follows. Section 2.2 provides an overview of topology control algorithms to discuss the notion of topology control and motivations for using topology control algorithms in WSNs. The terminology of topology control, the guidelines for designing energy-efficient algorithms and their classification are covered in this section. Since the concept of "network lifetime" is widely used for assessing the performance of these algorithms, various definitions of the term are highlighted in Section 2.2.3, and their implication on WSNs is discussed in Section 2.4.2. Section 2.3 presents topology control techniques and introduces their state-of-the-art in energy saving. A special emphasis is given to the subject of distributed approaches because of their low communication overhead and feasibility for not relying on server nodes to construct the topology [Bou09]. Section 2.4 presents the performance analysis of the algorithms. The cost comparison of these algorithms as well as their advantages and disadvantages are given. Finally, Section 2.5 concludes this chapter with insights on research directions for further investigation.

2.2 Topology Control in WSNs

Topology control is a technique used in WSNs to achieve energy saving and extend network lifetime. The idea of topology control is to give sensor nodes a control over certain parameters so that these parameters can be manipulated in a way that will benefit the network. In particular, sensor nodes have the capacity to adjust the transmission power of their radio, switch to various modes of operation or even decide on the nodes eligible to participate in the network topology to create hierarchical structure called clustering. These features are the parameters that are exploited to dynamically modify the network topology so that the overall network performance is improved, particularly from an energy efficiency perspective. It is common practice for topology control to prune redundant links, thus resulting in a sparse topology. However, it should be noted that network characteristics such as connectivity and coverage must be preserved after the networks are subjected to topology control.

The main motivation for adopting topology control methods in WSNs is to save energy for the purpose of extending network lifetime. Topology control provides energy saving in three ways:

- It allows nodes to adjust their transmit power in which reducing the transmit power will result in a shorter transmission range. The reduced transmit power causes the long distance communication links that are not energy-efficient to be dropped. Instead, the short distance communication links are chosen. From the perspective of energy consumption, communication over short distance links is more energy-efficient than over the long distance links [San05a].

- It can switch nodes to sleep mode. This minimizes the energy consumed for idle listening and overhearing since it is found to be significant relative to the energy consumption of sleeping nodes [FN01].

- It can select a certain set of nodes to be the backbone of networks. The backbone formation serves to reduce the amount of traffic involved in packet forwarding and routing.

All these three ways eliminate redundant links and consequently restrict the neighbor size in the network

2.2.1 Topology Control Terminology

In the research literature of WSNs, the word "topology control" has been used in many contexts. For example, this term is commonly adopted to describe the technique that vary their transmission range with the aims of minimizing energy and prolonging the network lifetime [BJ02, BWZ04, LHB$^+$05, LLM$^+$05, San05b, PS03, WWK03]. This technique is referred to as power adjustment in Section 2.2.4.

In this thesis, the term topology control is defined with a larger scope. This term does not restrict topology control to the techniques of adjusting the transmit power (power adjustment) as in [BJ02, BWZ04, LHB$^+$05, LLM$^+$05, San05b, PS03, WWK03]. Instead, the term considers the following four techniques as topology control: power adjustment, power mode, clustering and hybrid methods as shown in Figure 2.1. All these techniques can tune the parameters of nodes to control the network topology. For example, the power adjustment can adjust the node's transmission range, the power mode can exploit the operating modes of nodes, the clustering can organize nodes into a hierarchical network and the hybrid combines the clustering techniques with either the power mode or the power adjustment to gain further energy saving. These four techniques are later used in Section 2.2.4 to categorize the topology control algorithms.

2.2.2 Design Criteria and Cost Metric

When designing topology control algorithms, important criteria that influence the design of the algorithms should be addressed carefully. These criteria will be used as cost metrics in Section 2.4, to compare the performance among the surveyed topology control algorithms. Specifically, message complexity and time complexity are the two typical measures used in numerous works, namely in

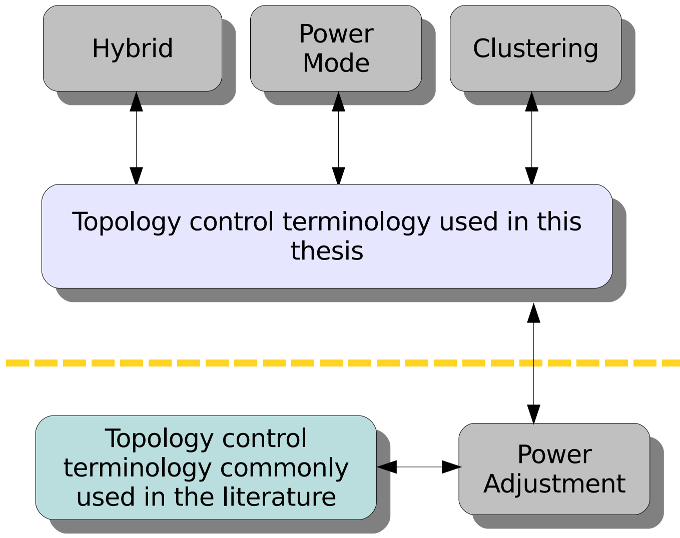


Figure 2.1: The terminologies of topology control.

[BDTC04, WAF04, YWD09] for evaluating the quality of the topology control algorithms.

The criteria are discussed below:

Distributed Construction: Topology control algorithms can be computed either in centralized or distributed manner. A centralized computation is a poor solution because of its unacceptable communication overhead involved in gathering information and its reliance on the existence of centralized administrators, which are typically absent in WSNs. Distributed computation on the other hand does need centralized administrators, thus is more practical.

Localized Computation: Efficient algorithms should be constructed based on local information gathered from immediate neighbors via exchanged messages. This feature allows the algorithms to be simple, scalable and configurable.

Low Message Complexity: In order to construct a topology, sensor nodes require sufficient information to make a decision. This information is provided through exchanged messages among neighbors. The message complexity or communication overhead is defined as the maximum number of messages needed to be broadcast by each node in the worst case. High communication overhead contributes to high energy consumption, which can shorten a node's lifetime; hence it should be avoided.

Low Time Complexity: Time complexity measures the amount of time an algorithm needs to run in order to make topology control decisions. Time complexity indicates the speed of a given algorithm in the worst case. Typically, a simple algorithm requires less time to run. In WSNs, nodes frequently move and/or fail, therefore requiring frequent topology updates. Topology control algorithms should be easily re-configured to quickly respond to network changes.

Strong Connectivity: The connectivity feature is required to support data communications in a network. The network is connected if there is a path connecting two nodes directly or through multiple hops. A network generated by a topology control algorithm often has fewer links than the network generated using maximum transmit power because redundant links are removed. It is important for topology control algorithms to preserve connectivity after the removal of redundant links.

Node Mobility Consideration: Node mobility can affect the performance of topology control algorithms. In the presence of node mobility, topology control algorithms require frequent message exchanges to continuously update topology changes. This could entail significant message overheads and increase energy consumption. Topology control algorithms should be explicitly designed to account for node mobility so as to minimize energy consumption.

2.2.3 Definitions of Network Lifetime

Network lifetime is a primary metric used to measure the energy-efficiency of WSNs. In general, there is no unique definition of network lifetime for WSNs as it heavily depends on the objectives determined by an application. For example, a monitoring application may require a fraction of working sensor nodes to maintain sensing coverage of each region in order to collect temperature readings. The lifetime of just this fraction of "alive" nodes determines the lifetime of the entire network, as long as coverage is kept at a sufficient level. On the other hand, in other applications the lifetime might be defined by the first node that fails to forward packets to the base station.

Despite no universal definition adopted for WSNs, some definitions are commonly used, sometimes with multiple definitions for a given application. Some of these more common definitions are provided below:

The first node to die: The first node which fails in the network is used to define the network lifetime [WGS01]. The failed node is often called a critical node.

The number of alive nodes: The number of alive nodes as a function of time is taken as a measure of network lifetime. In [LW01, HCB00] the number of alive nodes for several algorithms are recorded at time t and the algorithm with the highest number of alive nodes is considered to outperform the rest of the algorithms in terms of network lifetime.

The fraction of alive nodes: The network lifetime is described by the fraction of survived nodes as a function of time [YHE01]. The fraction of alive nodes can be set to a certain threshold value. The network is alive if the fraction of survived nodes is above the threshold value set.

The time until the network fails to construct a backbone: The time until the network can no longer construct a backbone [YJY06] is typically used to define the longevity of a network employing a clustering method.

The fraction of connected dominating set (CDS) nodes remaining alive: It is commonly used in CDS techniques to assess the lifetime based on the fraction of connected dominating set nodes that remain alive [CJBM02]. The fraction of alive connected dominating set nodes can be set to a certain threshold value. The network fails if the fraction of alive connected dominating set falls below the threshold.

The time t until the packet delivery ratio drops drastically: The network dies at time t when the packet delivery ratio, typically set to a pre-defined threshold, drops below this value.

The number of nodes which remain connected to the base station: The survival rate of the network is evaluated based on the number of nodes remaining connected to the base station [LW01]. It captures the issue of connectivity of the network to the base station. The number of nodes that must stay connected to the base station can be predetermined.

2.2.4 Classification of Topology Control Algorithms

In general, energy-efficient topology control algorithms can be broadly classified into centralized or distributed algorithms depending on the approach adopted for constructing the networks. Centralized approaches such as [Hu93, JO02, RKH00] can provide accurate global information but their implementations are expensive in practice due to significant communication overheads required for gathering information [Sto05]. These approaches are not feasible for WSNs that typically have a large number of sensor nodes. Distributed approaches will be the main focus in this thesis since they are more practical for dynamic networks and large-scale deployment scenarios.

Distributed topology control algorithms in this chapter are classified according to their energy conservation technique. According to this criterion, they are grouped into four categories, as shown in Figure 2.2. The four categories are described below:

Power adjustment: deals with a technique that reduces energy consumption by varying the transmission power of nodes.

Power mode: saves energy by switching-off the radios of idle nodes and placing them into a sleep mode.

Clustering: conserves energy by carefully selecting a set of neighbor nodes that can construct an energy-efficient backbone in the network.

Hybrid: gains the energy saving by integrating the clustering approach with either the power mode or power adjustment approaches.

2.3 Distributed Topology Control Algorithms

In this section, a discussion of a representative set of distributed algorithms is provided. These algorithms are assigned into one of the four categories described in Section 2.2.4. Three leading algorithms are chosen to represent each category,

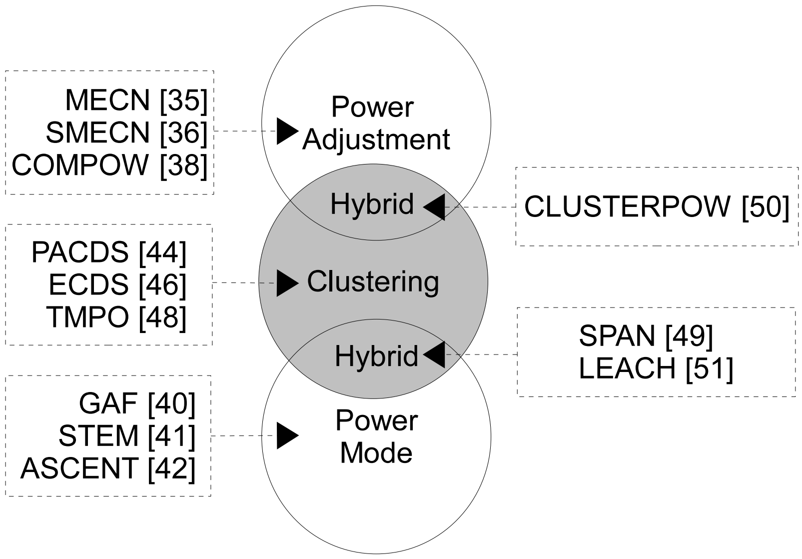


Figure 2.2: The four categories of distributed topology control algorithms and twelve representative algorithms.

as shown in Figure 2.2.

2.3.1 Power Adjustment Approaches

The power adjustment approach is based on the concept of modifying the transmission power of nodes to minimize the energy incurred during communication. This is due to the fact that the communication is the major source of power consumption in WSNs [RSPS02]. To reduce the energy, nodes avoid using maximum power to transmit but instead work in a collaborative manner to adjust the transmission power to exactly cover its furthest neighbors to form a connected network. That is, the low transmission power is used to remove the long inefficient links in the network, leading to the use of short multihop links for routing. This approach generates a sparse topology that is easy to maintain and good for reducing contention and energy consumption. Three power adjustment algorithms are presented in this section.

Minimum Energy Communication Network (MECN) Rodoplu et al. [RM99] proposed a localized and position-based algorithm that minimizes the energy involved in transmission of packets in a WSN. The idea of this algorithm is to construct a topology consisting of lowest energy paths to transmit from any wireless sensor of the network to a sink node by using the concept of “relay transmission”.

The MECN algorithm operates in two phases. In the first phase, each node finds its neighbor set. Note that, the authors do not use the ”neighbor set” definition in the conventional sense in their paper. Usually, a neighbor set of a node contains all the nodes that are within its communication range. Here, a node adds into its neighbor set only the ones it can communicate directly by spending minimum packet transmission energy. In other words, a node only accepts another node into its neighbor set if

- it can communicate with this node directly, and
- there is no other way of communicating with this node by using relays and spending smaller amount of transmission energy than direct communication.

Figure 2.3 illustrates this distinction. Algorithm 1 describes the neighbor set construction process.

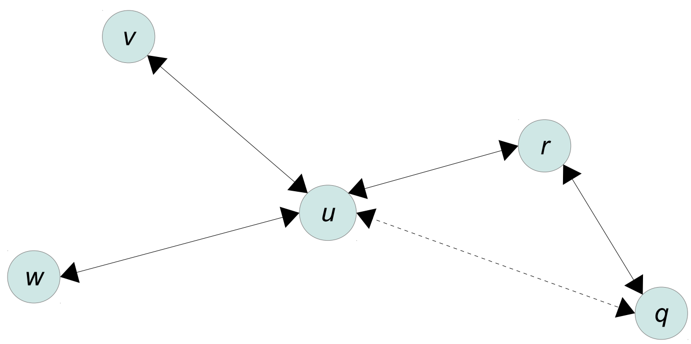


Figure 2.3: Neighbors of the node u: It can directly exchange packets with v, w, r and q, but discovers that relayed packet transmissions to q via r is more efficient then direct communications. So, it does not include q in its neighbor set.

In the second phase, the nodes run the Bellman-Ford shortest path algorithm to determine the minimum energy path to the sink node. Each node broadcasts the cost of using itself as a relay towards the sink (here, the cost is the minimum power consumption required to send a packet to the sink). When a node u receives the cost information from a neighbor node v, it calculates the minimum cost of the path to the sink relayed through v as

$$\mathrm{Cost}(u, v) = \mathrm{Cost}(v) + \mathrm{d}(u, v)^{\mathrm{n}} + \beta \tag{2.3.1}$$

where, $\mathrm{d}(u, v)$ is the Euclidean distance between the nodes u and v (it is assumed that the nodes know their locations), n is the path loss exponent and β is the power consumed at a receiver acting as a relay node.

Based on the costs, node u chooses a path which involves minimum packet transmission cost among its neighbors. The chosen node with the minimum cost is the next node to initiate the minimum energy path construction. The cost calculations are kept updated and broadcast to neighbors. To further optimize the energy consumption, a node can switch to sleep mode after the completion of the second phase if it is not transmitting any messages.

To handle the dynamic changes in the environment (fluctuations in the propagation paths, faulty nodes etc.), the MECN algorithm also includes a mechanism called "Flip" by the authors. It is used to handle the following cases:

1. nodes are removed from the neighbor set if it is found that direct communication with them is not efficient any more (i.e., it is possible that, due to the dynamic changes in the environment, communication with this node could become more efficient if another neighbor is used as relay), or
2. a node is added to the neighbor set since direct communication with them

become more efficient. A freshly added node triggers the cases mentioned in item 1 above.

For the details of the Flip mechanism, [RM99] can be referred.

Algorithm 1 Discovery of neighbors that are energy-efficient to communicate

---

$\mathrm{P}_{u \to v \to q}$ is the total transmit power used for sending a packet from node u to q via node v

$N(u)$ is the neighbor set of node u that is energy-efficient to communicate directly

procedure FINDNEIGHBORSET(u)
  $N(u) \leftarrow \emptyset$
  for all received beacon packets **do**
    $q \leftarrow$ Sender of the beacon
    if $q \notin N(u)$ **then**
      $\mathrm{P}_{u \to q}$ ▷ Compute the power cost of sending a packet from node u to node q
      neighbor_candidate $\leftarrow$ true
  ▷ Check whether transmission via relay node is energy-efficient than direct transmission
      for all $v \in N(u)$ **and** neighbor_candidate = true **do**
        if $\mathrm{P}_{u \to v} + \mathrm{P}_{v \to q} < \mathrm{P}_{u \to q}$ **then**
          neighbor_candidate $\leftarrow$ false
        end if
        if neighbor_candidate $=$ true **then**
          $N(u) \leftarrow N(u) \cup \{q\}$
        end if
      end for
    end if
  end for
end procedure

Small Minimum Energy Communication Network (SMECN) The SMECN algorithm [LW01] is an extension of the MECN algorithm. It aims to construct a network that is simpler, faster and more energy-efficient than the one generated in MECN [RM99]. The objective of SMECN is to generate a topology which is smaller than the topology of MECN. Being a variant of MECN, SMECN uses the same energy model and assumptions as in MECN. The implementation of

SMECN also consists of two phases that resemble MECN as indicated in Figure 1. The only difference between SMECN and MECN lies in the method of determining the nodes for the enclosure graph. Unlike in MECN, SMECN has no "Flip" mechanism. An energy-efficient reconfiguration algorithm that is based on SMECN was later proposed in [LH04]. The proposed algorithm was able to construct a minimum energy graph under dynamic topology changes.

Common Power (COMPOW) The energy conservation strategy in COMPOW [NKSK02] finds and uses the minimum common power level that is sufficient to maintain the connectivity of the entire network. Based on theoretical studies in [NKSK02], it is argued that the minimum common power level can provide several benefits to networks including improvement in the traffic carrying capacity, energy consumption and contention at the MAC layer. The choice of using the smallest common power level also results in bidirectional links, an important feature required for efficient routing and proper communication at MAC layer. This protocol is the first that was implemented in a real wireless testbed and explored various power levels available in a CISCO 350 series Aironet wireless network interface card. COMPOW combines both power control and routing due to the fact that they both affect each other.

COMPOW adopted parallel modularity at the routing layer to achieve asynchronous and distributed operations. This is done by having each node running several routing daemons in parallel, one daemon for each transmit power level P. Thus, each node constructs multiple routing tables for all available power levels through exchanged hello messages. Initially, each node constructs a routing table using the maximum power level to find all nodes in the network. Then, it constructs a routing table for all power levels and finds the smallest power level whose entries of the routing table are equal to the entries of the routing table at the maximum power level. The smallest power level is chosen as the optimum power level and its routing table is installed as the master routing table to be used by the kernel to transmit packets between nodes.

2.3.2 Power Mode Approaches

The power mode approach is the technique that exploits the feature of the operating mode available in the network interface of sensor nodes to gain energy saving. There are four operating modes of the nodes: sleep, idle, transmit and receive modes. The energy consumed during the transmit and receive modes is generally higher than that in the sleep mode [JSAC01]. In order to transmit or receive packets, nodes must transit to idle mode. However, continuous listening of incoming packets that are not addressed to the idle nodes always contribute to high energy dissipation that is quite significant compared to that in sleep mode [FN01]. This suggests that the redundant nodes sitting in idle can be switched to energy saving mode by placing them in the sleep mode. This feature has been used in topology control to optimize the energy and prolong network lifetime without sacrificing network capacity and connectivity. In this section, a discussion of three power mode algorithms that deal with powering-off idle nodes as well as coordinating their sleep and wake-up scheduling is presented.

Geographical Adaptive Fidelity (GAF) The main ideas of GAF [YHE01] are to have a sufficient number of nodes to remain in active communication and place the redundant nodes to sleep mode without affecting the network connectivity. In order to identify the active nodes from the redundant nodes, GAF divides the network area into small size virtual grids. All nodes are associated with these grids through the use of location information and an idealized radio model.

GAF uses the term "equivalent nodes" to describe nodes that are capable of communicating with all nodes in their adjacent grids. The equivalent nodes can be exploited to conserve energy by keeping only some of them alive for routing while the remaining nodes can remain asleep. Nodes with a longer expected lifetime are used first. For example in Figure 2.4 [YHE01], nodes v, w and x are equivalent nodes because in order for node u to communicate with node z it can relay packets through either v, w, or x. In this example, energy saving is achieved by placing node w and x into sleep mode while node v performs data forwarding and they alternate between sleeping and listening.

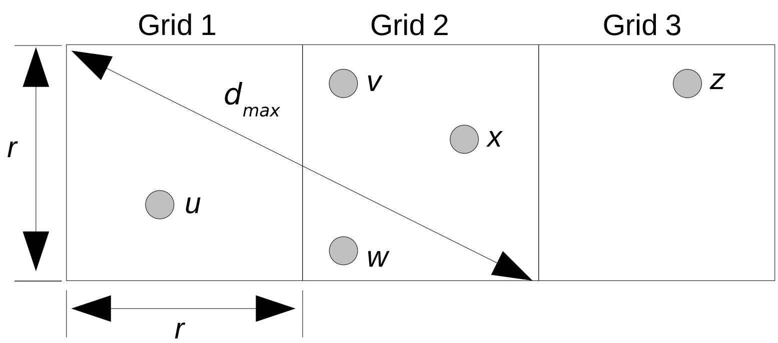


Figure 2.4: The virtual grid structure in GAF.

There are three states in which nodes operate in GAF as shown in the state transition diagram in Figure 2.5. The states of the nodes consist of sleeping, discovery and active. In discovery state, nodes identify their neighbors in the grid by turning on their radios and exchanging discovery messages. In the active state, nodes participate in routing. In the sleeping state, nodes turn off their radio and remain inactive. All nodes initially begin with the discovery state. During this state, nodes set their discovery time for T_d seconds, broadcast discovery message to locate nodes within the same grid and then enter active state. Nodes that enter into active state set their timer to a timeout value T_a to define the duration they stay in active state. After T_a, nodes will return to discovery state and rebroadcast their discovery message every T_d seconds. Nodes in discovery or active state may switch to sleeping state if they find other equivalent nodes to handle routing. When transitioning to sleeping, nodes cancel all pending timers and power down their radios.

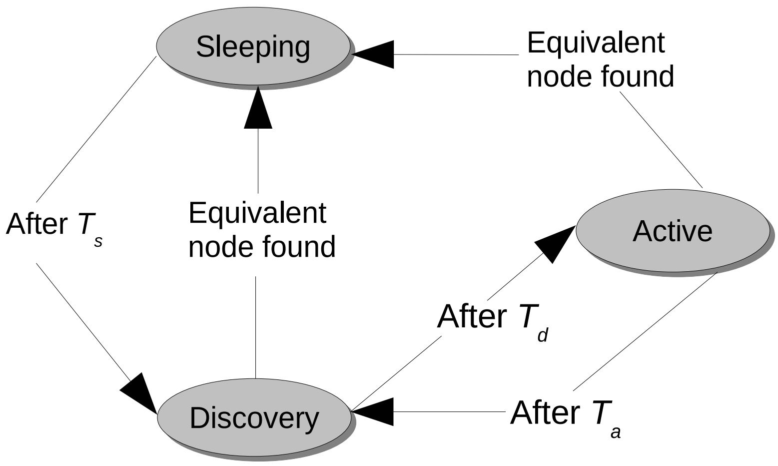


Figure 2.5: State transition of GAF.

Sparse Topology and Energy Management (STEM) Similarly to GAF, the principle of STEM [STS02] is to put as many nodes as possible to sleep mode so that the energy consumption is reduced and network lifetime is extended. Schurgers et al. [STS02] argued that this idea is relevant for a network that spends most of its time in monitoring activity and has less data forwarding activity. The idle nodes that are in monitoring activity can be powered down and woken up only when they have data to forward to the base station. The common challenge of the power down approach is to manage nodes' sleeping transition such that the asleep nodes are activated only when an event occurs. STEM solves this challenge by periodically turning on the node's radio for a short time to listen for incoming communications.

There are two operations involved in STEM, the wake-up and actual data transmission processes. The wake-up process ensures that the radio of sleeping nodes is turned on to allow nodes to listen for an incoming message, then the actual data transmission process ensures that data is safely transferred between a source and sink. In STEM, each node sends a wake-up message and transmits data in two different frequency bands using two separate radios to avoid interference. The wake-up message happens in the wake-up plane operating on radio frequency f_1 while data transfer happens in the data plane operating on radio frequency f_2. The operation of STEM is illustrated in Figure 2.6. Assuming that node v and w are sleeping, suppose that node u detects an event and wants to transmit data to the sink through node v and w. Node u then sends a wake up message to the target node v on radio frequency f_1 and waits for a response from node v. After receiving the response, both nodes turn on their radios and start data transfer on radio frequency f_2. This process is repeated between node v and node w with node v now becoming an initiator while node w becomes a target until the data is successfully received by the sink.

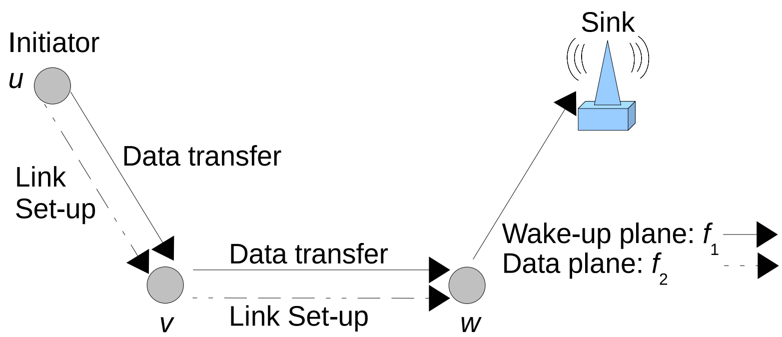


Figure 2.6: Radio setup of a sensor node in STEM.

STEM is later integrated with GAF scheme to achieve two objectives [STS02]. The first objective is to gain additional energy saving. GAF ensures that each grid must contain one active node acting as a leader, but this leader may not have data to transfer. Thus by running STEM on the leader in each GAF's grid, the leader which is sitting idle waiting for data transfer can be turned off to reduce power consumption. The result shows that in comparison to a network without any topology control, integrated STEM reduces the energy consumption by up to 7 percent. This improvement is equivalent to an increase by a factor of 14 in a node lifetime as reported in [STS02]. The second objective is to improve STEM's latency. STEM makes use of the leader election process in GAF to minimize the number of interferences during the wake-up process and speeds up the link set-up phase.

Adaptive Self-Configuring Sensor Network Topologies (ASCENT) ASCENT [CE04] is a self-reconfigurable algorithm that allows nodes to locally measure the operating conditions. Based on these conditions, nodes then decide whether they need to participate in routing or not. To achieve energy-efficiency, ASCENT selects a subset of nodes to remain active to serve as a routing backbone. The remaining nodes in the network stay passive listening to other nodes and periodically check in case they need to join the routing backbone. For instance, when the packet loss is high the passive nodes are activated to preserve connectivity. Otherwise these nodes turn-off their radio to conserve energy.

Nodes in ASCENT will stay in one of the four states, namely test, passive, active and sleep as depicted in Figure 2.7. Nodes in a passive state stay in listening mode to participate in routing if required, thus the radio of the nodes remains active. Nodes in the active state perform data forwarding and monitoring, while nodes in the sleep state switch-off their radios to save energy. Nodes in the test state check the conditions on whether the network has enough active nodes to maintain connectivity. If active nodes are insufficient, nodes in passive state will join the active nodes or else they switch to sleep state to reduce energy consumption.

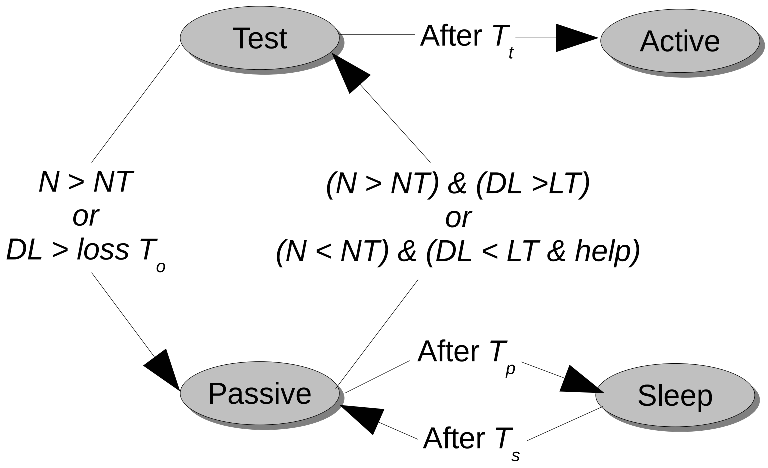


Figure 2.7: State transitions representing the operation of ASCENT.

All nodes initially remain in the test state. The nodes then set their timer T_t and send neighbor announce messages to discover their neighbors. While in the test state, a node will check whether the number of active neighbors N is above a neighbor threshold NT, or the average data loss rate DL is higher than the average loss T_o before entering into the test state. If the condition is true, the node transits into the passive state. The higher node ID in the announcement message is used to break a tie if multiple nodes compete to transit to the test state. If the condition is false, the node remains in the test state and later moves into the active state upon the expiry of timer T_t. In the active state, the node participates in routing until it runs out of energy. The active node sends help messages when DL is greater than the loss threshold LT.

A node that enters the passive state sets up a timer T_p. It sends new passive node announcement messages to be used by active nodes to estimate the total density of nodes in the neighborhood. While in the passive state, a node decides whether it has to transit to the test state to support the routing backbone or transit to sleep state to save energy. The decision to transit to the test state is made locally if either one of the two conditions is met: (1) the number of neighbors is below NT and DL is higher than LT or (2) the number of neighbors is below NT, DL is below LT and the node receives a help message from an active neighbors. Otherwise it will remain in the passive state until the timer T_p expires. The node later moves into sleep state and switches off its radio to conserve energy. Upon the expiry of the timer T_s, it will transit into the passive state.

2.3.3 Clustering Approaches

The idea of clustering is to select a set of nodes in the network to construct an efficient topology. The selection of neighbors can be made on various criteria namely, energy reserve, density of the network or node identifier. Unlike in power adjustment or power mode approaches, the clustering approach constructs a topology with hierarchical structures that are scalable and simple to manage. The advantage of clustering is that a certain task can be restricted to a set of nodes called clusterheads and they can be assigned for collecting, processing and forwarding packets from non-clusterheads. This mechanism provides an efficient network organization. Other attractive features of the clustering approaches include the load balancing and data aggregation offered for prolonged network lifetime. Load balancing allows the role of clusterheads to be rotated among nodes with a higher remaining energy to achieve fair load distribution. In some clustering approaches, the selection of the clusterheads remains fixed. Hence, clusterheads typically experience faster energy depletion because they are heavily loaded with various tasks. This problem is overcome by randomizing the selection of clusterheads to distribute loads fairly among nodes in the network.

Many of the clustering approaches construct the virtual backbone using the connected dominating set (CDS) concept [BDTC04]. A CDS has been widely used as a topology control to conserve network energy resources. A dominating set (DS) is defined as a subset of nodes in a graph such that each node not in the subset has at least one direct neighbor that belongs to the subset [DP04]. If the nodes in the dominating set form a connected graph, the set is called a CDS. Figure 2.8 shows an example of a CDS generated in a network that consists of fourteen nodes. In this figure, nodes u, v, w, x, y and z form the backbone to perform data forwarding while the remaining nodes do not participate in data forwarding. This strategy reduces the communication overhead and energy. The CDS is a well-known technique that offers scalability, provides various energy management strategies to extend the network lifetime and reduces the communication overheads in the network. Thus, the following section includes three CDS algorithms used for topology control.

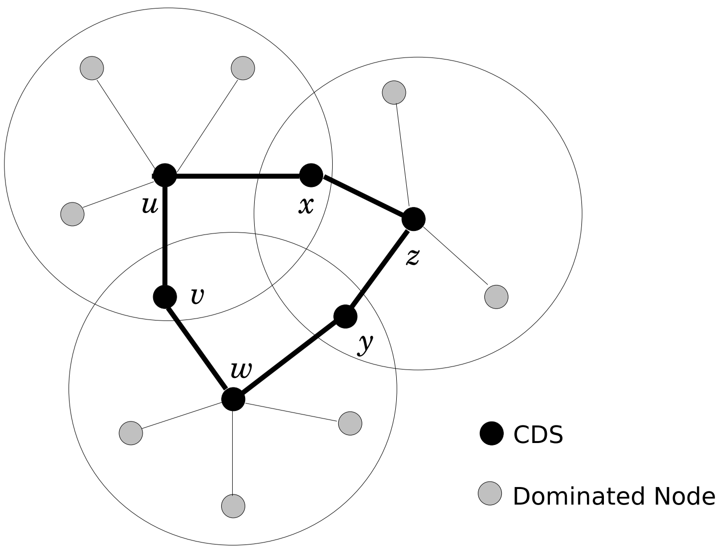


Figure 2.8: A backbone in the network built using a CDS.

The CDS algorithms can be classified into two categories based on the method used for constructing the CDS. They are single initiator and multiple initiator approaches.

Single Initiator Algorithm

The concept of single initiator algorithms is to find a CDS by growing a tree rooted at a sink node called initiator until all nodes are greedily added to the tree. Thus, the CDS generation is a sequential process. This section provides a single initiator algorithm.

Energy-efficient Distributed Connecting Dominating Sets (ECDS) Yuanyuan et al. [YJY06] presented ECDS to solve the energy constraints in wireless sensor networks and minimize the size of a CDS. ECDS constructs the CDS in two phases as illustrated in Figure 2.9. It first constructs a dominating set which is a maximal independent set (MIS) and then finds gateway nodes to connect the MIS. A MIS is defined as an independent set (IS) that is not a subset of any other IS [LW09]. An IS of graph G is a subset of V where no two nodes within the set have an edge. The notation V refers to a set of vertices V in graph G. Therefore, every MIS is a dominating set (DS) which is not connected.

ECDS uses a coloring technique to identify nodes during the CDS construction. Initially, all nodes are in white color and at the end of the first phase, nodes will either be in black or gray color. The black nodes form a MIS while the gray nodes are the non-MIS nodes. After the completion of the second phase, all nodes in the network are either in blue color or in gray color. The blue nodes form the CDS.

In ECDS, nodes regularly broadcast messages to notify and update their status and weight to neighbors. The nodes with the largest weight are chosen as the MIS and CDS nodes. The weight is calculated based on the node's residual energy and effective degree. The effective degree is defined by the number of neighbors each node has.

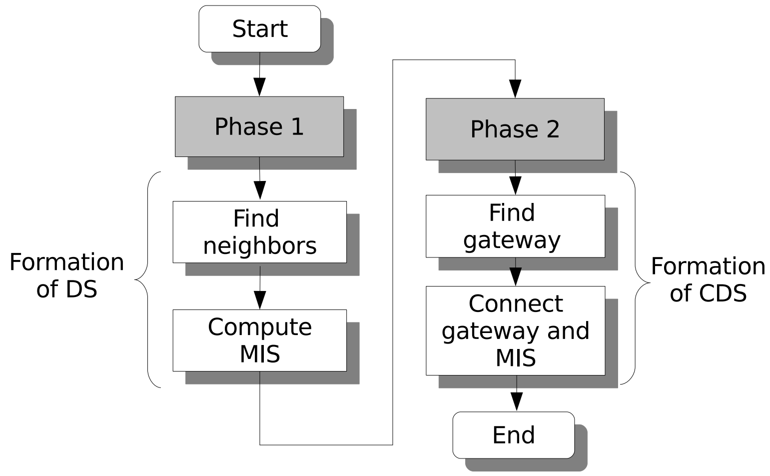


Figure 2.9: The construction of a CDS in ECDS.

The first phase starts with an initiator volunteering to be a MIS node and coloring itself black. It then sends a black message to its neighbors. The white neighbors that receive the message are colored gray (non-MIS nodes) and they broadcast a gray message to their white neighbors to update their color change. The white nodes receiving the gray message will then be chosen as MIS nodes if they have the largest weight. This process continues until there are no more white node left in the network. The second phase starts when all nodes in the network are in gray or black color. It begins with initiator sending a blue message to neighbors to find gateways to connect the MIS. The gray nodes with the highest weight that receive this message will become blue nodes. The CDS generation continues until all blue nodes are connected and there are no more black node left in the network.

Multiple Initiator Algorithms

In multiple initiator algorithms, multiple nodes are simultaneously elected as initiators to generate a CDS. Two distributed multiple algorithms are presented in this section.

Power Aware Connected Dominating Set (PACDS) Wu et al. [WGS01] proposed a simple algorithm based on the CDS concept. PACDS finds a CDS using a simple marking process. PACDS improves the work proposed in [WL99] to achieve two goals. The first goal is to construct a small size CDS while the second goal is to prolong the lifetime of nodes. In a CDS, nodes in the backbone are commonly overloaded with various tasks and they are the first to be drained of energy in the network. Load balancing can overcome this problem by randomizing the role of the backbone among nodes with higher remaining energy.

The construction of PACDS involves two stages as shown in Figure 2.10. The first stage is the formation of a CDS. Initially, node u broadcasts a hello message to its neighbor to gather neighbor information. If node u has two unconnected neighbors, it will be marked as a CDS node. The second stage is the pruning process, in which redundant CDS nodes are removed to reduce the size of the CDS constructed. The pruning process is required because the size of the CDS formed during the first stage is not minimal. Two rules that are based on node ID are used for the CDS removal. The pruning rules state that if node u has a neighbor with higher ID which can cover all of its neighbors or if u has two connected neighbors with higher ID which can cover all of its neighbors, u can be eliminated from the CDS.

Other than using node ID, PACDS introduces additional pruning rules to the CDS using node degree and residual energy. The first additional pruning rule uses node degree with an aim to keep the size of CDS as small as possible. The second additional rule uses the residual energy to gain prolonged node lifespan. The residual energy rule gives higher priority to nodes with a higher energy level to become a clusterhead and removes the lower energy level nodes from the CDS.

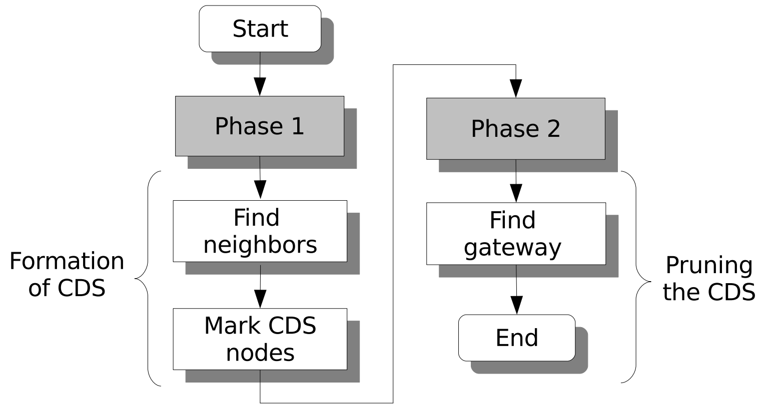


Figure 2.10: The generation of a CDS in PACDS.

Topology Management by Priority Ordering (TMPO) TMPO [BGLA03] is a dynamic algorithm that considers movement and residual energy when forming a backbone. TMPO introduces the concept of gateways and doorways (which is adopted from clustering method) to connect the dominating sets. There are several outstanding features of TMPO. First, the formation of minimal dominating sets and CDS is free from any negotiation process, thus unnecessary overheads involved during the clusterhead election are avoided. Second, an identifier called node priority is calculated periodically, allowing the role of clusterhead to be rotated among nodes to extend the network lifetime. Third, the algorithm incorporates the mobility and energy capacity of nodes through the use of parameter known as willingness value. Fourth, apart from gateway and clusterhead, TMPO adds a new function called doorway that bridges two clusterheads.

There are two phases involved in the construction of a CDS as shown in Figure 2.11. The first phase is the clusterhead election process. It finds the clusterheads that can create a minimal dominating set in the network to minimize the size of the CDS. The selection of clusterheads is made according to the priority rule. A node becomes a clusterhead if it has the highest priority among its one-hop neighbors or its two-hop neighbors. The priority considers the identifier of the node's neighbor, present time and willingness value. The willingness value is assigned to each node as a function of node mobility and energy level. The priority values are changed periodically to provide random election of clusterheads and these values are unique.

In the second phase, doorway and gateway nodes are elected and they connect the minimal dominating set generated in the previous phase to form a CDS. A doorway node is described as the node that can connect two clusterheads that are separated three hops away and there are no other clusterheads between them. The doorway must have the highest priority between the two clusterheads. The gateway node is defined as the highest priority node that can connect two clusterheads two hops away or connect one clusterhead and one doorway separated two hops away and there are no other clusterheads between them. After the election of gateway and doorway nodes, the CDS is formed.

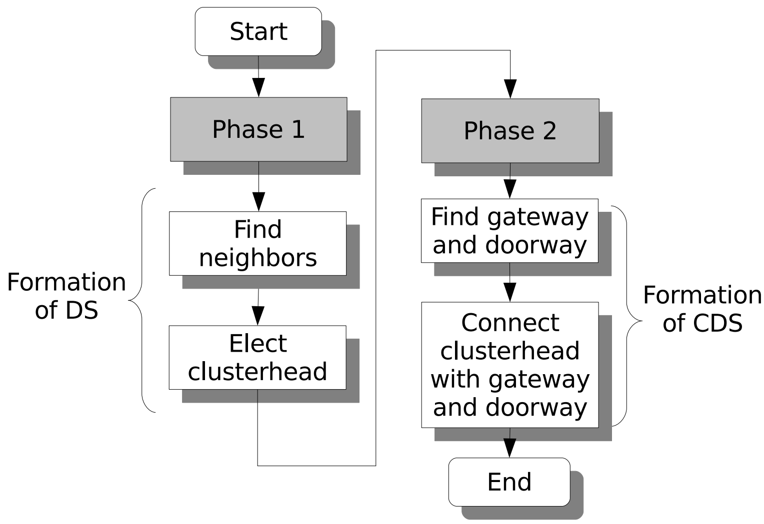


Figure 2.11: The construction of a CDS in TMPO.

2.3.4 Hybrid Approaches

The hybrid approach is a topology control technique that uses a clustering approach in combination with power adjustment or power mode approaches. For instance, SPAN [CJBM02] combines the clustering approach with the power mode approach in which the non-clusterheads that are sitting idle are switched to the sleep mode. Another example is the CLUSTERPOW [KK03] algorithm which integrates the clustering approach with the power adjustment approach to achieve additional energy saving. The following section discusses three hybrid algorithms that aim to conserve energy.

Energy-efficient Coordination (SPAN) The SPAN [CJBM02] algorithm is a hybrid of power mode and clustering approaches. It selects a subset of nodes to form a forwarding backbone using a CDS approach. The backbone is capable of forwarding packets, maintaining network connectivity and preserving network capacity. Based on local decisions, nodes in SPAN decide whether they should join or sleep in the forwarding backbone. Nodes in the forwarding backbone are called coordinator nodes while the remainder are called non-coordinator nodes. Non-coordinator nodes remain in a sleep mode to save power and periodically wake up to exchange traffic with coordinator nodes. They constantly check whether they need to participate in coordinator election or coordinator withdrawal. One of the main features of SPAN is the use of the power saving features of 802.11 to improve routing throughput and packet delivery latency. Using SPAN on top of 802.11 power saving mode allows packets sent to a sleep node to be stored temporarily at its neighbor. The packets are later retrieved when the node wakes up, thus preventing packet loss.

SPAN is designed to meet the following four goals [CJBM02]. First, it elects a sufficient number of coordinators such that every node is in the radio range of at least one coordinator to provide network connectivity. Second, it employs a load balancing technique that rotates the coordinators so that the coordinator task is fairly distributed among all nodes. Third, it tries to minimize the number of coordinators (CDS) without suffering a significant loss of capacity or increased latency. As a result, the connected dominating set in SPAN may not be the minimal connected dominating set. Fourth, the coordinator election is based on local information gathered at each node. Each node consults the state stored in local routing tables during the election process.

The operation of SPAN is governed by two processes called coordinator election and coordinator withdrawal. The information needed for a node to withdraw or elect itself as a coordinator is exchanged among neighbors via HELLO messages. During coordinator election, a non-coordinator node periodically determines if it should become a coordinator node or not based on the coordinator eligibility rule. The rule states that a non-coordinator node will become a coordinator if it finds two neighbors which cannot reach each other directly or through one or two coordinators. In the case of announcement contention, when multiple nodes decide to become a coordinator at the same time, SPAN uses a random back-off delay to resolve the contention. The potential coordinator node delays the coordinator announcement and re-sends the coordinator announcement at the end of the delay to re-evaluate its eligibility. It then becomes a coordinator if the eligibility rule is still valid. The back-off delay is a function which considers the energy level of nodes and the ability of nodes to connect additional pairs of nodes among their neighbors. The energy level in this case refers to the ratio of the remaining energy level to the maximum energy supply at each node.

During coordinator withdrawal, each node periodically checks if it should withdraw as a coordinator. The rule states that a coordinator must withdraw if each pair of its neighbors can reach each other directly or via one or two other coordinators. The withdrawal coordinator is marked as a tentative coordinator and it remains in this state for a certain duration of time, W_T before withdrawing its coordinator status if other nodes can connect its neighbors. This process also ensures that the coordinator's role is rotated among other nodes. Results of SPAN indicate that it can preserve network connectivity, maintain capacity and provide significant energy savings. SPAN simulations show that the system lifetime with SPAN is more than a factor of two better than without SPAN.

Cluster Power (CLUSTERPOW) The CLUSTERPOW [KK03] algorithm joins the clustering approach with the power control approach to gain network connectivity, network capacity and energy-efficiency. The design of CLUSTERPOW is motivated by the limitation of COMPOW [NKSK02] in dealing with non-homogeneous node distributions. The choice of using a minimum common power level in COMPOW is not appropriate for non-homogeneous networks because the lowest common power level is determined by a faraway node. For illustration, consider node u in Figure 2.12. All the nodes within the cluster $C1$ use the power level 1mW to communicate. When a node w joins the network, the rest of the nodes in cluster $C1$ are forced to use unnecessarily higher power level of 100mW to communicate with node w. As a result, the minimum common power level is set to a much higher level. As a solution, CLUSTERPOW offers a joint topology control and routing solution that selects an optimum minimum power level for each cluster. CLUSTERPOW provides implicit clustering which means that the small transmit power control chosen automatically creates clusters. Consequently it has no clusterheads or gateways.

Similarly to COMPOW, CLUSTERPOW requires each node to keep separate routing tables, one for each power level constructed using exchanged HELLO messages. It also employs parallel modularity at the network layer by running multiple routing daemons as in COMPOW. When node u has a message to send to node v, it computes the lowest transmit power level P such that the destination is

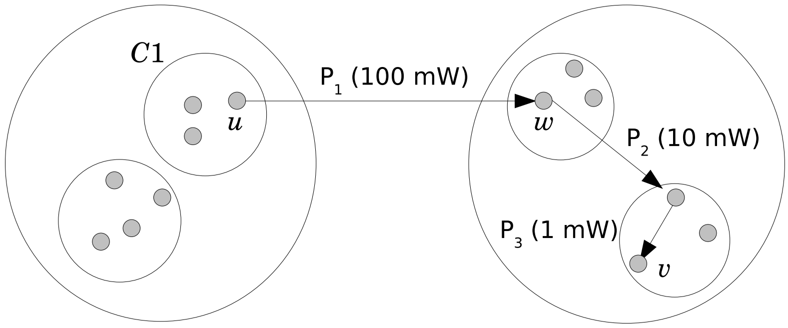


Figure 2.12: The CLUSTERPOW multihop routing uses a smaller power to create a non-homogeneous network.

reachable in multiple hops by using the power levels smaller than P. This process is executed at the source, and at every intermediate node along the route from the source to the destination. For example, the network in Figure 2.12 has three levels of clustering corresponding to power levels of 1mW, 10mW and 100mW. To transmit from node u to node v, a power level of 100mW is used at each hop until the packet gets to the 10mW cluster. Then 10mW is used at each hop and the transmit power is lowered down to 1mW as the packet gets closer to the destination.

Low-Energy Adaptive Clustering Hierarchy (LEACH) LEACH was introduced in [HCB00] to reduce energy consumption in a wireless sensor network (WSN) by means of clustering, data aggregation, load balancing and TDMA/CDMA. LEACH integrates the clustering approach and power mode approach to prolong network lifetime. In WSNs, a considerable amount of energy is involved when all nodes participate in data transmission over long distance. The use of clustering can minimize the energy spent by limiting the number of nodes that participate in long distance transmissions. In clustering, only clusterhead nodes can transmit data to the base station. To compress the amount of transmitted data, LEACH assigns data aggregation and fusion tasks to the clusterhead nodes. LEACH uses a load balancing mechanism that periodically rotates the role of clusterhead nodes. Fair and uniform election of clusterhead nodes are also used to ensure that nodes die randomly. LEACH uses TDMA to reduce intra-cluster communications which solves collision, hidden problems, overhearing and idle listening. This is achieved by switching-off the radios of non-clusterhead nodes when they are not in use. LEACH uses CDMA to overcome the collisions among clusterhead nodes competing for simultaneous data transmissions to the base stations.

The operation of LEACH is divided into rounds with each round consisting of two phases. Figure 2.13 shows the two phases. The set-up phase is responsible for cluster formation while the steady-state phase is responsible for data forwarding operations to the base station. The set-up phase begins with clusterhead election. The clusterhead election is rotated in each round to provide uniform load distribution and extend the node's lifetime. Clusterheads are elected randomly according to two criteria. The criteria are based on the suggested percentage of clusterheads (decided a priori) and the number of times the node has been a clusterhead. Therefore, the chances of a node becoming a clusterhead are low if it has been selected as a clusterhead in the previous round. The elected clusterhead nodes then broadcast their election to the rest of the nodes in the network. The non-clusterhead nodes that receive the broadcast will then measure the signal strength received in order to choose a cluster. The nodes will join the cluster with the largest signal strength value and inform the clusterhead nodes of their decision. This information is required by the clusterhead nodes to create a TDMA schedule for each member in the cluster.

In the steady-state phase, nodes that are scheduled for data transmission will begin their data transmission to the clusterhead node. The nodes that are not scheduled for transmission will switch to sleep mode to conserve power. The data received by the clusterhead nodes are aggregated or fused to compress the size before being sent to the base station. After a certain time, the next round starts again and the two phases are repeated.

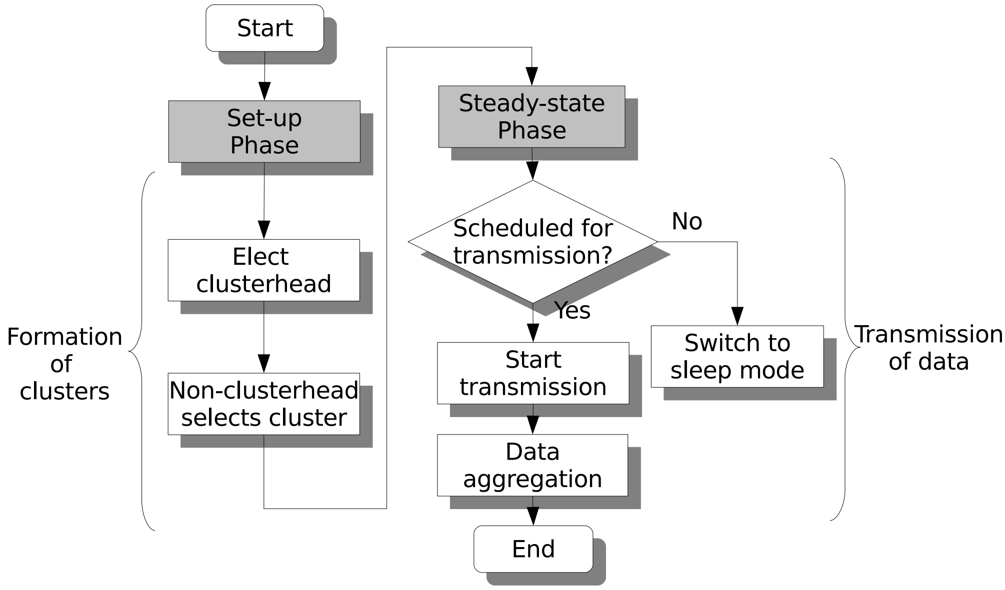


Figure 2.13: The construction of clusters and integration of the power mode approach in LEACH.

2.4 Comparative Evaluation of Distributed Topology Control Algorithms

So far, twelve algorithms have been discussed and classified into four categories based on the energy-efficient mechanisms that they use for constructing a network topology.

This section provides a detailed evaluation of the performance of the algorithms, in which they are first compared against the cost metrics in Section 2.2.2 and then thoroughly discussed in Section 2.4.1. Based on this discussion, the overall comparison among them which include the advantages and disadvantages is given. Finally, the detailed discussion of the network lifetime definition used in each algorithm is provided in Section 2.4.2 to highlight the strengths and weaknesses of the definition.

Algorithm	Localized	Time Complexity	Message Complexity	Connec-tivity	Mobility
Power Adjustment Algorithms					
MECN	Yes	$O(V^3)$	n/a	High	Low
SMECN	Yes	$O(V^2)$	n/a	High	Low
COMPOW	No	n/a	$O(Pn)$	Low	Low
Power Mode Algorithms					
GAF	No	n/a	$O(V)$	Low	Low
STEM	Yes	n/a	n/a	Low	No
ASCENT	Yes	n/a	n/a	Low	No
Clustering Algorithms					
PACDS	Yes	$O(\Delta^2) - O(n^3)$	$O(n\Delta) - O(n^2)$	Low	Low
ECDS	Yes	$O(n)$	$O(n)$	Low	No
TMPO	Yes	n/a	n/a	High	High
Hybrid Algorithms					
SPAN	Yes	$O(n)$	$O(n)$	Low	Low
CLUSTERPOW	No	n/a	$O(Pn)$	Low	Low
LEACH	Yes	$O(n)$	$O(n)$	Low	Low

Table 2.1: Comparison of the algorithms. In the table, n represents the total number of nodes, V is the number of neighbors, P is the number of transmit power levels, and Δ is the maximum degree in the graph. Also, n/a represents the cases if the authors of a study do not provide the relevant information.

2.4.1 Overall Comparison of the Topology Control Algorithms

In this section, the cost metrics are used to measure the quality of a network topology constructed by the twelve algorithms. To provide fair comparison among the algorithms, the cost comparison and its discussion are presented according to the previously presented four categories namely power adjustment, power mode, clustering and hybrid. Table 2.1 summarizes the comparison of the algorithms in all four categories.

Power Adjustment Approach The discussion of the three power adjustment algorithms is provided in this section.

MECN guarantees strong connectivity of the network. In the worst case condition, every node is able to maintain communication links with all the nodes inside its enclosure [RM99]. MECN also generates a sparse network, which means that the number of links increases linearly with the number of nodes in the network. The impacts of the sparse network are the reduction in the level of interference and improvement in the energy conservation. However, there are several assumptions made in the MECN algorithm. The assumption that all nodes know their exact location in the deployment region by means of a global positioning system (GPS) is impractical. This is due to the message overhead incurred for updating location information and also installing additional hardware. MECN also assumes that each node can communicate with all its neighbors and neglects the obstacles that usually exist in the deployment region between two nodes. Another downside of MECN is its reliance on an explicit propagation channel model to compute the relay region and enclosure graph [San05a]. For example, in order to determine the lowest energy route, nodes need to compute all the possible routes based on the actual transmit power level. Therefore realistic propagation channel conditions must be used when computing the optimum topology. One main challenge reported in [RM99] is to limit the search region so that the algorithm terminates. When nodes are highly mobile, the computation of relay nodes and enclosure region can be energy consuming. The time complexity of MECN given in [SCCZ07] is $O(V^3)$, where V is the number of neighbors of a node. Even though the message overhead of MECN is not provided, it is believed that a considerable message overhead is also introduced during the second phase, in which MECN relies on global information to compute the best topology. MECN is designed for static or slowly changing networks. Nevertheless, because of its localized property it is also appropriate for mobile networks but possibly at the expense of a relatively high message overhead.

SMECN uses the same network model and energy model as in MECN [RM99].

Thus it exhibits all the advantages and disadvantages of MECN which are relevant to these two models. SMECN outperforms MECN in terms of power efficiency and time efficiency due to its smaller generated subgraph. The time complexity of SMECN is $O(V^2)$ [SCCZ07]. It converges faster than MECN because the subgraph constructed has a fewer number of links that can also result in lower link maintenance costs and achieve a significant energy saving [LW01]. It is observed that during neighbor search, the choice of transmit power is influenced by network density. For instance, a lower transmission power is sufficient to enclose a dense network whereas a much higher transmission power is needed to enclose a sparse network. This suggests that SMECN may not be a power efficient solution for sparse networks where the maximum transmission power is frequently used [SCCZ07]. In such cases, the battery of nodes can quickly drain and shorten the network lifetime.

COMPOW has a modular structure that allows topology control to be plugged into any proactive routing protocol, thus making it flexible. However there are several shortcomings of COMPOW. The first shortcoming is its significant message overhead. Each node runs six different power levels and exchanges significant link state information with other nodes to set the optimum power level. This process creates extra message overhead, which may exhaust the nodes' energy reserve and shorten the nodes' lifetime. The decision on setting the optimum power level is also made based on the global information provided by various routing tables running multiple power levels. As a result, it incurs a considerable message overhead to maintain and update the network topology. In the worst case scenario COMPOW's message overhead is $O(Pn)$ [SBC03], where P is the number of power levels used by nodes while n is the total number of nodes in the network. Obviously, a significant message overhead is required if nodes run more power levels. In practical situations P can reach as high as 10 as reported in [San05b]. The second shortcoming of COMPOW is apparent in a non-homogeneous network whereby nodes are required to converge to a much higher common power level set by a new node joining the network that resides faraway from the majority of the nodes. As a result, a higher transmit power is used to maintain the network graph, thus defeating the purpose of minimizing power consumption through the use of a minimum common power level.

Power Mode Approach The discussion of the three power mode algorithms is provided in this section.

GAF connectivity is very much influenced by the network density and the accuracy of nodes' radio model. In dense networks, the connectivity and routing fidelity are guaranteed by the existence of multiple communication paths. But in sparse networks the connectivity and routing fidelity are low if no active node is present in a grid. GAF is a location-based algorithm. It depends heavily on the availability of global location information to form virtual grids and associate nodes with the grids. Although the information provided by the global location information is highly accurate, the use of the global location information places a burden on the networks that have limited resources. The network lifetime also increases proportionally with the node density. The network lifetime extension is more significant in a dense network because of the large amount of energy saving achieved by powering down many redundant nodes that participate in routing. In contrast, the network lifetime saving may not be apparent in a sparse network. GAF promises small communication overhead. Each node broadcasts only one message during the discovery and active states. The message overhead of GAF is $O(V)$, where V represents the number of neighbors of each node.

STEM exploits the node redundancy to gain energy saving in the network. Similarly to GAF, the energy saving achieved by a dense network is much higher than the energy saving achieved by a sparse network. This is contributed to by the fact that the dense network has more redundant nodes that can be rigorously switched to sleep mode. STEM assumes that nodes in the network mostly reside in the monitoring state and they have infrequent data forwarding activities. This assumption implies that STEM is an application-specific algorithm, thus STEM is energy-efficient for sensor nodes that have an occasional data transfer which is triggered by an event. STEM requires nodes to turn on the radios of the receiver nodes and the subsequent nodes along the communication paths prior to data transfer. These nodes must also wait for an acknowledgment from the receiver nodes. Hence, there is a probability of nodes experiencing delay which could lead to a data latency issue. The advantage of STEM is that nodes rely on local information to decide their wake-up time. The decision to switch to sleep mode is also made locally whereby nodes immediately turn off their radios after transferring data. A significant energy saving is also gained when nodes spend the majority of their time in this mode. But a regular periodic switching between the sleep and active states to listen for incoming packets typically consumes a significant amount of energy. In STEM, the energy consumption associated with this switching process is not specified. In addition, the time, space and message overheads for setting up and transferring data are not mentioned. The connectivity of STEM is defined by the average number of neighbors M which is given by

$$M = \frac{NR^2\pi}{L^2}. \tag{2.4.1}$$

where N is the total number of nodes in the network, R is the transmission range of nodes and L is the length of the square field. The strong connectivity is possible for a dense network deployed in a small region.

ASCENT takes advantage of the redundancy of node density to prolong network lifetime like GAF and STEM. This means that the energy saving is significant when nodes are densely deployed in the network. The advantages of ASCENT lie in its flexibility and adaptive mechanisms that allow parameters to be tuned to cater for the requirements of applications. However, setting the parameters to accurately reflect the requirements of applications is not a simple task, which can make ASCENT impractical. The parameters involved are the neighbor threshold (NT) value, the loss threshold (LT) value, the sleep timer T_t value, the passive timer T_p value and the sleep timer T_s value. The (NT) value can be adjusted to optimize the network connectivity. By setting the (NT) to a much lower value, the average degree will be lower thus resulting in low connectivity in the network. The low connectivity can partition the network when the energy of active nodes is depleted. The (LT) parameter specifies the maximum amount of data losses that a network can withstand. When the data loss rate exceeds the (LT) value, nodes in the passive state will be switched to the active state to participate in data forwarding. The decision on the (LT) value is made based on the application of the network. For example, networks that are highly mobile tend to experience high data losses whereas networks that are used for environmental monitoring such as a bushfire event are likely to have low data losses.

The T_t, T_p and T_s values define the amount of time a node stays in the test, passive and sleep states, respectively. The choices of T_t, T_p and T_s values trade-off the energy saving with the decision quality. For example, setting the T_p to a larger value means that nodes will have sufficient time to collect data losses information from their neighbors enabling them to make further decisions. This contributes to the high accuracy of ASCENT but at the expense of high energy consumption associated with turning on more nodes' radios. When T_s is set larger, the energy saving is improved but the number of nodes in the passive state drops. These nodes are used to back-up the active nodes and are required to join them when the connectivity of the network is low. Consider the case when the connectivity of the network is very low due to the depletion of energy in active nodes and there is no node in the passive state. The network may experience partitioning if the active nodes are the only links connecting to other nodes. The time, message and space complexity of ASCENT are not provided.

Clustering Approach The discussion of the three clustering algorithms is provided in this section.

PACDS uses a simple marking process to calculate a CDS. This process provides a quick and simple way to build the network backbone. PACDS requires one round of message exchanges for the marking process and one more round for the pruning process. Thus, PACDS can be completed in a constant number of rounds. Wu et al. [WGS01] claimed that the time complexity to calculate the CDS is $O(\Delta^2)$, where Δ is the maximum node degree in the graph. The message complexity is given by $O(n\Delta)$, where n is the total number of vertices or nodes in the graph [WGS01]. However, these claims are refuted in [WAF04], in which the time and message complexity of PACDS may be higher. According to [WAF04], in the pruning process, a node u may need to examine as many as $O(\Delta^2)$ pairs of neighbors. Also, for each pair of neighbors, as much as $O(\Delta)$ time may be taken to find out whether such a pair of neighbors together dominates all other neighbors of u. Hence, the time complexity of PACDS could possibly be as high as $O(n^3)$. The estimated message complexity of PACDS is actually $O(m)$ where m is the number of edges in the unit-disk graph, as each edge contributes to two messages in the first stage [WAF04]. The number of edges m can be as many as $O(n^2)$. Therefore, the accurate message complexity of PACDS is $O(n^2)$, instead of $O(n\Delta)$. Due to the disagreement in the time and message complexities of PACDS in the literature, Table 2.1 specifies the range of these complexities. In PACDS, each time a topology changes a backbone is reconstructed to update the changes. Frequent topology changes triggered by highly mobile nodes can waste the energy resources in the network. Therefore, PACDS is only appropriate for static and low mobility networks.

ECDS employs local information to achieve a desired global property. The simulation result in ECDS shows that nodes in ECDS can survive longer than in PACDS [WGS01]. It is proven in [YJY06] that the message complexity of the ECDS algorithm is $O(n)$ because in the worst condition each node sends out one message during each phase. The time complexity of ECDS is also $O(n)$, which is estimated from the construction of the MIS. But the message complexity in ECDS is possibly higher than reported in [YJY06]. This is due to the frequent exchange of messages during the search of MIS and connector nodes. Therefore, the ECDS algorithm is appropriate for static networks. In the second phase, the decision to choose a connector node is made according to the weight metric. Hence, MIS nodes need to compare the weight of their neighbors and consult with other nodes before appointing a connector. This process may impose a high message overhead. The connectivity of the network is guaranteed as long as the CDS remained connected. The correctness of ECDS in constructing a CDS has been proven in [YJY06].

TMPO offers several advantages. First, TMPO performs a local computation on the minimal dominating set (MDS) based on two-hop neighbors information. Second, it uses a priority parameter that considers the node movement and energy level. As a result, the node with a higher energy level and low mobility has a higher chance to become a clusterhead. The priority parameter is used to achieve a stable topology. An unstable network that requires frequent topology constructions is computationally expensive. Different mobility environments have been used to test the stability of the topology constructed by TMPO. Therefore, TMPO is suitable for low and high mobility networks. Third, the clusterhead election is also rotated after some period of time to distribute the load fairly. The performance analysis of TMPO shows that it has better load balancing capability and higher topology maintenance stability against other heuristics. The willingness value in TMPO is used to control the network connectivity. The clusterheads that are highly mobile are likely to be disconnected from their neighbors. In order to avoid the network partitioning, TMPO adjusts the willingness value to a much lower value to remove the clusterheads from the dominating sets. The disadvantage of TMPO is the difficulty in managing the hierarchical structure of the network. TMPO requires at most 3-hop away neighbor information to find doorway and gateway nodes. Any changes in the role of clusterhead, host, doorway or gateway will require nodes to propagate changes to their neighbors, resulting in a delay for updating the topology changes.

Hybrid Approach The discussion of the three hybrid algorithms is provided in this section.

SPAN combines the clustering approach with the power mode approach to allow idle non-CDS nodes to switch to sleep mode, thus conserving energy consumption and simplifying the switching mode operation. The sleeping nodes are also able to receive packets because SPAN runs on top of 802.11 ad hoc power saving mode. Thus, SPAN minimizes packet losses and packet retransmissions. However, the power saving mode feature can limit SPAN's ability to save energy if nodes frequently switch from sleep mode to active mode to listen for traffic advertisements. The message complexity of SPAN is $O(n)$ since each node exchanges one message during the coordinator announcement or coordinator withdrawal [LW09]. The time complexity of SPAN is $O(n)$ because SPAN needs to consider n total number of nodes in the network for constructing the CDS backbone [LW09].

SPAN has to piggyback HELLO information onto the broadcast updates, thus it is dependent on the routing protocol. SPAN is practical because it does not require any location information system to determine the position of nodes. In SPAN, the location information is provided by the GOD module of ns through the exchange of HELLO messages. However, the location given by the GOD module is less accurate than the location given by a location information system. In SPAN, each node needs to keep a maximum of 3-hop neighborhood information for coordinator announcement and withdrawal processes. Maintaining and updating the 3-hop information may impose a significant message overhead. The connectivity of SPAN is low since it is governed by the rules used during the coordinator announcement and withdrawal.

CLUSTERPOW provides an implicit and adaptive clustering approach. It is not tightly coupled to a specific routing protocol and therefore it can be used with any routing protocol. Since CLUSTERPOW is an extension of COMPOW, it can be used in a homogeneous network by setting the common power to a minimum value. CLUSTERPOW does not have any leader or gateway and the clusters are automatically generated when the power level is chosen. This attribute simplifies the cluster formation process as nodes do not need to elect clusterheads or gateways. In this way, the energy resources consumed for electing the clusterhead nodes and building the clusters can be saved. The architecture of CLUSTERPOW is similar to COMPOW hence, its message overhead is determined by the number of power levels used in the network. The CLUSTERPOW's overhead is $O(Pn)$, where P is the number of power levels used by nodes while n is the total number of nodes in the network. CLUSTERPOW depends on global information in deciding on the minimum power level needed for routing because each node has to consult the master routing table. The master routing table is formed by taking an input from different routing tables. The message overhead for building and maintaining the multiple power routing levels is significantly high. The advantage of CLUSTERPOW is that its' design has been tested on CISCO wireless card and its correctness has been verified. Even though Kawadia et al. [KK03] experienced a technical problem when changing the transmit power level, they managed to test it on laptops. Thus, the practical implementation of CLUSTERPOW is guaranteed.

LEACH is energy-efficient and offers network lifetime extension. It is a localized algorithm that allows each node to gather information from its neighbors to form clusters. Each node sends out one message during cluster set-up, thus the message and time complexity of LEACH are low. The message and time complexity of LEACH are $O(n)$, where n is the total number of nodes in the network [LW09]. Unlike other clustering or hybrid approaches, LEACH uses data aggregation to compress the size of messages before sending them to the base station. It reduces the energy involved in transmitting a large amount of data over long distance. However, there are several drawbacks of LEACH. First, LEACH executes many tasks. Therefore, the operation of LEACH is quite complicated. Each clusterhead node is assigned for data aggregation, TDMA scheduling and data forwarding tasks. These demanding tasks can drain the energy of the clusterhead nodes and shorten the lifetime of the nodes. Second, the clusterhead nodes use long distance transmission to send data directly to the base station. In other words, LEACH has a scalability issue in which it did not exploit the multihop communication between two clusters that can contribute to energy saving and network scalability. The scalability issue of LEACH has been addressed in HEED [YF04], in which HEED proposed inter-cluster routing between clusterhead nodes to support multihop communication with the base station. Third, LEACH assumes that all nodes have data to transmit whereas in practice this assumption maybe untrue, hence energy is wasted. Fourth, the criteria adopted for the clusterhead selection do not account for the remaining energy capacity at each node. As a result, it is possible that the node with a lower energy level is selected as a clusterhead node. Thus, it can shorten the network lifetime as it does not have sufficient energy to perform various tasks assigned to it. Finally, even though the use of TDMA-based schedule can avoid multiple retransmissions, it is not easy to synchronize nodes.

Based on the discussion of the twelve topology control algorithms, their features in terms of advantages and disadvantages are highlighted and summarized in Table 2.2.

From the discussion in Section 2.4.1, the energy conservation strategies adopted by the algorithms are highlighted and they are listed in Table 2.3.

2.4.2 Evaluation based on the Network Lifetime Definition

Most topology control algorithms discussed in Section 2.4.1 strive to achieve a common objective to increase the lifetime of WSNs. As mentioned in Section 2.2.3, different definitions were typically used by these algorithms to estimate the network lifetime. This section presents the definitions adopted and evaluates their impact on the topology control performance. Their merits and drawbacks are given.

Table 2.4 summarizes the network lifetime definitions of all algorithms. It can be seen that MECN, COMPOW, STEM, ASCENT, TMPO and CLUSTERPOW do not have a network lifetime definition. Even though the definition of the network lifetime is application-specific, majority of the algorithms in Table 2.4 do not specify a possible application for WSNs, apart from STEM.

SMECN defines the network lifetime as the number of nodes remaining alive over

Algorithm	Highlights of the Algorithms	
	Advantage(s)	Disadvantage(s)
Power Adjustment Algorithms		
MECN	Strong connectivity	Needs a GPS to build topology
SMECN	Strong connectivity. More power and time efficient than MECN	Needs a GPS to build topology
COMPOW	Practical-based topology control. Built on a wireless testbed	High message overhead for computing multiple power levels
Power Mode Algorithms		
GAF	Low communication overhead	Relies on GPS to compute the grid and allocate nodes into the grid
STEM	Energy-efficient for event-triggered applications	Trade-off energy savings with setup latency
ASCENT	Self-reconfigurable and adaptive to react to applications' dynamic events	Possibly fast energy depletion among active nodes due to uneven load distribution
Clustering Algorithms		
PACDS	Simple and quick to calculate the CDS and location service-free	Not feasible for mobile network
ECDS	Energy remained at nodes is considered for CDS construction	High message overhead
TMPO	Stable topology and load balancing features. Appropriate for high mobility networks	High message overhead and computationally intensive
Hybrid Algorithms		
SPAN	Location service-free and exploits the advantage of power saving 802.11 for routing	Nodes have to periodically wake-up and listen for traffic advertisements
CLUSTERPOW	Easy maintenance of clusters and practical implementation on a wireless card	Significant message overhead for computing multiple power levels
LEACH	Offers various energy-efficient mechanisms	Complicated tasks performed by clusterheads and not scalable

Table 2.2: A highlight of the features of various distributed topology control algorithms.

Algorithm	Energy conservation strategies				
	Use minimum energy path	Control the transmit power	Alternate to sleep mode	Apply load balancing	Aggregate data
Power Adjustment Algorithms					
MECN	Yes	Yes	Possible	No	No
SMECN	Yes	Yes	Possible	No	No
COMPOW	No	Yes	No	Yes	No
Power Mode Algorithms					
STEM	No	No	Yes	No	No
ASCENT	No	No	Yes	No	No
GAF	No	No	Yes	Yes	No
Clustering Algorithms					
PACDS	No	No	No	Yes	No
ECDS	No	No	No	Yes	No
TMPO	No	No	Possible	Yes	No
Hybrid Algorithms					
SPAN	No	No	Yes	Yes	No
CLUSTERPOW	No	Yes	No	No	No
LEACH	No	No	Yes	Yes	Yes

Table 2.3: The strategies employed for energy conservation in topology control.

some duration of time. A simulation was conducted to compare the network lifetime performance of SMECN over MECN. The result shows that SMECN has more alive nodes than in MECN. However, the lifetime measured by this definition alone is not accurate because it cannot represent the criticality of nodes. Since SMECN chooses the path that has the minimum energy routes, each node tends to send messages to the base station via the same route through its closest neighbors. The nodes along this route are the critical ones that can experience faster energy depletion and early death.

Another important metric to be considered in the network lifetime definition is the connectivity of the network to the base station. In certain applications such as data monitoring, the failure of transmitting data to the base station is used to describe the end of the network lifetime though the number of alive nodes is significantly high. In SMECN, the connectivity to the base station is defined by the number of alive nodes which remain connected to the base station. SMECN

Algorithm	Network lifetime definition	Target application
Power Adjustment Algorithms		
MECN	None	No
SMECN	1. Number of alive nodes 2. Number of alive nodes connected to the base station	No
COMPOW	None	No
Power Mode Algorithms		
STEM	None	Activity monitoring
ASCENT	None	No
GAF	1. Fraction of alive nodes 2. Time the packet delivery ratio drops	No
Clustering Algorithms		
PACDS	The time the first node dies	No
ECDS	The time the network survives	No
TMPO	None	No
Hybrid Algorithms		
SPAN	The fraction of remaining CDS nodes	No
CLUSTERPOW	None	No
LEACH	The number of alive nodes	No

Table 2.4: The definitions used by topology control algorithms to measure the network lifetime of WSNs.

uses this definition to describe the ability of the network to communicate with the base station. Incorporating this definition with the number of alive nodes provides more accurate definition of the network lifetime. For example, consider a scenario in which the number of alive nodes is low but the network can still provide a useful task of transmitting data to the base station. In this case, the network should be considered alive if both network lifetime definitions are used.

GAF uses two metrics to define the network lifetime. The first metric measures the network lifetime as the fraction of survived nodes as a function of time. The network lifetime metric is used to analyze the performance of GAF under both low mobility (1 m/s) and high mobility (20 m/s) patterns. GAF considers a range of pause times to indicate node movements. The shorter pause time represents moving nodes whereas the longer pause time represents no node movement.

GAF shows that nodes with shorter pause times result in better network lifetimes than nodes with longer pause times. This is because the moving nodes possibly move into grids with fewer nodes allowing them to share the load with other nodes.

The second metric employed to define the network lifetime is the time t until the monitored packet delivery ratio drops dramatically. This definition measures how long the network can successfully deliver packets until the ratio drops below a certain threshold value. It measures the ability of GAF to connect to the base station.

GAF uses both metrics to define the network lifetime because the use of either the first or second metric alone is insufficiently accurate to represent the network lifetime. Consider the case of using the fraction of alive nodes to define the network lifetime. Since the network lifetime of GAF is closely related to the density of the network, under a dense network the small fraction of alive nodes can deliver the traffic without affecting routing fidelity, whereas in a sparse network the routing might be disrupted. The second metric on the other hand requires one to define an appropriate packet delivery ratio. However, the method to set the ratio and identifying the best packet delivery ratio are not mentioned in GAF.

PACDS measures the network lifetime at the time when the first node in the network dies when it runs out of energy. This definition does not provide the connectivity information to the base station. If the first failure node is the backbone node then there is a possibility that the backbone has to be reconstructed. If the first failure node is the node outside the backbone, then the network can still operate. This definition might be appropriate for a network consisting of nodes that are equally important and failure of one node is unacceptable.

ECDS defines the network lifetime as the number of periods that the network can survive until it can no longer construct a CDS. In other words, the network fails when it can no longer construct a backbone. The network lifetime definition used by ECDS can describe the successful delivery of messages to the base station as long as the backbone exists. But under harsh environments whereby nodes often fail, this definition is inaccurate to describe the network lifetime. This is because ECDS may not be quick enough to respond to the dynamic changes in the environments. Moreover, a frequent re-computation of the backbone can consume a significant amount of energy which can later drain the energy of the backbone nodes.

The network lifetime in SPAN is defined by the fraction of the CDS nodes that remained alive. But the appropriate figure detailing the number of alive nodes that must remain alive to support the operation of the network is not provided. This definition might be reasonable if the fraction of the alive CDS nodes can provide the connectivity to the base station. In a particular application such as routing, the dominating set constructed in SPAN must remained connected. Therefore, the fraction of the CDS nodes that remained alive must be connected otherwise the routing is disrupted.

In LEACH, the number of sensors still alive at a predefined time is used to measure the network lifetime. LEACH also measures the number of rounds from the first node until the last node dies. Each node is assigned with a certain energy threshold. The system lifetime of LEACH is shown to be higher than other algorithms regardless of the energy thresholds assigned to the node. Even though LEACH can prolong the network lifetime, it is not known whether the remaining nodes in the network can form a backbone or not. If the backbone cannot be constructed, data cannot be transmitted to the base station. Nevertheless, LEACH shows that nodes die in a random fashion which indicates a fair load distribution in the network. This characteristic is desirable for monitoring applications in which the network can cover every region of interest, as each region has at least one node to monitor the area.

2.5 Conclusions

This chapter has presented various topology control approaches to achieve energy efficiency in WSNs: Power adjustment algorithms aim to conserve energy in nodes by controlling the transmission range; Power mode algorithms schedule the idle nodes so that they can be put into sleep mode to save energy; Clustering algorithms select sets of nodes to form backbone to reduce communication overheads in the network; Hybrid algorithms combine the clustering approaches with either the power mode or power control approaches to create a better energy solution to topology control.

Among the four categories of topology control techniques presented in this chapter, clustering offers many great features that are not available in other techniques. Even though hybrid techniques inherit the benefits of the clustering techniques, they can be difficult to implement such as in [HCB00] and the implementation complexity may outweigh their benefits.

Under the clustering category, special attention is drawn to CDS approaches for their simplicity and efficiency in dealing with topology control. The advantages offered by the topology control through the CDS approach are:

- It can avoid the duplicate packet transmission problem, which cannot be prevented in either the power adjustment nor the power mode techniques [LW09].
- It is useful for addressing scalability issues and simplifying network organization.

- It offers many possible energy management strategies for WSNs. These include uniform energy distribution across the network leading to network lifetime improvement, data aggregation and compression to minimize the transmission energy and node scheduling activities.

- It can support homogeneous or heterogeneous networks, thus providing flexibility to trade off the energy-efficiency over the lower hardware cost if required [MR04].

- It guarantees network connectivity. As long as nodes have at least one clusterhead neighbor or they themselves are a clusterhead, the network remains connected.

2.5.1 Identified Research Problems

Although many research efforts have been devoted to topology control problems, there are a number of problems lacking in the existing techniques.

The first problem refers to the failure of the techniques to account for wireless communication effects contributed by the physical radio layer. The common assumption made is if the links exist between two nodes, they are able to receive and send packets without any loss. However, in a realistic environment, these links can become unidirectional or unstable as they are influenced by many factors, including the communication distance between nodes, the interference level at the receiver and the signal strength of the sender. As a result of this simplified assumption, the performance of the techniques can vary widely between different simulators and they may not work under real world implementations. These claims are supported in [TMB01, HCG09], which demonstrate the significant impact of physical layer on the behavior and the performance of protocols. Another issue is related to inaccurate radio model adopted by the topology control techniques. For example, to calculate the minimum energy path in a network, the technique in [RM99] ignores the energy dissipation at receivers and considers the energy consumption of the transmitter only [HCB00]. Hence, the topology constructed may not be energy-efficient if different radio characteristics are employed. Another simplistic assumption of the radio model is the sensor nodes in networks have homogeneous radios (equal radio range) [KNG+04]. This assumption is invalid in applications where all nodes have non-identical radios or antennas. In addition, the radio coverage is non-circular in reality. The techniques using a simple model based on a unit disk graph (UDG) often simplify the radio coverage as in a circle. This implies that they are not able to capture the realistic behavior of the wireless links, in which the link quality constantly fluctuates, thus affecting the link connectivity. Even though, the transmission distance is an important factor for successful communication between two nodes, there are other nontrivial factors reported in [KNG+04]. They include the type of radios used, the angle between sender and receiver antenna and the existence of obstacles such as tall building or terrains.

The second problem is the absence of important performance measures for evaluating the quality of the techniques. For instance, techniques designed for energy conservation seldom present the message overhead involved in constructing the network. In general, the more messages exchanged with nodes, the higher the energy consumption in the network. In the case of dense networks, the message and energy costs could be higher. If prolonged network lifetime is the main concern then the techniques with a low message overhead are desirable. A further example is the lack of node density consideration when analyzing the performance of the techniques. To illustrate this case, let's consider the performance analysis of the work in [YJY06] with respect to the CDS size. It claims to be superior than the other two chosen techniques; nevertheless, this claim only holds true when the node density is high. Thus, the performance results of the techniques may be incorrect.

2.5.2 Research Direction

It can be concluded that many topology control techniques adopt a simplified model of the physical radio layer for simulations and they require significant communication overheads, which shorten the network lifetime. Failure to address the realistic behavior of wireless networks can affect performance and correctness of these techniques, particularly when the sensor nodes are deployed in applications.

Therefore, there is a need to develop topology control techniques which are capable of handling more realistic scenarios and consume low energy consumption for achieving network lifetime extension. Motivated by the many merits and interests of the research community in the CDS, the proposed solutions adopt the connected dominating sets (CDS) concept for constructing a topology. Despite the extensive research done in the subject of the CDS, the existing solutions are often not suitable for practical implementations. This thesis presents three novel topology control algorithms that are modeled using a realistic wireless communication framework. Other issues addressed include adopting a more accurate radio model, and integrating important evaluation metrics that are missing in the existing techniques.

Chapter 3

Distributed Single Initiator CDS Algorithms

3.1 Introduction

As concluded in Chapter 2, the CDS approach offers significant potential in the synthesis of efficient network topologies. Thus, two novel "single initiator" algorithms that rely on CDS construction are proposed in this chapter: (i) Three-phase single initiator (TPSI) and ii) Single-phase single initiator (SPSI).

TPSI has several contributions. First, it produces a minimal CDS size and constructs the CDS with low communication and computational costs. This is important to conserve the limited energy and computational resources in the network. Although several CDS algorithms [MGG10, WLD06, WL99, YJY06] have been proposed to minimize the CDS, they use an extensive exchanged messages when forming the CDS, thus resulted in high message overhead and energy consumption in the network. Second, TPSI is a distributed algorithm that uses localized information to build the CDS. Previously studied algorithms that use a centralized approach and rely on non-localized information as in [GK98, LTW+05, MDJ+06] are not practical for WSNs since the cost for gathering and updating this information is expensive. Moreover, WSNs have no fixed topology prior to deployment and this topology can quickly become outdated as nodes move or die. Third, TPSI takes into account the quality of the radio links when searching for neighbors and does not assume that the radios of sensor nodes to be equal. Whereas, previously studied algorithms [MGG10, WLD06, WL99, YJY06] simplify these two assumptions, thus they may not work well in practice than they do in simulation. However, TPSI trades off the CDS size with the convergence time. The convergence time here is defined as the time taken to build the CDS. As the network becomes denser, the CDS size becomes smaller but the convergence time becomes larger. The large convergence time is due to the three phases involved in TPSI.

SPSI on the other hand converges rapidly and generates a CDS with low communication and computational costs. In contrast to TPSI, SPSI uses a single phase to generate a CDS in order to converge fast. It is also a distributed and localized algorithm that eliminates the unrealistic network model adopted in the previously studied algorithms. Unlike in TPSI which is efficient in minimizing the CDS size of dense networks, SPSI generates a slightly larger CDS size than the one in TPSI.

TPSI and SPSI algorithms improve the existing CDS approaches [BGLA03, GK98, LTW+05, MGG10, MDJ+06, SDB98, WL99, YJY06] in the following respect:

- They are simple and practical to implement. TPSI and SPSI do not require unrealistic assumptions to operate. They address the issue of asymmetrical wireless links and can cope with non-uniform transmission ranges of nodes.
- They are distributed algorithms that use localized information for constructing the CDS. This ensures that the communication overhead in gathering the information remains low to account for dynamic nature of WSNs, where

their topology often change when nodes move or die.

- They construct a small-size CDS to simplify the topology maintenance process and keep the message overhead low.

- They reduce the number of messages sent to find a CDS, thus minimizing the energy spent on communication activity. Instead of relying on negotiation process to determine CDS nodes, where nodes frequently exchange many messages to decide their eligibility, nodes in TPSI and SPSI perform a local check based on the neighbor information to find out whether they have been nominated as a CDS.

- They minimize the energy consumption in the network to prolong the network lifetime. Various network parameters are introduced to achieve this.

- They can distribute the loads in the network fairly among nodes in the network based on the energy capacity of nodes. The role of the clusterheads can be rotated among nodes with high energy level to avoid the low energy nodes from being depleted.

This Chapter begins with Section 3.2 detailing the assumptions made by the algorithms in order to work. Section 3.3 provides the implementation details of the TPSI algorithm and Section 3.4 describes the proposed SPSI algorithm.

3.2 WSN Model and Assumptions

A WSN is represented by an *undirected graph* $G = (V, E)$, where V is a set of sensor nodes in the network, called vertices and E is a set of a communication link between a pair of sensor nodes, called edges usually denoted as $(u, v) \in E$. Two vertices u and v are neighbors if (1) they are within their maximum transmission range $R_{\max}$ and (2) the communication links between them are symmetrical. Figure 3.1 illustrates the example of the graph used. Node v_5 in Figure 3.1 is within the communication range of node u and it can hear node u. However, in the presence of interference, node u cannot hear node v_5, thus creating an asymmetrical link. Due to the failure of meeting the condition (2), node v_5 is not a neighbor of node u.

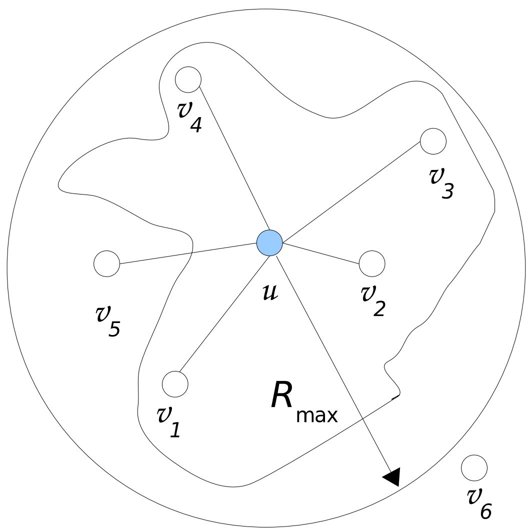


Figure 3.1: Network graph used by the proposed algorithms. Nodes v_1, v_2, v_3 and v_4 are node u's neighbors while nodes v_5 and v_6 are not the neighbors of node u. Node v_6 is not in u's transmission range. Although, node u and node v_5 are located within their transmission range, node u cannot hear node v_5.

The following assumptions are made in the design of these algorithms:

- Each node has a unique identifier (ID)
- Each node has identical energy capacity.
- Each node maintains a neighbor table to store the neighbor list information required for computing the CDS. The neighbor table is frequently updated to reflect the changes in the network topology.

- Nodes in the network do not have a similar radio capability. Thus, it possible that the $R_{\max}$ of node u is different from the $R_{\max}$ of node v.
- TPSI algorithm assumes each node is able to estimate the distance between two nodes.
- Nodes perform a basic functionality of gathering temperature readings from an event of interest. The tasks these nodes are dealing with are therefore simple.

3.3 Three-phase Single Initiator (TPSI) Algorithm

3.3.1 Overview

The three-phase single initiator (TPSI) algorithm is based on the concept of generating a maximal independent set (MIS). The aim of the TPSI algorithm is to construct a CDS that is practical, energy-efficient and small-size. The graph theory definitions for MIS, DS and CDS are referred to in Appendix A.

It involves three phases to construct the CDS. The first phase is the generation of a MIS in the graph G, in which the MIS is a dominating set (DS). The second phase is the generation of a CDS, in which connectors are selected to connect the MIS. The third phase is the pruning of redundant nodes in the CDS to further reduce the size of the CDS. Algorithm 2 summarizes the three phases involved in the TPSI algorithm.

The node can be in one of the four states. The four states are uncovered, dominator, dominatee and CDS. All nodes initially stay in uncovered state and at the end of the CDS construction, they will subsequently become either a dominatee or CDS. Figure 3.2 shows the state transition diagram of a node during the

Algorithm 2 The implementation of TPSI algorithm

Require: Graph $G(V, E)$
Ensure: Connected Dominating Set $(CDS) \in G(V, E)$

```
procedure TPSI(G(V, E))                      ▷ Main procedure
    GenerateMIS(G(V, E))                     ▷ Phase 1 of TPSI
    GenerateCDS(G(V, E))                     ▷ Phase 2 of TPSI
    EliminateCDS(G(V, E))                    ▷ Phase 3 of TPSI
end procedure
```

construction of the CDS. Before the CDS construction process starts, each node

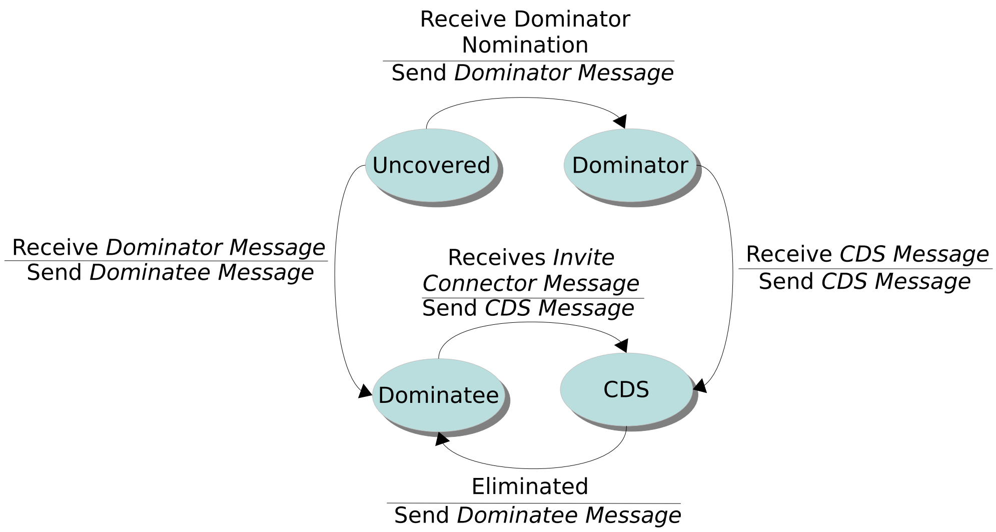


Figure 3.2: The state transition diagram of a node in the TPSI algorithm.

executes neighbor discovery process to obtain neighbor information required for the construction. The neighbor discovery process is described in Section 3.3.2 while the detailed processes of the three phases are explained in Sections 3.3.4, 3.3.5 and 3.3.6.

3.3.2 Neighbor Discovery

At the beginning, every node periodically broadcasts a beacon to discover neighbors and gather its neighbor information. This information is required by nodes to construct the CDS. The TPSI algorithm assumes symmetrical links exist in the network in order to construct the CDS. However, in real life scenarios the links can become asymmetrical in the presence of signal interference or when the transmission ranges decrease as the battery of nodes becomes weak. To cope with the asymmetrical links, a symmetrical neighbor discovery and maintenance mechanism as is presented in Appendix B is adopted to selectively identify nodes with symmetrical links as neighbors and maintain these neighbors for communication.

The TPSI algorithm uses the beacon format shown in Figure 3.3. This beacon is implemented by TPSI to provide the information required for computing the CDS. The information in the beacon varies between states. The dominatee has

Uncovered	Node ID	State	Symmetrical Neighbors	Broken Neighbors	Residual Energy	Distance	
Dominator	Node ID	State	Symmetrical Neighbors	Broken Neighbors	Residual Energy	Distance	Connector
Dominatee	Node ID	State	Symmetrical Neighbors	Broken Neighbors	Residual Energy	Distance	MIS
CDS	Node ID	State	Symmetrical Neighbors	Broken Neighbors	Residual Energy	Distance	

Figure 3.3: The beacon format of a node in the TPSI algorithm.

an additional field called MIS, which includes the list of chosen dominators required for the MIS construction. The dominator also includes one extra field for the connector list to announce the nomination of the CDS nodes for the CDS construction.

The content of the beacon format is described below:

- Node ID: represents the unique node identifier.
- State: identifies the state of the node.

- Symmetrical Neighbors: lists 1-hop neighbors with symmetrical links.
- Broken Neighbors: lists 1-hop neighbors with broken (asymmetrical) links.
- Residual Energy: specifies the remaining energy capacity of node.
- Distance: provides the distance to each neighbor.
- MIS: lists of neighbors that are nominated as dominators.
- Connector: lists of neighbors that are nominated as CDSs.

The cost of bearing this extra information is later shown in Chapter 6 to be marginal.

3.3.3 Selection Metric

A selection metric is used for finding eligible nodes to be included in the CDS. The metric can incorporate various network parameters to elect the CDS. Examples are the node degree, node identifier, transmission power, node mobility and power capacity. The node degree refers to the number of neighbors of nodes. The choice of the parameters can impact the performance of the CDS and the life of the network. For example, the use of node degree as a selection parameter is efficient to reduce the size of the CDS [GT95, WL99].

TPSI algorithm proposed a selection metric based on a *key* that takes into account the energy capacity of nodes, the number of neighbors and the energy consumption of nodes. The objectives of the *key* are to ensure: (1) nodes deplete their energy in an even and fair manner to extend the network lifetime, (2) the network forms a small CDS to reduce the communication overhead in the network, in which a node that has the largest number of neighbors is selected as a CDS and (3) the CDS consists of a set of nodes with high power reserves. If a sensor node with low energy is chosen as a CDS, this can lead to network partitioning as it may not have enough power to perform various tasks in the network such as relaying packets to other nodes and aggregating data.

The *key* is defined as follows:

Definition 1 (Key). *Node u with key(u) has a higher priority than node v with key(v) if key(u) is larger than key(v). In case of a tie, the node with the larger identifier is used to break the tie.*

The *key* of node u is computed using

$$key(u) = fairness(u) + nodeDegree(u), \tag{3.3.1}$$

The *fairness* is determined by

$$fairness(u) = \left(\frac{d(u,v)n}{d(u,v)^n + n - 1}\right)\left(\frac{E_r(u)}{E_i(u)}\right), \tag{3.3.2}$$

where *nodeDegree*(u) is the number of neighbors of node u, *fairness* is the fair metric adopted from [DMSL11], $d(u,v)$ is the distance between a pair of nodes u and v, $E_r(u)$ is the residual energy, $E_i(u)$ is the initial energy and n is the path loss exponent of value 3.

The initial energy is the energy capacity of node before the CDS construction starts, in which each node is assumed to have an identical energy supply. The residual energy represents the remaining energy of node at time t. It is computed by deducting the energy used for the transmit and received activities from the initial energy at time t. The residual energy is updated in the beacon format of node and sent to neighbors periodically. The *nodeDegree* ensures that nodes with the largest number of neighbors, which can cover more nodes are chosen as a CDS, hence creating a small CDS.

The *fairness* provides an optimized approach for choosing energy-efficient links and achieving uniform energy consumption throughout the network. It takes into account the energy consumption involved during multihop communication as well as the energy reserves of nodes when choosing eligible nodes for the CDS. Nodes with high energy reserves are selected as CDS nodes. The method employed for determining the energy-efficient nodes is simple. It does not involve complex computation as in [RM99] and requires only one-hop neighbor information. A node will select a neighbor as a relay node if: (1) The energy consumption to communicate to the neighbor is minimum and (2) The neighbor has higher energy reserves. The theoretical proofs of the *fairness* can be found in [DMSL11].

3.3.4 MIS Generation

The goal of the first phase is to generate MIS. The MIS generation begins with the initiator election. The simplest and low cost election method is to use node identifier. TPSI algorithm selects the smallest node identifier with node id 0 to be an initiator. Alternatively, a leader election algorithm in [Awe87] can be used to find the initiator but at the cost of higher message overhead. The MIS generation will only start after the neighbor information has been gathered, which is decided upon the expiry of a timer. In order to find eligible nodes for the MIS, a selection *key* defined in Equation 3.3.1 is used.

The process of generating the MIS can be described as follows:

- At the beginning, all nodes stay in an uncovered state.
- Each node broadcasts a beacon to discover its neighbors. This beacon will attach the information described in Figure 3.3.

- Each node then applies the neighbor discovery and maintenance mechanism described in Appendix B to identify and store symmetrical neighbors.
- The MIS construction starts with node 0 volunteering to be an initiator. It changes its state to dominator and broadcasts a *Dominator Message*.
- Upon receiving the *Dominator Message*, an uncovered node changes its state to a dominatee. It then selects the uncovered neighbor with the largest *key* to be a dominator. The dominator election is broadcast to its neighbor by sending a *Dominatee Message*.
- An uncovered node receiving the *Dominatee Message* will first check whether it has been nominated as a dominator. If it is chosen, it changes its state to a dominator and broadcasts a *Dominator Message*. Otherwise, it will remain in the uncovered state.
- The above process is repeated until all uncovered nodes change their state to either a dominatee or a dominator. The decision to terminate the MIS generation is made locally when a node changes its state to dominatee or dominator and no longer has uncovered nodes.

Algorithm 3 shows the detailed description of the MIS generation executed at all nodes.

3.3.5 CDS Generation

Recall that the MIS generated in the first phase is a DS. It can form a CDS by connecting the DS with a set of nodes called connectors. Hence, the second phase is interested in choosing these connectors to join the DS. The DS and connectors both form the CDS in the network. At the completion of the CDS process, all nodes will remain either in a CDS state or a dominatee state. The CDS generation

Algorithm 3 The MIS generation of the TPSI algorithm

Require: *G(V,E)*
Ensure: *MIS* $\in$ *G(V,E)*

```
procedure GenerateMIS(G(V, E))          ▷ Phase 1: Find a MIS in the network
    MIS ← ∅ ; DOMINATEE ← ∅
    POTENTIALMIS ← ∅; UNCOVERED ← ∅
    for all u ∈ V do
        UNCOVERED ← UNCOVERED ∪ {u}
        i ← node 0                                      ▷ Initiator Election
        MIS ← MIS ∪ {i}
    end for
    for all u ∈ V do
        for all  u ∈ N(i) do
            DOMINATEE ← DOMINATEE ∪ {u}
        end for
        for all u ∈ UNCOVERED do                        ▷ Uncovered State
            for all v ∈ N(u) do
                if v ∈ MIS then                 ▷ Receives Dominator Message
                    DOMINATEE ← DOMINATEE ∪ {u}
                end if
                if v ∈ DOMINATEE then
                    if u ∈ POTENTIALMIS then            ▷ Potential Dominator
                        MIS ← MIS ∪ {u}
                    end if
                end if
            end for
        end for
        for all u ∈ DOMINATEE do                        ▷ Dominatee State
            for all v ∈ N(u) do
                if v ∈ UNCOVERED then
                    Compute key(v)
                end if
            end for
            if key(v) is the largest then
                POTENTIALMIS ← POTENTIALMIS ∪ {v}       ▷ Select Dominator
            end if
        end for
    end for
end procedure
```

begins as soon the MIS generation process is completed. Due to the order of the message propagation, the leader will be the first to complete the MIS generation and to begin the CDS generation.

The process of generating the CDS can be described as follows:

- The CDS generation starts after the completion of the MIS generation. A node changes its state to CDS and broadcasts a *CDS Message*.
- A dominatee receiving the *CDS Message* sends a *Volunteer Connector Message* to its neighbors and waits for an invitation to become a CDS.
- A dominator receiving the *Volunteer Connector Message* will set its timer T_{dom} to wait for the arrival of *Volunteer Connector Message* from surrounding neighbors. The timer is adopted from [KYZS05] and computed by

  $$T_{\text{dom}} = T_{\max} \left(\frac{1}{\textit{nodeDegree(u)}} \times \frac{1}{E_r(u)} \right), \tag{3.3.3}$$

  where $T_{\max}$ is a pre-defined time value, *nodeDegree*(u) is the number of neighbors of node u and $E_r(u)$is the residual energy of node u.

  Equation 3.3.3 allows the qualified dominator to be the first to find a connector, thus suppresses the chances of the less qualified dominator to nominate a connector. The selection criteria for the qualified dominator are based on the residual energy and node degree. Based on Equation 3.3.3, it is apparent that the dominator with the largest residual energy and node degree has a shorter delay than its counterpart. Therefore, it is the first to send an invitation to a potential connector.

  During the time-out, if the dominator receives a *CDS Message*, it will change its state to a CDS and broadcasts a *CDS Message*. This indicates that a connector has been chosen by other dominators. Therefore, it has to be a CDS to connect to this connector.

Otherwise, when the timer is expired, it computes the *key* of each dominatee neighbor using the Equation 3.3.1 and nominates the node with the largest *key* to be a connector. The connector nomination is then broadcast using an *Invite Connector Message*.

- Upon the receipt of an *Invite Connector Message*, the dominatee first checks whether it has been nominated as a CDS. If it is nominated, it changes its state to a CDS and broadcasts a *CDS Message*. Otherwise, it stays in the dominatee state.
- The above process is repeated until all nodes change their state to either a CDS or a dominatee. The decision to terminate the CDS generation is also made using localized information. The decision to terminate the MIS generation is made when all nodes are either in CDS or dominatee state.

3.3.6 CDS Pruning

The CDS formed at the completion of the second phase is not a minimum dominating set. Since, computing the minimum dominating set is a well known NP-hard problem [GJ79], the CDS pruning aims to reduce the CDS size. After the completion of the second phase, it is observed that some nodes in the CDS are already covered by at least a CDS. Therefore, they are redundant and can be removed from the CDS to minimize the size of the CDS.

To describe the CDS pruning process, the term "pendant node" is introduced in Definition 2.

Definition 2 (Pendant Node). *A pendant node is a node that has exactly one neighbor in the network.*

The pruning of the CDS is made when a CDS node satisfies Rule 1.

Rule 1. *A pendant node u is pruned from a CDS if it is a CDS and has a CDS node among its one-hop neighbors.*

It is evident from Rule 1, the pruning of the pendant node u will guarantee the network connectivity because it has a neighbor in the CDS. Since this rule applies only to nodes in the CDS, the computational cost of this process is low. Moreover, no message exchange is required to run the pruning process as the information is already available in the neighbor table. Once eliminated from the CDS, the pendant node changes its state to dominatee and broadcasts a beacon to inform the change of its state.

Algorithm 4 shows the detailed description of the CDS generation and elimination phases of the TPSI algorithm executed at all nodes.

Figure 3.4 illustrates the process of constructing the CDS using the TPSI algorithm. The number next to nodes represents the node identifier. The CDS process can be explained by the following steps:

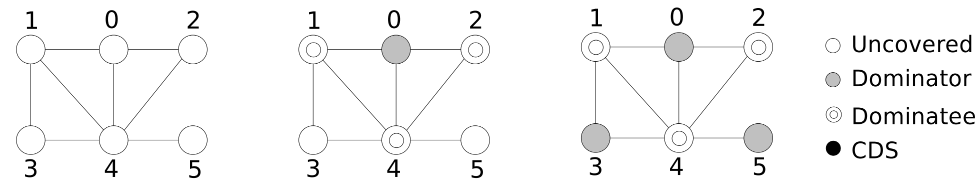
1. Figures 3.4(a), 3.4(b) and 3.4(c) demonstrate the MIS generation process.

    - In Figure 3.4(a), all nodes are initialized to uncovered state. The MIS generation begins with node 0 sending a *Dominator Message* to neighbors as illustrated in Figure 3.4(b).

    - Upon the receipt of the *Dominator Message*, nodes 1, 2 and 4 change their state to dominatee and select their uncovered neighbors with the largest *key* as a potential dominator. In this example, dominatee 1 and

Algorithm 4 CDS generation and CDS elimination of the TPSI algorithm

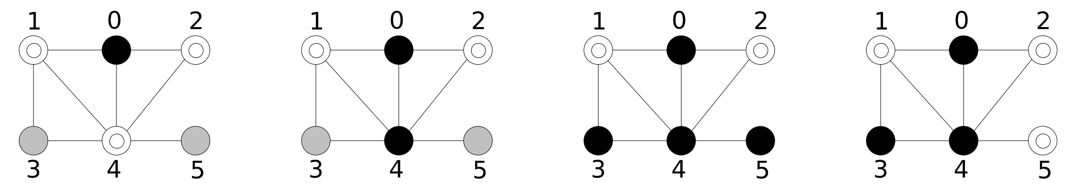
```
procedure GENERATECDS(G(V, E))  ▷ Phase 2: Find connectors to join the MIS
    CDS ← ∅ ; CONNECTOR ← ∅; VOLUNTEER ← ∅
    CDS ← CDS ∪ {i}                    ▷ Initiator invokes the connector process
    for all u ∈ DOMINATEE do                                  ▷ Dominatee State
        for all v ∈ N(u) do
            if v ∈ MIS and u ∈ CDS then
                CDS ← CDS ∪ {u}             ▷ Receives Invite Connector Message
            end if
            if v ∈ MIS and u ∉ CDS then   ▷ Sends Volunteer Connector Message
                VOLUNTEER ← VOLUNTEER ∪ {u}
            end if
        end for
    end for
    for all u ∈ MIS do                                              ▷ MIS State
        for all v ∈ N(u) do
            if v ∈ (DOMINATEE ∩ VOLUNTEER) then
                Compute key(v)
                if key(v) is the largest then
                    CONNECTOR ← CONNECTOR ∪ {v}        ▷ Selects a connector
                end if
            end if
            if v ∈ CDS then                             ▷ Receives CDS Message
                CDS ← CDS ∪ {u}
            end if
        end for
    end for
end procedure

procedure ELIMINATECDS(G(V, E))                 ▷ Phase 3: CDS Elimination
    for all u ∈ CDS do
        if u is an edge node then
            for all v ∈ N(u) do
                if v ∈ CDS then
                    DOMINATEE ← DOMINATEE ∪ {u}
                end if
            end for
        end if
    end for
end procedure
```



(a) All nodes are initially in uncovered state.

(b) Node 0 sends *Dominator Message* and its neighbors become dominatee.

(c) Uncovered neighbors of dominatee become dominators.



(d) Node 0 sends *CDS Message* to nodes 1, 2 and 4.

(e) Node 4 is chosen as a connector to connect to nodes 3 and 5 respectively.

(f) Dominators 3 and 5 join the CDS.

(g) Node 5 is pruned from the CDS.

Figure 3.4: An example of a CDS generated using the TPSI algorithm. The MIS generation process involves steps (a), (b) and (c), the CDS generation process involves steps (d), (e) and (f) and the CDS pruning process involves step (g).

4 choose node 3 as a dominator and broadcast the selection using a *Dominatee Message*. Dominatee 2 on the other hand does not have uncovered neighbors, thus it completes the MIS generation.

- In the next round, dominatee 4 still has one uncovered neighbor 5. It then selects node 5 as a dominator and broadcasts the selection.
- At the completion of the MIS generation as indicated in Figure 3.4(c), all nodes are either in the dominator or dominatee states.

2. Figures 3.4(d), 3.4(e) and 3.4(f) show the CDS generation process.

   - In Figure 3.4(d), node 0 initiates the process as it is the first to complete the MIS generation. It changes its state to CDS and broadcasts a *CDS Message* to dominatee 1, 2 and 4.

- Upon the receipt of the *CDS Message*, these dominatees need to send a *Volunteer Connector Message* to their dominators to determine whether they are potential connectors. Therefore, dominatee 1 sends a *Volunteer Connector Message* to dominator 3, while dominatee 4 sends the message to dominators 3 and 5.

- Upon the receipt of the *Volunteer Connector Message* , dominators 3 and 5 set their timer to decide the eligible connectors. Since dominator 3 has a larger node degree and residual energy than that in dominator 5, it has a shorter delay and becomes the first to select a connector. It chooses the dominatee with the largest *key* as a connector, in this example dominatee 4 is selected. Dominator 3 then sends an *Invite Connector Message* to dominatee 4.

- Upon the receipt of the *Invite Connector Message*, dominatee 4 changes its state to a CDS as shown in Figure 3.4(e) and sends a *CDS Message* to dominators 3 and 5.

- Upon the arrival of the *CDS Message*, dominators 3 and 5 subsequently change their state to CDS to connect the DS.

- At the end of CDS generation, all nodes are either in the CDS or dominatee states as shown in Figure 3.4(f).

3. Figure 3.4(g) is the CDS pruning process. Node 5 is the pendant node defined by Definition 1. Since node 5 is covered by a CDS node 4, it is removed from the CDS and its state changes to dominatee.

3.4 Single-phase Single Initiator (SPSI) Algorithm

3.4.1 Overview

The SPSI algorithm is proposed to address the drawback of the TPSI algorithm. The TPSI algorithm introduces a long delay when computing the CDS. The primary reason for this is the multiple phases involved in finding the CDS. Since the CDS construction is a sequential process, the next phase has to wait for the previous phase to complete before continuing to the next phase. The delay increases as the network size increases.

The SPSI algorithm on the other hand requires a single phase to build the CDS. It does not involve finding a MIS, therefore eliminating the need of having three phases to construct the CDS. The CDS building process is also started by a sink node acting as an initiator. The eligible nodes are added to the CDS sequentially until all nodes are either in the CDS or covered by at least one node in the CDS.

Unlike in TPSI algorithm which requires only one-hop neighbors to construct the CDS, the SPSI algorithm needs both one-hop and two-hop neighborhood information. The one-hop neighbors are the direct neighbors of node u while the two-hop neighbors are the neighbors of direct neighbors of node u. To compute the two-hop neighbors, each node exchanges its list of one-hop neighbors with its neighbors. When node u receives the neighbor list from its neighbors v, it computes the two-hop neighbors of neighbor v.

The node can be in one of the three states. The three states are uncovered, dominator and dominatee. All nodes are initially in an uncovered state and at the end of the CDS construction, they will subsequently become either a dominatee or a dominator. Figure 3.5 shows the state transition diagram of a node during the construction of the CDS. Each node will discover its neighbors in order to

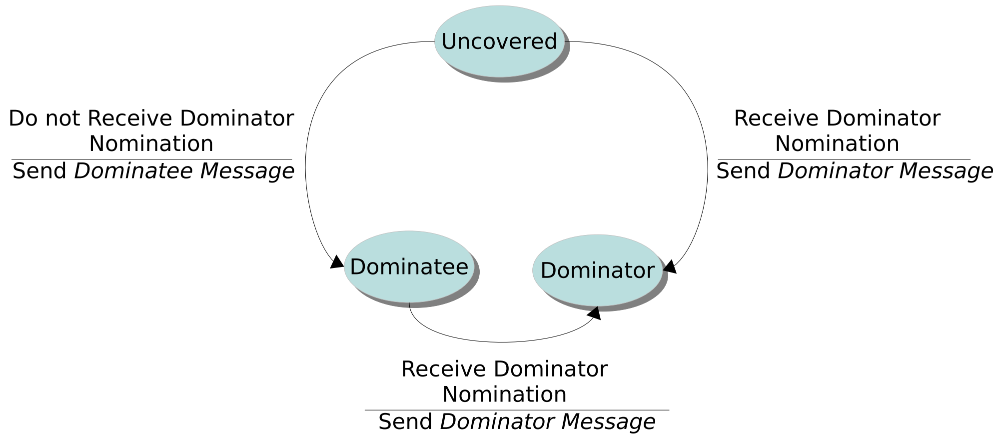


Figure 3.5: The state transition diagram of a node in the SPSI algorithm.

collect neighbor information required for the CDS construction. Section 3.4.2 explains the neighbor discovery mechanism executed by each node while Section 3.4.3 describes the process of generating a CDS.

3.4.2 Neighbor Discovery

The neighbor discovery process in the SPSI algorithm remains similar to that in the TPSI algorithm. Therefore, the SPSI algorithm will only consider nodes with symmetrical links as neighbors. These neighbors are updated from time to time as the link information changes. The neighbor discovery and maintenance mechanism in Appendix B is used to find and maintain the symmetrical neighbors.

The SPSI algorithm defines a beacon format as in Figure 3.6 and the beacon format varies between states. The dominatee and uncovered nodes have the same information in their beacon format. The dominator however, has an additional "Connector" field to choose the potential dominator to include in the CDS. The detailed description of the beacon information can be found in Section 3.3.2.

Uncovered	Node ID	State	Symmetrical Neighbors	Broken Neighbors	Residual Energy	Distance	
Dominator	Node ID	State	Symmetrical Neighbors	Broken Neighbors	Residual Energy	Distance	Connector
Dominatee	Node ID	State	Symmetrical Neighbors	Broken Neighbors	Residual Energy	Distance	

Figure 3.6: The beacon format of a node in the SPSI algorithm.

3.4.3 CDS Generation

The aim of the SPSI is to generate a small set of dominators while keeping the message overhead low. These dominators form the CDS in the network. Each dominator finds its local set of connectors among its one-hop neighbors to add into the CDS. The chosen connectors then become dominators and continue the connectors selection. This process is performed greedily until all nodes are either in the CDS or covered by a dominator in the CDS.

In order to find connectors, each dominator u needs to compute its two-hop neighbor information and then selects its connector among its 1-hop neighbor. The connector selection is based on the multipoint relay (MPR) concept in [QVL02] which has been enhanced to achieve energy conservation and reduce the CDS size. In order to minimize the CDS size, each dominator u chooses the connectors among its one-hop neighbors that have the most two-hop neighbors or selects the connectors that are the only intermediate nodes to u's two-hop neighbors.

The connector set for a given node u is computed as in Algorithm 5. Let u denote the dominator initiating the connector selection, $N_1(u)$ denote the one-hop neighbors of u, $N_2(u)$ denote the two-hop neighbors of u and $C(u)$ denote the set of chosen connector of node u. The term uncovered node refers to node in $N_2(u)$ that is not covered by $C(u)$. Let *span(v)* denote the quality of a potential connector

v used in case of a tie. The *span(v)* is computed by

$$span(v) = N_{\text{total}} E_r(v), \tag{3.4.1}$$

where N_{total} is the total number of one-hop neighbors of node v and $E_r(v)$ is the residual energy of node v.

Algorithm 5 The enhanced MPR Algorithm

1. Add $v \in N_1(u)$ to $C(u)$, if v is the only node that covers node $w \in N_2(u)$. Remove connector v from $N_1(u)$ and also the nodes covered by v from $N_2(u)$.
2. Add $v \in N_1(u)$ to $C(u)$, if v covers the largest number of uncovered nodes in $N_2(u)$. If there is a tie, choose the node with the largest *span(v)* to break the tie. Use node ID to break another tie.

Figure 3.7 illustrates the process of finding a connector set for dominator 4. Dominator 4 has to find connector nodes that can cover its two-hop neighbor nodes 6, 7 and 8. Based on the rule 1 of the Algorithm 5, dominator 4 chooses node 1 as the connector for its two-hop neighbor node 6 and based on the rule 2, it selects node 0 as the connector for its two-hop neighbor nodes 7 and 8. Hence, the connector set of dominator 4 is Connector(4)= {0,1}.

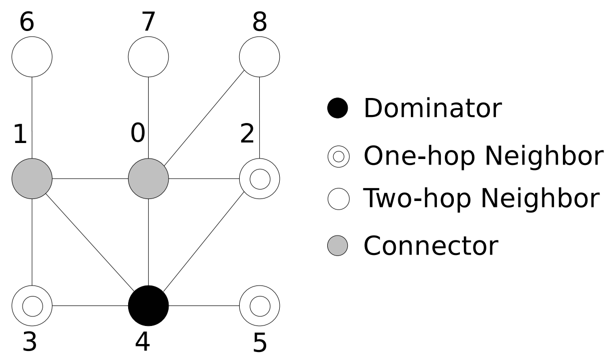


Figure 3.7: An example of finding a connector set for a given dominator 4.

CDS Construction Process

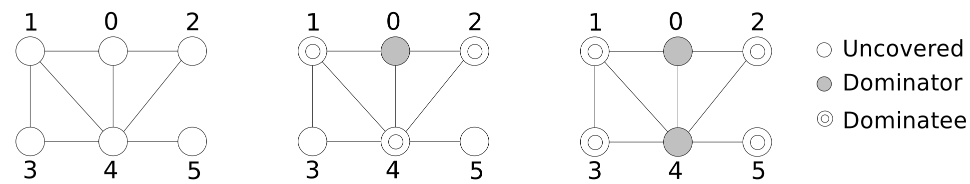
All nodes are initially in an uncovered state and at the completion of the CDS generation, they become either a dominator (CDS) or a dominatee (non-CDS).

The process of generating the CDS is described as follows:

- The CDS generation begins with the leader election based on the smallest node identifier. The leader changes its state to dominator and computes its connector set using the rules described by the Algorithm 5. It then broadcasts the connector selection to all neighbors.
- The uncovered node which is chosen as a connector changes its state to a dominator, then computes the connectors and finally broadcasts the connector selection along with its updated state to neighbors.
- The uncovered node which is not chosen as a connector changes its state to dominatee and broadcasts its updated state to neighbors.
- The dominatee which is chosen as a connector changes its state to dominator, then computes the connector and finally broadcasts the connector selection along with its updated state to neighbors.
- The dominator nodes will repeat the above steps until there is no uncovered node left in the network.

Algorithm 6 presents the pseudocode of the SPSI algorithm.

Figure 3.8 illustrates the process of constructing the CDS using the SPSI algorithm. The number next to nodes represents the node identifier.



(a) All nodes are initially in uncovered state.

(b) Dominator 0 chooses node 4 as a connector.

(c) Node 4 changes its state to dominator.

Figure 3.8: An example of a CDS generated using the SPSI algorithm.

Algorithm 6 CDS generation of the SPSI algorithm

Require: Graph $G(V, E)$

Ensure: Connected Dominating Set $(CDS) \in G(V, E)$

```
procedure GENERATECDS(G(V, E))
    CDS ← ∅ ; CONNECTOR ← ∅
    CDS ← CDS ∪ {i}                                   ▷ Initiator invokes the CDS generation
    for all u ∈ CDS do
        for all v ∈ N1(u) do
            Compute connector v using the enhanced MPR
            Add v ∈ CONNECTOR                         ▷ Elected dominator
        end for
    end for
    for all v ∈ N1(u) do
        if v ∈ CONNECTOR then
            CDS ← CDS ∪ {v}                           ▷ Dominator node
        end if
    end for
end procedure
```

The CDS process can be explained by the following steps:

1. In Figure 3.8(a), all nodes are initialized to uncovered state.

2. Upon receiving the beacon from each neighbor which contains the symmetrical neighbor list, each node then computes the two-hop neighbors of its one-hop neighbors.

3. The CDS generation begins with node 0 as shown in Figure 3.8(b). It changes its state to dominator, selects potential connectors and broadcasts the selection to its neighbors. In this example, dominator 0 chooses the uncovered node 4 as a connector since it is the only node covering the two-hop node 5 and can cover the most two-hop neighbors.

4. The uncovered nodes 1, 2 and 4 that receive the broadcast message from dominator 0 will then check whether they have been chosen as a connector or not. Since only node 4 is chosen as a connector, nodes 1 and 2 change

their state to dominatee while node 4 changes its state to dominator. All two-hop neighbors of dominator 4 are covered, thus dominator 4 terminates the connector selection and broadcasts its updated status to neighbors.

5. The uncovered nodes 3 and 5 then change their status to dominatee upon the receipt of the broadcast message from dominator 4.

6. At the end of the CDS generation, all nodes are either in a dominator or a dominatee state as indicated in Figure 3.8(c).

3.5 Implementation Considerations

There are two issues considered during the implementation of the TPSI and SPSI algorithms: (i) Topology maintenance which is the mechanism for handling topology changes as explained in Section 3.5.1 and (ii) Guaranteed connectivity which is the act of preserving a link between a pair of nodes in the CDS as described in Section 3.5.2.

3.5.1 Topology Maintenance

Topology maintenance concerns with the ability of the algorithms to adapt to changing network topology while maintaining the CDS intact. Changes in the network topology occur when:

1. Nodes are mobile thus causing them to either leave or join the CDS.

2. Nodes become malfunction due to depletion of energy supply or failure in hardware devices.

3. Links between nodes are interrupted or lost due to interference.

When topology changes, the algorithms should be quick enough to adapt to these changes without incurring extra communication and computation overheads. These requirements are the popular subject of study in topology maintenance algorithms. According to [LW09], these algorithms can be implemented in three ways depending on the time when the new topologies are built. They can build a set of topologies for the reconstruction before the maintenance process, known as *static topology maintenance* or during the reconstruction process called *dynamic topology maintenance* or a combination of both *static topology maintenance* and *dynamic topology maintenance*.

In general, topology maintenance algorithms can be classified into two categories according to the types of reconstruction required. The reconstruction of the whole CDS is about recomputing the entire CDS. For instance, when the existing CDS no longer able to maintain a CDS [SSKO08]. While the partial reconstruction of the CDS is done if the following event happens: (i) part of clusters of the CDS is broken [ZSL05] or (ii) connectors or bridges connecting two or more clusters fail [AWF02].

Regardless of the types of reconstruction, these algorithms operate only when they are triggered by a certain event. This event typically uses a criterion to invoke the reconstruction. Various possible criteria used for triggering the reconstruction are described in [LW09]. These criteria ranges from time, energy, random variable, failure-based, density-based to the combination of these criteria. The choice of criteria may affect the performance of the network from the aspects of energy saving, coverage, reliability, message overheads, delay to throughput [LW09].

Since TPSI and SPSI are designed for energy-efficiency, energy should be adopted as a criterion for triggering the computation of the CDS. When any of the nodes in the CDS runs out of its energy, the whole CDS will be recomputed. This approach avoids changing the CDS too often to save energy resources. The reconstruction of the whole CDS is chosen over the partial reconstruction of the CDS to keep the CDS size small and limit the number of exchanged messages involved when reconstructing certain clusters of the broken CDS. The implications of this criterion on the network performance however, are not thoroughly investigated as they are beyond the scope of this thesis.

3.5.2 Guaranteed Connectivity

Connectivity measures the presence of a path between any pair of nodes in the network after the network is subjected to topology control techniques. Specifically, after the exercises of TPSI and SPSI algorithms on the network topology, any two nodes in the CDS should be linked by a path to allow smooth data transfer between nodes and to avoid network partitioning. In WSNs, preserving the connectivity among nodes in the CDS is very crucial as these nodes are bound to fail during their operation.

The property of the CDS requires nodes that are not in the CDS to be a neighbor to a node in the CDS. Therefore, it ensures that there exist at least one path between a set of two nodes, thus providing a simple mechanism to maintain the connectivity between nodes. As long as the CDS is preserved throughout the operation, the connectivity of the network is hence guaranteed.

In this thesis, the connectivity of nodes in the CDS is guaranteed by validating the formation of the CDS using a parameter called coverage. The 100% coverage indicates the success of the TPSI and SPSI algorithms in building the CDS, whereas the coverage other than 100% shows that the algorithms fail to construct a CDS. TPSI and SPSI should yield a 100% coverage to ensure nodes that are not part of the CDS (dominatees) are covered by at least one-hop neighbor in the CDS (dominator). Once the CDS is successfully constructed, the backbone operates for a duration of time until further reconstruction is invoked.

3.6 Conclusions

This chapter has presented two algorithms TPSI and SPSI, which construct the CDS based on a single initiator. They compute the CDS using localized information and perform the CDS generation in distributed fashion. In order to reduce the communication overhead and energy consumption, TPSI and SPSI minimize the number of exchanged messages among nodes. To keep the CDS size small and to prolong the lifetime of the CDS, the metrics that consider the node degree and residual energy are used.

Even though TPSI can generate a small CDS, it requires three phases to build the CDS. Thus, it converges slowly. SPSI on the other hand is designed for fast convergence applications. Using a single phase, it computes the CDS quickly with low message exchanges and forms a small backbone in networks. Unfortunately, TPSI and SPSI cannot handle the dynamic topology changes in the network. Everytime initiators fail, new construction of CDS is needed. Therefore, a topology control algorithm that can manage frequent topology changes is proposed in the following chapter.

Chapter 4

Distributed Multiple Initiator CDS Algorithm

4.1 Introduction

This chapter presents a novel CDS algorithm for handling topology changes called two-phase multiple initiator (TPMI) algorithm. TPMI is an adaptive and scalable algorithm designed for constructing and maintaining a CDS. CDS maintenance is a challenging issue in WSNs due to the dynamic nature of the networks. Moreover, the networks are often deployed in harsh environments. Therefore, nodes are highly susceptible to failures. In such volatile environments, an efficient algorithm must be able to cope with the topology changes while maintaining the CDS as long as possible. Unnecessary reconstruction of the CDS should be avoided to minimize the communication overhead in the networks, hence conserving the energy resources.

There are several drawbacks of the TPSI and SPSI algorithms described in Chapter 3. Firstly, the CDS construction is a chained process. The CDS computation begins from a single initiator, and then propagates sequentially to the whole network. As a result, the entire process in general is time consuming because an inconstant number of rounds is involved. Under dynamic topology such as node mobility, the single initiator approach is not quick enough to disseminate updated information to all nodes. As a consequence, some information may already become outdated.

Secondly, since the CDS construction starts from a single root (initiator) node, it is highly sensitive to the root node failure. The failure would trigger the reconstruction of the CDS. A frequent and unnecessary CDS reconstruction often leads to high energy consumption in the network as additional communication overheads are involved for re-running the CDS construction from scratch.

Finally, it is assumed that the topology of the network must be maintained throughout the CDS construction. In other words, the information used for constructing the CDS remains the same for a period of time until the CDS is finally constructed. In practice, this assumption is not true because the neighbor information is frequently updated as the topology changes. As a result, the CDS constructed may become invalid due to incorrect neighbor information used for building the backbone.

Motivated by the above drawbacks, TPMI is proposed to cope with the limitations of the single initiator algorithms. TPMI does not rely on a single node (leader) to initiate the construction of the CDS. Instead, all nodes execute the CDS construction simultaneously across the whole network. This strategy allows nodes to make decisions quickly based on two-hop neighbor information. Whenever the neighbor information changes, the relevant nodes will update their changes to the surrounding nodes within a limited number of hops. Hence, any local changes will not affect the rest of the nodes in the network and will not necessarily trigger the CDS reconstruction.

The disadvantage of multiple initiators over the single initiator is that it can generate a large CDS size in the network due to the simultaneous election of multiple nodes. As a result, the CDS size of TPMI can be larger than the CDS size of TPSI and SPSI.

TPSI algorithm has several unique features:

- It is a fully distributed and localized algorithm capable of maintaining the CDS under dynamic topology changes. It is run at multiple nodes as opposed to a single node and relies on localized information provided by neighbors through beacons.

- It is realistic to implement in practice, namely, it has a mechanism to handle the presence of both asymmetrical and symmetrical links in the network and can cope with a variable transmission range of sensor nodes.

- It guarantees the network connectivity such that every node is either in the CDS or is adjacent to one of the nodes in the CDS.

- It keeps the size of the backbone to a minimal level to minimize the CDS maintenance overhead and conserve energy.

- It extends the network lifetime by considering several metrics such as load balancing, residual energy and energy involved in transmission.

The rest of this chapter is organized as follows. Section 4.2 presents the assumptions made in the design of the TPMI algorithm. Section 4.3 describes the design aspects of the algorithm and the processes involved for building the CDS.

4.2 WSN Model and Assumptions

The network model and assumptions of the TPMI algorithm are similar to the ones used in the TPSI and SPSI algorithms proposed in Chapter 3.

4.3 Two-phase Multiple Initiator (TPMI) Algorithm

4.3.1 Overview

The TPMI algorithm constructs a CDS in two phases. The first phase is the DS generation process which finds a MIS of the graph G. The second phase is the CDS generation process which finds connectors to connect the DS found by the first phase. Together, the DS and connectors form a CDS in the network. The two phases are summarized in Algorithm 7.

Algorithm 7 The implementation of TPMI algorithm

Require: Graph $G(V, E)$
Ensure: Connected Dominating Set $(CDS) \in G(V, E)$

```
procedure TPMI(G(V, E))                 ▷ Main procedure
    GenerateDS(G(V, E))                 ▷ Phase 1 of TPMI
    GenerateCDS(G(V, E))                ▷ Phase 2 of TPMI
end procedure
```

The algorithm assigns every vertex in a given graph $G = (V, E)$ to a state. The possible states are uncovered, dominator and dominatee. All vertices are initially in an uncovered state. At the completion of the CDS construction, they will remain in either a dominatee or a dominator state as indicated by Figure 4.1. The dominators form the CDS of the graph.

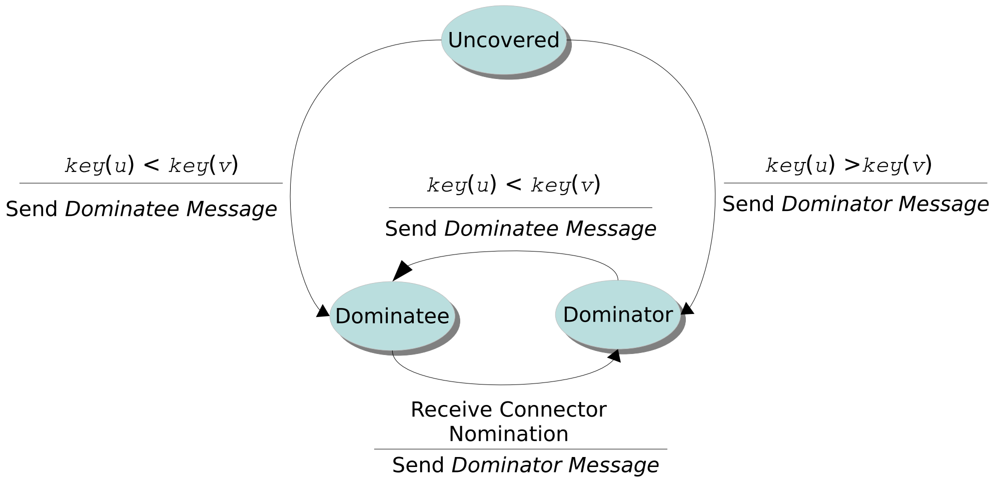


Figure 4.1: The state transition diagram of node u in the TPMI algorithm. Node v is the neighbor of node u.

4.3.2 Neighbor Discovery

The TPMI algorithm uses the similar neighbor discovery mechanism as in TPSI and SPSI algorithms. Hence, it requires the neighbor discovery and maintenance mechanism in Appendix B to find symmetrical neighbors.

Each node in the network exchanges a beacon among its one-hop neighbors at a periodic time interval. The beacon created for TPMI contains information needed for constructing a CDS. Figure 4.2 shows the content of the proposed beacon. The content of the beacon varies between states.

Uncovered	Node ID	State	Symmetrical Neighbors	Broken Neighbors	Residual Energy	Distance	
Dominator	Node ID	State	Symmetrical Neighbors	Broken Neighbors	Residual Energy	Distance	Connector
Dominatee	Node ID	State	Symmetrical Neighbors	Broken Neighbors	Residual Energy	Distance	

Figure 4.2: The beacon format of a node in the TPMI algorithm.

The dominator has an additional field for connectors other than the uncovered or dominatee nodes to broadcast the selection of the connectors. The beacon information is similar to one used in the single initiator algorithms in Chapter 3, thus the detailed description of the beacon can be referred to Section 3.3.2.

4.3.3 DS Generation

TPMI algorithm finds a maximal independent set (MIS) to create a dominating set (DS). In order to select the DS in the network, a $key(u)$ which represents the key of node u is adopted. It is basically a function of node degree and residual energy defined by Equation 4.3.1. The key is expected to select nodes with large node degree and high remaining energy to reduce the size of the CDS and extend the network lifetime respectively. In case of a tie, a larger node id is chosen.

$$key(u) = nodedegree(u) + residualenergy(u) \tag{4.3.1}$$

The process of generating the DS can be described as follows:

- Initially, assign an uncovered node state to every node $u \in V$.
- Each node u broadcasts a beacon containing information described in Figure 4.2 and discovers its symmetrical neighbors.
- An uncovered node u assigns its state to a dominator if it has the largest $key(u)$ among its one-hop neighbors or its neighbors are all dominatees. It then broadcasts a *Dominator Message* to neighbors.
- An uncovered node u assigns its state to a dominatee if it has a dominator v with a larger $key(v)$ than $key(u)$ and broadcasts a *Dominatee Message*.
- A dominator u assigns its state to a dominatee if it has a dominator v with a larger $key(v)$ than $key(u)$. It then announces its state using a *Dominator Message*.
- In case of a tie between node u and v, the largest id node is selected.

Algorithm 8 provides the detailed implementation of the DS generation.

Algorithm 8 The DS generation of the TPMI algorithm

Require: *G(V,E)*
Ensure: *DS* $\in$ *G(V,E)*

  procedure GENERATEDS($G(V,E)$) $\triangleright$ Phase 1: Find a DS in the network
    DS $\leftarrow \emptyset$; *DOMINATEE* $\leftarrow \emptyset$; *UNCOVERED* $\leftarrow \emptyset$
    for all $u \in V$ **do**
      UNCOVERED $\leftarrow$ *UNCOVERED* $\cup \{u\}$
    end for
    for all $u \in UNCOVERED$ **do**
      for all $v \in N_1(u)$ **do**
        if *key(u)* $>$ *key(v)* **then**
          DS $\leftarrow$ *DS* $\cup \{u\}$
        else
          DOMINATEE $\leftarrow$ *DOMINATEE* $\cup \{u\}$
        end if
      end for
    end for
    for all $u \in DS$ **do**
      for all $v \in N_1(u)$ **do**
        if *key(u)* $<$ *key(v)* **then**
          DOMINATEE $\leftarrow$ *DOMINATEE* $\cup \{u\}$
        end if
      end for
    end for
  end procedure

4.3.4 CDS Generation

This phase aims to find connectors to connect to the DS formed by the first phase. The connectors and the DS will form the CDS in the network. Each dominator u finds a connector set $C(u)$ among its one-hop neighbors that can reach its two-hop neighbors. The chosen connector will then become a dominator and triggers the connector selection among its one-hop neighbors. This process is performed in a distributed manner across the network until all nodes in the DS are connected.

Connector Selection

Similar to the TPSI algorithm, the connector selection process uses the enhanced MPR algorithm to keep the connector set small in size. The only difference is that this process is computed simultaneously by multiple dominators across the network. In order to compute the connector, each node requires the knowledge of its two-hop neighbors. Each node has the list of symmetrical neighbors of each of its one-hop neighbors. Therefore, it can easily determine the two-hop neighbors locally.

Each dominator u computes its connector $C(u)$ among its one-hop neighbors $N_1(u)$, that has the most two-hop neighbors $N_2(u)$ using an enhanced multipoint relay (MPR) described in Algorithm 5 in Chapter 3.

The process of generating the CDS can be described as follows:

- Each dominator u computes its connector set as in Algorithm 5.
- A dominatee chosen as a connector changes its state to a dominator and broadcasts a *Dominator Message*.
- The above processes are repeated until all two-hop neighbors of dominator u have at least one dominator among their one-hop neighbors.

Algorithm 9 provides the detailed description of the CDS generation.

Figure 4.3 illustrates the process of constructing the CDS using the TPMI algorithm. In this figure, the number labeled beside each node represents the node identifier. For simplicity, let assume that all nodes have identical energy level. Therefore, their energy can be ignored so that the *key* value is determined solely by node degree.

Algorithm 9 The CDS generation of the TPMI algorithm

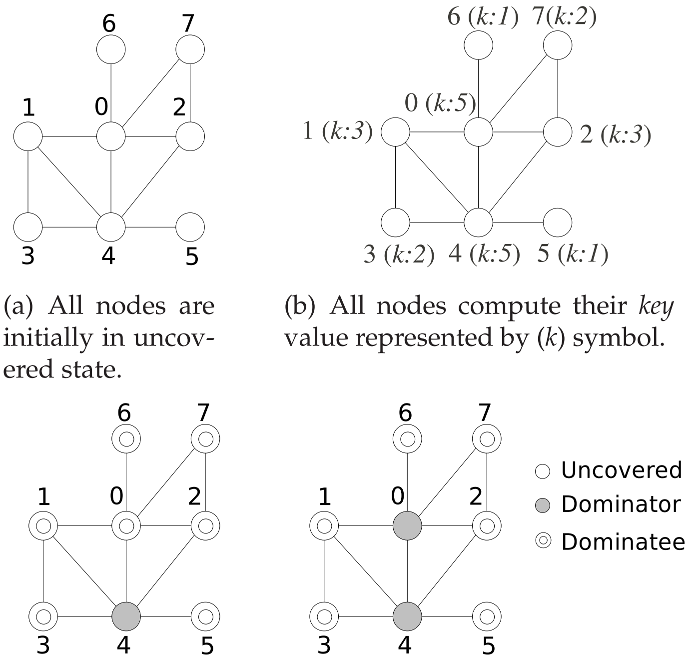
Require: *DS* and *DOMINATEE*

Ensure: *CDS* $\in$ *G(V,E)*, where *CDS* $\leftarrow$ *DOMINATOR* $\cup \{DS\}$

```
procedure GenerateCDS(G(V, E))          ▷ Phase 2: Find a CDS in the network
    CONNECTOR ← ∅; DOMINATOR ← ∅
    for all u ∈ DS do
        for all v ∈ N_1(u) do
            Compute v using Algorithm 4                ▷ Connector Nomination
            CONNECTOR ← CONNECTOR ∪ {v}
        end for
    end for
    for all u ∈ DOMINATEE do
        if u ∈ CONNECTOR then
            DOMINATOR ← DOMINATOR ∪ {u}       ▷ Dominator Assignment
        end if
    end for
end procedure
```

The CDS process can be explained by the following steps:

1. In Figure 4.3(a), all nodes are initialized to uncovered state.
2. Each node u sends a beacon to find its symmetrical neighbors.
3. All nodes u compute their *key(u)* and *key(v)* of their neighbors v. Figure 4.3(b) indicates the computed *key(u)* values.
4. Figure 4.3(c) illustrates the DS generation. All nodes u evaluate their own *key(u)* simultaneously in the network upon the receipt of a beacon from a neighboring node v. If its *key(u)* is the largest, it declares itself as a dominator. In this figure, both uncovered nodes 4 and 0 have the largest key among their neighbors. To break the tie, the larger node id is used. Thus, node 4 changes its state to dominator. Node 0 on the other hand becomes a dominatee.



(a) All nodes are initially in uncovered state.

(b) All nodes compute their *key* value represented by (*k*) symbol.

(c) Node 4 becomes a dominator because it has the largest *key* and id among its one-hop neighbors.

(d) Node 0 changes its state to dominator since it has been chosen as a connector that can connect the two-hop neighbors of dominator 4.

Figure 4.3: An example of a CDS generated using the TPMI algorithm. Figure 4.3(c) is the DS generation and Figure 4.3(d) is the CDS generation.

5. The rest of the nodes in the network become dominatees because their neighbors have a larger *key*. At the end of the DS generation all nodes remain in a dominator state or a dominatee state.

6. Figure 4.3(d) shows the CDS generation. During this process, the dominator 4 finds its connector among its one-hop neighbors to reach its two-hop neighbors 6 and 7. The potential connectors in this case are dominatee 0 and dominatee 2. Dominatee 0 is chosen as a connector because it covers the largest number of two-hop neighbors and it is the only node connecting the dominator 4 to node 6.

7. Upon the nomination of connector, dominatee 0 changes its state to dominator as illustrated in Figure 4.3(d).

4.4 Conclusions

A multiple initiator algorithm TPMI has been proposed in this chapter. The aim of this chapter is to design a CDS technique that can cope well with topology changes in the network. Similarly to TPSI and SPSI, TPMI uses localized information gathered from neighbors that are within two-hops away to compute the CDS. The residual energy and the node degree of nodes are both considered when choosing the CDS nodes so that the CDS size can be minimized and the network lifetime can be extended.

Chapter 5

Theoretical Analysis of Proposed Algorithms

5.1 Introduction

This chapter presents the verification and performance analysis of the proposed algorithms when the underlying topology is modeled as a unit disk graph. The verification is conducted to confirm the ability of the algorithms in generating a CDS of a graph. The performance analysis on the other hand evaluates two important measures: (1) the quality of the generated CDS measured by its size and (2) the implementation costs of the algorithms in terms of message and time complexities. These two measures determine the computation and communication overheads as well as the energy consumption involved in constructing and maintaining the CDS. This chapter shows that the proposed algorithms not only construct a CDS of a bounded size but they also have low communication and computation overheads. Due to the low costs and small generated CDS, they provide energy-efficient and low maintenance solutions for a dynamic environment.

5.2 Verification of Proposed Algorithms

This section verifies the algorithms proposed in Chapter 3 and Chapter 4 by investigating the correctness of the generated CDS.

5.2.1 Verification of TPSI Algorithm

This section provides the verification of the TPSI algorithm in generating a CDS. Since TPSI algorithm constructs a CDS in three phases, the verification of the algorithm is also proven in three steps. The first step confirms that the MIS generation phase can correctly generate a MIS of a graph. The second step proves that the CDS generation phase can connect the MIS formed during the first phase to guarantee the connectivity of the network. Finally, the third step shows that the CDS pruning phase can still maintain the CDS.

Lemma 5.1. *For a given connected graph $G(V, E)$, the MIS generation creates a dominator d that is two hops away from any other dominators in G.*

Proof. Recall that all nodes are initially in an uncovered state except the initiator, which later becomes a dominator. The neighbors of the dominator must become dominatees and choose exactly one dominator among their uncovered neighbors. The rule of the TPSI algorithm makes sure that only one uncovered node (i.e. with the largest key) can be chosen as a dominator at a given time. If the key is a tie, a unique identifier breaks the tie, in which the node with the larger identifier is chosen as a dominator. When the chosen node becomes a dominator, its neighbors must become dominatees. These dominatees then select a dominator among their uncovered one-hop neighbors. As a consequence, the dominators in G are two hops away from each other.

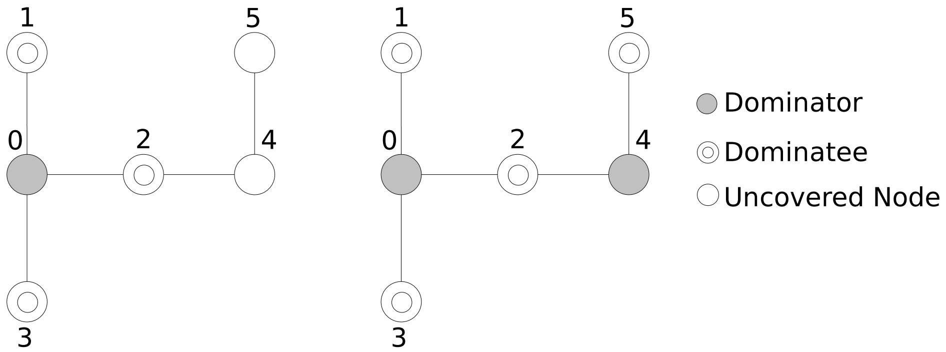
Lemma 5.2. *For a given connected graph $G(V, E)$, the MIS generation process forms a MIS which consists of a set of dominators, D. Thus, D is a MIS of G, which is also a DS.*

Proof. To prove this lemma, D must satisfy all three properties of the MIS. First, D is an independent set (IS) of G. Next, any node not in D must have at least one neighbor in D. Finally, the addition of any node not in D will break the independence property of the MIS.

For the first property, in order to become an IS, no two dominators in D can be adjacent. According to Lemma 5.1, two dominators in G are two hops away from each other. Thus, D must be an IS.

The second property is proven using contradiction. Let assume that there is node 5 which is not a member of D and does not have any neighbor in D as shown in Figure 5.1(a). Node 5 has to be an uncovered node in order to satisfy this assumption and is three hops away from dominator 0. Based on the MIS generation process described in Section 3.3.4, dominatee 2 will select node 4 as a dominator. Node 4 will change its state to a dominator and broadcasts a *Dominator Message* to node 5. As a result, node 5 is a neighbor to a node in D and must change its state to a dominatee as depicted in Figure 5.1(b). This result contradicts the assumption made.

The final property can be easily proven as in [LSM06] using contradiction. Recall that D is a set of dominators. Assume that D is not a MIS of G. Hence, there must be a node in G for example node 2 as shown in Figure 5.2 that is not a dominator and it can be added to D. To add node 2 into D it has to be a dominator, thus it must be at least two-hops away from node 0 to satisfy an independent property of MIS. Assume that node 1 connects node 0 to node 2, and based on the rule of the MIS generation described in Section 3.3.4, node 2 will eventually change its state to a dominator, which contradicts the assumption.



(a) Node 5 is not a member of D and not adjacent to any node in D.

(b) Node 5 will eventually have a dominator neighbor in D.

Figure 5.1: Validation of the MIS property. Any node u not in D must have a neighbor in D.

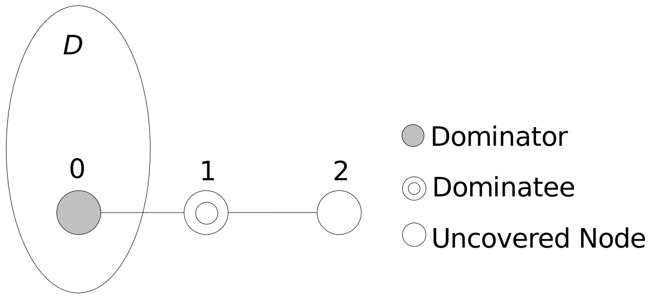


Figure 5.2: Adding node 2 into D breaks the independence property of the MIS.

Since D satisfies all three properties of the MIS, this lemma is proven.

Lemma 5.3. *All dominators in D are joined by at least one connector during the CDS generation process.*

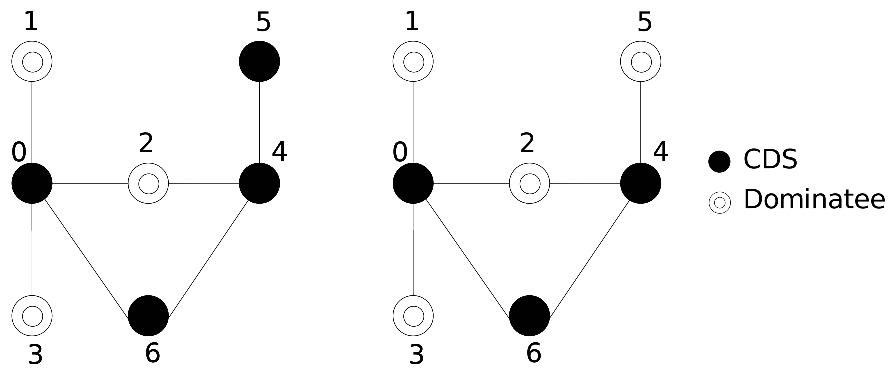
Proof. Let D be a set of dominators and E is the set of dominatees formed after the completion of the MIS generation as shown in Figure 5.1(b). Based on Lemma 5.1, two dominators 0 and 4 are two hops away from each other while Lemma 5.2 proves that all dominatees 1, 2, 3 and 5 must have at least one dominator as neighbor. This implies that any two dominators always have at least one node among these dominatees. When the rule of the CDS generation described in Section 3.3.5 is applied to these dominatees, they become potential connectors and the most eligible nodes among them will be chosen as a connector. Thus, all dominators are joined by at least one connector.

Theorem 5.4. *For a given connected graph G, the TPSI algorithm generates a CDS of the graph.*

Proof. To form a CDS, the TPSI algorithm must first construct a DS in the graph and then connect the DS with connectors. Both steps are proven by Lemma 5.2 and Lemma 5.3 respectively. Hence, this theorem is proven.

Theorem 5.5. *For a given connected graph G consisting of more than two vertices, the CDS pruning process preserves the CDS in the graph.*

Proof. The CDS pruning rule applies only to a pendant node 5 that is in a CDS state and has a neighbor in the CDS as illustrated by Figure 5.3(a). It can be clearly seen from the figure, when node 5 is eliminated from the CDS, it becomes a dominatee as shown in Figure 5.3(b). Since node 5 is adjacent to dominator 4 in the CDS, the CDS is still formed in the network.



(a) Node 5 is a pendant node to be pruned from the CDS.

(b) The network remains connected after the CDS pruning.

Figure 5.3: Proving Theorem 5.5. The CDS is maintained even after the pruning of the pendant node.

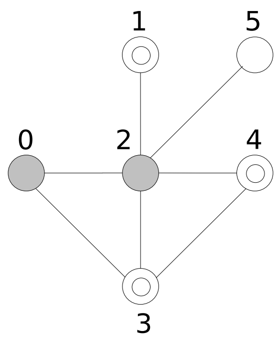
Based on the Theorem 5.4 and Theorem 5.5, the TPSI algorithm can construct a CDS in a given network.

5.2.2 Verification of SPSI Algorithm

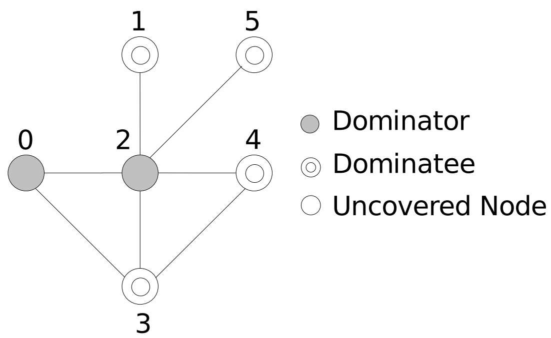
This section verifies that the SPSI algorithm correctly builds a CDS in a network.

Lemma 5.6. *For a given connected graph G, there is no uncovered node at the completion of the CDS generation.*

Proof. This lemma is proven by contradiction. Assume there exists an uncovered node 5 when the CDS generation terminates. Let consider two possible scenarios of node 5: (i) its neighbor is a dominator, node 2 as illustrated in Figure 5.4 and (ii) its neighbor is a dominatee, node 1 as shown in Figure 5.5. Given that every node must send beacons to discover neighbors, if the uncovered node has edges connecting to its neighbors, it has to change its state to either a dominatee or dominator. In these examples, since node 5 has a dominator among its neighbors, it changes its state to a dominatee as illustrated in Figure 5.4(b) and Figure 5.5(b), which contradicts the assumption.

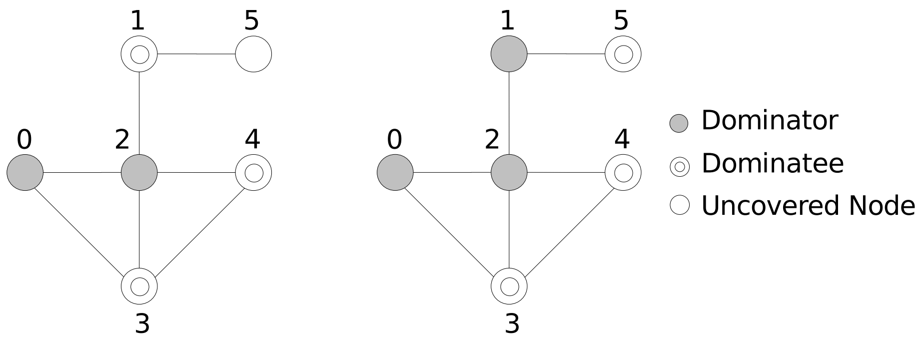


(a) Uncovered node 5 connected to a dominator.



(b) Node 5 will eventually become a dominatee when node 2 changes its state to a dominator.

Figure 5.4: All nodes in a connected network will either remain in a dominatee or a dominator state upon the completion of the CDS generation.



(a) Uncovered node 5 is connected to a dominatee.

(b) Node 5 will eventually become a dominatee when node 1 changes its state to a dominator.

Figure 5.5: All nodes in a connected network will either remain in a dominatee or a dominator state upon the completion of the CDS generation.

Lemma 5.7. *For a given connected graph G, the dominatees are adjacent to at least one dominator.*

Proof. Assume that D is a set of dominators. Once a node becomes a dominator, it remains in the state. As described in Section 3.4.3, an uncovered node changes its state to a dominatee only if it receives a message from a dominator d in D. Hence, a dominatee has at least one dominator in D as illustrated in Figure 5.4.

Lemma 5.8. *For a given connected graph G, the dominators are adjacent to at least one dominator, thus forming a connected subgraph G′ at the completion of the CDS generation.*

Proof. The CDS generation process in Section 3.4.3 requires each dominator d to choose a connector among its one-hop neighbors. Recall that the chosen connector will become a dominator which will then continue searching for connectors until all its two-hop neighbors are adjacent to a dominator. This can be explained using Figure 5.5. Assume that dominator 0 initiates the CDS generation and chooses node 2 as a connector. Node 2 then changes its state to a dominator and further initiates the connector selection. It selects node 1 as a connector since its two-hop neighbor node 5 has no dominator. Upon receiving the connector election from node 2, node 1 changes its state to a dominator. The search of connector stops when all nodes in the figure have a dominator. At the completion of the CDS generation, two dominators are adjacent and the network remain connected.

Theorem 5.9. *For a given connected graph G, the dominators form a CDS at the completion of the CDS generation.*

Proof. At the termination of the CDS generation, a node is either in dominatee or dominator state as proven by Lemmas 5.6 and 5.7, and the dominators are connected as shown in Lemma 5.8. Therefore, the set of dominators D generates a CDS in the network.

5.2.3 Verification of TPMI Algorithm

In this section, the validation of the TPMI algorithm is performed in two steps. The first step shows that the algorithm can form a DS of a graph and the second step proves that the DS is fully connected.

Lemma 5.10. *For a given connected graph $G(V, E)$, any pair of dominators formed by the DS generation are separated by at least two hops or at most three hops.*

Proof. Each node u evaluates its *key* against its neighbor's *key* as discussed in Section 4.3.3. Because the *key* is unique for each node, two adjacent nodes cannot be dominators. As a result, dominators are at least two hops aways from each other.

A dominator can be at most three hops away from other dominator. This is proven by contradiction. Let D denote the set of dominators created by the first phase of the TPSI algorithm as shown in Figure 5.6. In this figure, dominators 1 and 5 are four hops away from each other. Let assume that node 6 is in uncovered state and is not a member of D. Since all neighbors of uncovered node 6 are in dominatee state, node 6 changes its state to a dominator. Hence, it is in D and two hops away from dominators in D, which contradicts the assumption. Therefore, dominators cannot be separated by more than three hops.

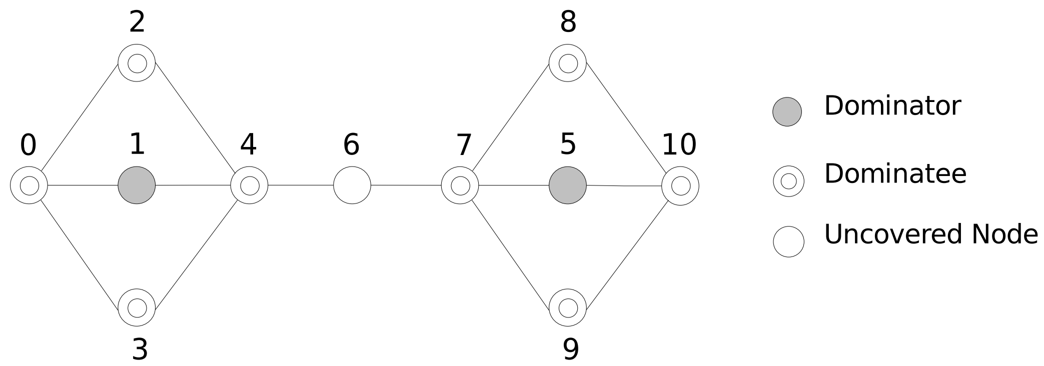


Figure 5.6: Proving that two dominators in DS cannot be more than three-hops away from each other.

Lemma 5.11. *At the completion of the DS generation process, all dominators form a DS of a graph. Therefore, a DS is also a MIS.*

Proof. This lemma is proven similarly as in Lemma 5.2, in which three properties of the MIS must be proven. It is obvious from Lemma 5.10 that the set of dominators is an IS due to the fact that no two dominators are adjacent. Thus, the IS property is proven. The lemma also shows that each dominatee in the graph has at least one dominator among its one-hop neighbors. Therefore, the second property of the MIS is also satisfied. The third property can be easily proven that no other nodes in the graph can be added to the DS. Assume that node u is an uncovered node and one hop away from any dominators. According to Lemma 5.10, it is not possible for u to be a dominator as dominators must be separated by at least two hops distance. Therefore, node u cannot be added to D or else it will violate the independence property of the set. Thus, the third property is achieved.

Since the DS generated by the TPMI algorithm satisfies all three properties of the MIS, then the DS is an MIS. Hence, this lemma is proven.

Lemma 5.12. *The CDS generation process connects the DS by a set of connectors C.*

Proof. The second property of the Lemma 5.11 proves that a pair of dominators must have at least one dominatee among their neighbors. These dominators will find a connector among its dominatees using the enhanced MPR rule explained in Section 4.3.4 until all two-hops neighbors are covered by a dominator. The CDS generation terminates when the connectors are found, which establishes that dominators are connected. Hence, this process guarantees the connectivity of dominators via connectors, which then proves this lemma.

Theorem 5.13. *Given a connected graph $G(V, E)$, the TPMI algorithm constructs a CDS of the graph.*

Proof. This theorem is proven by combining the Lemma 5.11 with Lemma 5.12.

5.3 Performance Analysis of Proposed Algorithms

This section measures the performance of the proposed algorithms according to their message complexity, time complexity, information range and approximation ratio so that the comparison to the surveyed algorithms in Section 2.4 can be made. The message complexity measures the communication complexity of the algorithms, the time complexity evaluates the computational complexity of the algorithms and the information range defines the range of information (in hops) needed by the algorithms to compute a CDS. Generally, algorithms that rely on a small range of information can quickly gather information in the network, thus they are more efficient.

Since computing a minimum CDS is an NP-problem [GJ79], the TPSI, SPSI and TPMI algorithms are approximate CDS solutions. An approximation ratio measures the performance of these algorithms in terms of their ability to minimize the CDS size. It is computed by taking the ratio of the largest CDS size obtained by the approximate algorithms to the minimum CDS [WDJ+06]. Obviously, the smaller performance ratio gives the better performance.

5.3.1 Performance of TPSI Algorithm

In TPSI algorithm, node u requires the ID of its two-hop neighbors to compute the node degree of its one-hop neighbors and determine the symmetrical neighbors. Recall that during neighbor discovery process, node v broadcasts the list of its one-hop neighbors to node u together with the the residual energy and state. Hence, the information range of the TPSI algorithm is two hops.

Theorem 5.14. *TPSI has $O(n)$ total time complexity and $O(n)$ total message complexity, where n is the total number of nodes in the network.*

Proof. The overall time complexity of the TPSI algorithm is the time consumed for constructing the MIS and CDS and the time spent on pruning the generated CDS. During the MIS generation, each dominatee needs $O(\triangle_d)$ time to find the potential dominator for the MIS, where $\triangle_d$ is the node degree. Therefore, this process has a constant time. During the CDS generation, the time complexity is measured by the time taken for finding connectors to join the MIS. Each CDS node waits a constant time, $O(\triangle_d)$ to build its connector set. The time involved in the CDS pruning is also $O(\triangle_d)$ because each pendant node has a constant number of one-hop neighbors to prune. Due to the constant time involved in the three phases, the algorithm has a total time complexity of $O(n)$.

In this algorithm, each node broadcasts a constant number of messages. During the MIS generation, each node sends at most one message of either a *Dominator Message* or a *Dominatee Message*. During the CDS generation, the worst case occurs when a dominatee, which is nominated as a connector sends two messages; *Volunteer Connector Message* and *CDS Message*. Since each node sends a bounded number of messages, its total message complexity is $O(n)$.

Lemma 5.15. *The size of a MIS generated by the first phase is at most* $3.8 \cdot opt + 1.2$*, where opt is the size of a minimum CDS (MCDS) as deduced in [WDJ+06].*

Proof. Wu et. al [WDJ+06] proved that each node is adjacent to at most $3.8 \cdot opt + 1.2$ nodes in the MIS. Therefore, the size of the MIS generated by the algorithm during the first phase is also bounded by $3.8 \cdot opt + 1.2$.

Lemma 5.16. *The size of the connector set C found during the second phase of the algorithm is bounded by* $3.8 \cdot opt + 0.2$.

Proof. The TPSI algorithm chooses the connectors among the dominatees formed by the first phase. Let *M* denote the MIS size and *C* is the size of a connector set. Recall that to become a connector, each dominatee must receive two messages: a *CDS Message* from a node in *M* and also an *Invite Connector Message* from another node in *M*. Therefore, $C \leq M - 1$. From Lemma 5.15, $C \leq 3.8 \cdot opt + 1.2 - 1 \leq 3.8 \cdot opt + 0.2$.

Theorem 5.17. *TPSI algorithm has an approximation ratio of at most* $7.6 \cdot opt + 1.2$.

Proof. The approximation factor of the TPSI algorithm is bounded by the size of the MIS M and the size of connectors C found during the first phase and second phase respectively. Based on Lemmas 5.15 and 5.16, the CDS size generated by the algorithm is given by $\leq M + C \leq 7.6 \cdot opt + 1.2$.

5.3.2 Performance of SPSI Algorithm

SPSI algorithm also has two-hop information range as in TPSI algorithm. To perform the connector computation and find the symmetrical neighbors, each node u needs to know the ID of its two-hop neighbors.

Theorem 5.18. *SPSI has* $O(3\Delta C + 3\Delta)$ *time complexity and* $O(n)$ *message complexity, where* n *is the total number of nodes in the network,* C *is the number of chosen connectors and* Δ *is the maximum node degree in the network.*

Proof. The time complexity of the SPSI algorithm is the time taken for computing the connector set in the network. Since the computation of connectors is based on the MPR algorithm [QVL02], thus the time complexity of the SPSI algorithm is $O(3\Delta C + 3\Delta)$, which can be proven similarly as in [LSM06].

The overall message complexity of SPSI is $O(n)$ since each node u exchanges exactly one message, either a *Dominatee Message* or a *Dominator Message* to build the CDS.

Theorem 5.19. *SPSI algorithm has an approximation ratio of at most* $O(log(\Delta))$, *where* Δ *is the maximum number of neighbors in the network. Under sparse networks, the approximation ratio is within a small constant factor.*

Proof. The SPSI algorithm uses the enhanced MPR rule discussed in Section 3.4.3 to find a connector set for building the backbone in a network. This process involves finding the two-hop neighbors covered by each node in the neighbor set. As proven in [Vie98], the algorithm that computes a connector set using this approach has an approximation ratio bounded by $log(\Delta)$, where Δ is the maximum number of neighbors in the network. Since, the SPSI algorithm uses the same approach as in [QVL02], it approximates the CDS size of at most $O(log(\Delta))$ times the CDS size of MCDS. Although it does not guarantee a small CDS size, the finding in [QVL02] shows otherwise. Under sparse networks, the approximation ratio is within a small constant factor of 4.7 [QVL02]. A sparse network here refers to a network that has a node degree below 4, in which node degree is defined as the average number of neighbors per node [Sto08]. This approximation ratio is comparable to the approximation ratio of the TPSI algorithm.

5.3.3 Performance of TPMI Algorithm

Since TPMI algorithm collects the ID of two-hop neighbors to find connector sets and symmetrical neighbors, its information range is also two hops.

Theorem 5.20. *TPMI has $O(3\Delta C + 3\Delta + n)$ time complexity and $O(n)$ message complexity.*

Proof. The overall time complexity of the TPMI algorithm is computed by the time taken for building the CDS construction. This process involves two phases, the DS and CDS generations. During the DS generation, the algorithm takes $O(n)$ to compute the MIS (or DS), as proven in [AWF02]. During the CDS generation, the time taken is measured by the connector selection process, which is based on the MPR concept in [QVL02]. Therefore its time complexity is $O(3\Delta C + 3\Delta)$, which can be derived from the proof in [LSM06].

This algorithm uses a constant number of messages when constructing a CDS. Each node sends exactly one message to its one-hop neighbors. The dominator sends a *Dominator Message* to inform its state and the chosen connector. The dominatee chosen as a connector also sends one message, the *Dominator Message* and broadcasts its selected connectors to neighbors. The uncovered node that has a smaller key than its neighbors sends one *Dominatee Message* to announce its change of state. Therefore, the TPMI algorithm has an overall $O(n)$ message complexity.

Lemma 5.21. *The size of a DS generated by the TPMI algorithm at the completion of the first phase is at most* $3.8 \cdot opt + 1.2$*, where opt is the size of a minimum CDS (MCDS) as deduced in [WDJ$^+$06].*

Proof. As shown in Lemma 5.11, the DS generated by the algorithm is the MIS of a graph. Thus, the size of the DS must not exceed $3.8 \cdot opt + 1.2$ as proven in [WDJ$^+$06].

Theorem 5.22. *TPMI algorithm has an approximation ratio of at most* $\leq 182.4 \cdot opt + 57.6$.

Proof. The size of the CDS generated by the TPMI algorithm is determined by the size of the DS and the number of connectors involved in connecting a pair of nodes in the DS. Let M denote the size of the DS. According to the proof in [AWF02], the total number of a pair of dominators is at most $\frac{47(M)}{2}$. Lemma 5.10 proves that in the worst case, there are at most two dominatees chosen as connectors in between a pair of dominators. Therefore, the total number of connectors separating two dominators is at most $47(M)$. Using Lemma 5.21, the approximation ratio of the TPMI algorithm is bounded by $\leq 48(M) \leq 182.4 \cdot opt + 57.6$.

5.4 Conclusions

In this chapter, the correctness of the proposed algorithms and their performances have been evaluated and summarized in Table 5.1. The theoretical analysis has indicated that the proposed algorithms correctly construct a CDS of a graph. The performance analysis on the other hand has shown the following beneficial features of the algorithms: (1) they guarantee a good CDS size as their approximation ratio is bounded by a constant ratio and (2) they have linear message and time complexities, thus they are efficient and easy to maintain especially under a dynamic network topology.

Algorithms	**Time Complexity**	**Message Complexity**	**Approximation Factor**
TPSI	$O(n)$	$O(n)$	$7.6 \cdot opt + 1.2$
SPSI	$O(3\Delta C + 3\Delta)$	$O(n)$	$O(log(\Delta))$ or 4.7
TPMI	$O(3\Delta C + 3\Delta + n)$	$O(n)$	$192 \cdot opt + 48$

Table 5.1: A summary of the performance analysis of the proposed algorithms. In the table, n represents the total number of nodes, Δ is the maximum number of neighbors in the network and opt is the size of the minimum CDS.

Although the SPSI algorithm does not always guarantee a small approximation ratio such as under dense networks, its implementation costs are lower than the costs of other existing approaches in [BGLA03, GK98, SDB98, WL99]. However, as the networks become less denser, its approximation ratio yields a better result which is bounded by a small constant value. Thus, its performance is comparable to the performance of the TPSI and TPMI algorithms. In the next chapter, the performance of the algorithms is further evaluated against several leading algorithms using a simulation based analysis to confirm whether they outperform the existing algorithms.

Chapter 6

Simulation of Proposed Algorithms

6.1 Introduction

The simulation results for the proposed algorithms are covered in this chapter. The algorithms have been implemented in simulation-based frameworks. These implementations were used to analyze the performance of the algorithms under various network conditions and to compare them with two leading algorithms surveyed in Chapter 2, which were also simulated as ”benchmarks“.

This chapter is organized as follow: Section 6.2 specifies the simulation framework and set-up scenarios for the algorithms. In Section 6.3, the metrics used for investigating the performance of the proposed algorithms are defined. The analysis of the findings is given in Section 6.4. Finally, the conclusion drawn from the results and highlights of the findings can be found in Section 6.5.

6.2 Simulation Set-up and Testing Environment

The details of the simulation set-up is given in this section. Section 6.2.1 explains the simulation tools while Section 6.2.2 presents the energy consumption model for the algorithms. Finally, Section 6.2.3 describes the network topologies generated for the performance analysis.

6.2.1 Simulation Framework

Since the goal of this chapter is to perform a realistic evaluation of the algorithms, a simulation framework, MiXiM [WSKW] was chosen for a number of reasons:

1. It can concisely model and directly support wireless communication.
2. It is capable of modeling the energy consumption of wireless devices.
3. It inherits the advantages of OMNeT++. Thus, it is freely available for academic purposes, highly modular, flexible, portable for Windows, Mac OS/X and Linux platforms, and there is extensive support for libraries and GUI for the visualization of graphs.

MiXiM requires the OMNeT++ [Var] simulation engine to run. Details of the MiXiM and OMNeT++ frameworks are described in Appendix C.

6.2.2 Energy Consumption Model

The energy consumption involved in the CDS construction was measured based on the energy dissipation of transmitting and receiving packets activities as modeled in [HCB00]. The transmitting operation is more costly than the receiving operation because it involves the electronic and amplifier parts, whereas, the receiving operation only involves the electronic part [CS04]. All nodes were initially assumed to have an identical energy reserve which was set to 0.01 mWs. Every time a node sends a packet, its transmit energy $E_{\text{TX}}(k, d)$ drops by

$$E_{\text{TX}}(k, d) = \delta k + \gamma k d^2, \tag{6.2.1}$$

where δ is the receiver circuitry constant assumed as 50 nJ/bit, γ is the transmit amplifier constant assumed as 100 pJ/bit/m^2, d is the transmit distance and k is the packet length assumed as 2000 bits long.

The energy consumption for receiving packets $E_{\text{RX}}(k)$ is computed by

$$E_{\text{RX}}(k) = \delta k. \tag{6.2.2}$$

6.2.3 Network Topologies

In the simulations, realistic topologies generated using the measurements obtained from an actual wireless testbed were used. Figure 6.1 shows the testbed which was set-up in a computer laboratory.

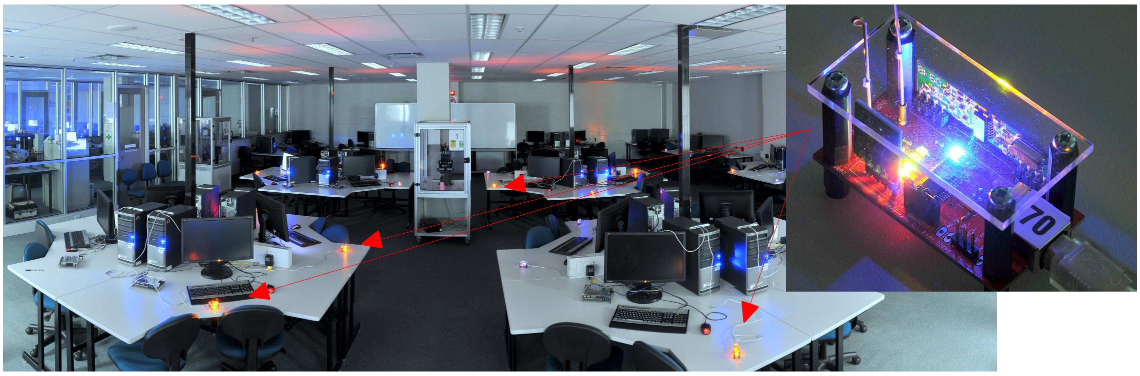


Figure 6.1: A packet radio network testbed consisting of 33 packet radio devices. The devices are cheap transmitter/receiver operating at the frequency of 433MHz ISM band and they were connected to a desktop computer via a serial connection.

Figure 6.2 is an example of a generated topology of a 100 nodes WSN which does not use a unit-disk graph (UDG) model. This topology has a variable transmission radius and the communication links are strongly affected by the quality of the signal reception. Interference is expected due to the presence of obstacles that may prevent nodes from receiving or sending packets. A total of 150 topologies were generated for the performance evaluation.

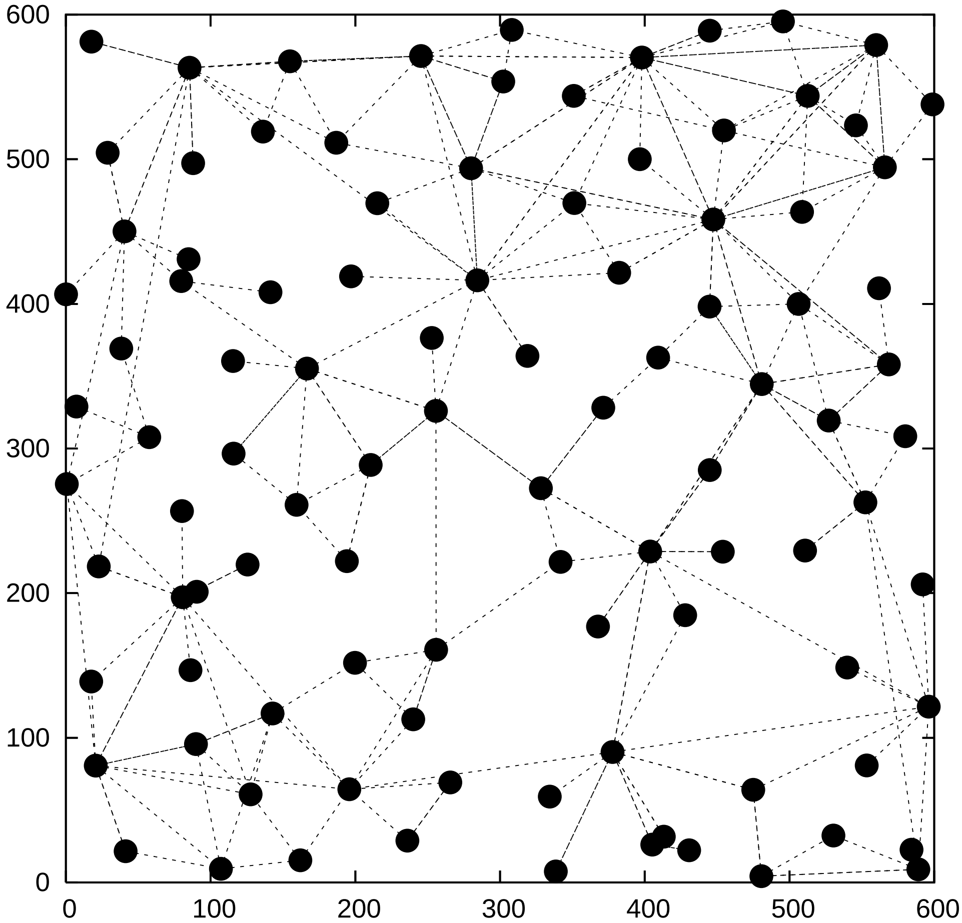


Figure 6.2: An example of a topology generated using the topology generator consisting of 100 nodes with an average number of 4 neighbors per node.

6.3 Performance Quality Metrics

The following metrics were used to investigate the efficiency of the proposed algorithms:

1. **Size of the CDS**. The size of the CDS represents the backbone size of a network. A small backbone is preferable over the large backbone for simplifying routing and avoiding high communication overhead. To measure this metric, the number of CDS nodes generated was recorded when the simulation terminated at 2000s.

2. **Message overhead**. This metric measures the message overhead involved for constructing the CDS. It affects the network performance in several ways. The more message overheads required during the CDS generation, the more nodes spend their energy and the shorter the lifetime of the network. To minimize the message overhead, the number of messages should be kept low and the length of messages should be kept short. Therefore, the message overhead was computed by multiplying the average number of messages sent with the message size (number of bytes in a message). The message overhead of each algorithm in this work was measured at the duration of 2000s.

3. **Energy consumption**. Minimizing the energy consumption can prolong the lifetime of the network. As mentioned in Section 6.2.2, the energy consumption here refers to the energy consumed during the transmit and receive packet activities. In this simulation, the energy consumed by these two activities for a duration of time was measured. The duration of time set was 2000s.

4. **Network lifetime**. The network lifetime in this simulation was defined by

the time when the first CDS node dies. It was assumed that the death of one CDS node was very critical to the network. As this node dies, the backbone was no longer connected and a new CDS must be reconstructed. The algorithm that can run longer is preferable over the one with the short run time. When the battery of a CDS node was completely depleted, the simulation stopped. The time when this event happened was recorded as the network lifetime.

5. **Convergence time**. This metric measures how much time an algorithm needs in order to construct a CDS. It reflects the time complexity of the algorithm. An algorithm that has a shorter convergence time is desirable as it can quickly reconstruct the CDS when the network topology changes. To measure the convergence time, the time taken for the algorithm to successfully build the CDS was recorded.

6. **Average route path length**. Since it is common for the CDS to be used for routing application, the average route path length over the subgraph, G_a formed by the CDS is measured. This metric is used to prove that the CDS generated by the proposed topology control algorithms can still preserve the shortest path length, thus making it reliable for the operation of any routing algorithm. To obtain the average route path length, Dijkstra's algorithm was run on the topologies of the CDS and the average shortest path length from a source to a target node was then recorded.

6.4 Analysis of the Results

SPSI, TPSI and TPMI algorithms were compared against the ECDS [YJY06] and PACDS [WGS01] algorithms surveyed in Chapter 2. Since the ECDS is the leading algorithm for generating a small CDS [KWL+09], it was selected for comparison.

Although PACDS does not guarantee a small CDS size, it has the ability to converge fast [SSK+08, YSS11]. The convergence here refers to the time taken by the PACDS to form a CDS. Hence, it served as a benchmark for analyzing the convergence speed of SPSI, TPSI and TPMI algorithms.

The performance of the proposed algorithms was evaluated in two aspects: network size and network density. The network size compares the performance of the algorithms in both small and large networks. It was used to determine the scalability of algorithms. The network density on the other hand, measures the performance of the algorithms as the number of neighbors at each node increases. In [Sto08], network density has been recommended as one of the independent variables for performance comparison due to its significant impact on the performance of algorithms. Despite its importance, it is rarely evaluated in many works [SSK+08, WGS01, WW03, YJY06, YWD09].

The network density is defined by the average number of neighbors per node or known as node degree. The types of networks (sparse, medium and dense) are classified according to the node degree value. Table 6.1 specifies the corresponding values for each network type defined in [Sto08]. As shown later in Section 6.4.1, a comparison made solely based on the network size without the consideration of the network density would have significant consequences on the performance results. For example, given a particular network density, one can always claim a superior performance. Therefore, in order to get a thorough analysis, the algorithms in this thesis were tested in sparse, medium and dense networks.

For simulations, 50 topologies of various network sizes were generated for each three network densities: sparse, medium and dense networks. Hence, a total of 150 topologies were used. The number of nodes N in the topologies varied from 100 to 500 with an interval of 100. Ten runs of different seeds were created for each N. The nodes were deployed in a two-dimensional (2D) space and their transmission radius was not fixed. The parameters used in the simulations are given in Table 6.1.

Network Type	**Sparse**	**Medium**	**Dense**
Network density (node degree) [Sto08]	4– 6	8–10	12
Deployment area (m^2)	600-1341		
Number of nodes	100, 200, 300, 400 and 500		
Number of seeds per run	10		
Total number of runs per network type	50		
Node initial energy (mWs)	0.01		
MAC protocol	CSMA/MAC		
Physical layer model	simple path loss		
Path loss exponent	3		

Table 6.1: The parameters.

The remaining parts of this section report the simulation results obtained and analyze the results based on the metrics discussed in Section 6.3.

6.4.1 Evaluation of the CDS Size

The main goal of this simulation is to compare the CDS size formed by the algorithms. As mentioned before, the idea is to keep the CDS size as small as possible to minimize the communication overhead. For this comparison, two parameters, the network size and the network density were used.

Figure 6.3 shows the CDS size generated plotted against the two parameters. When the network size increases from 100 to 500, the number of CDS nodes of all algorithms also increases in a linear fashion. This is expected as large networks will require more number of CDS nodes to cover neighbors that are not in the CDS.

From Figure 6.3, the network density has a significant impact on all algorithms except for TPMI. It is clear that the SPSI, TPSI and ECDS are best for sparse, medium

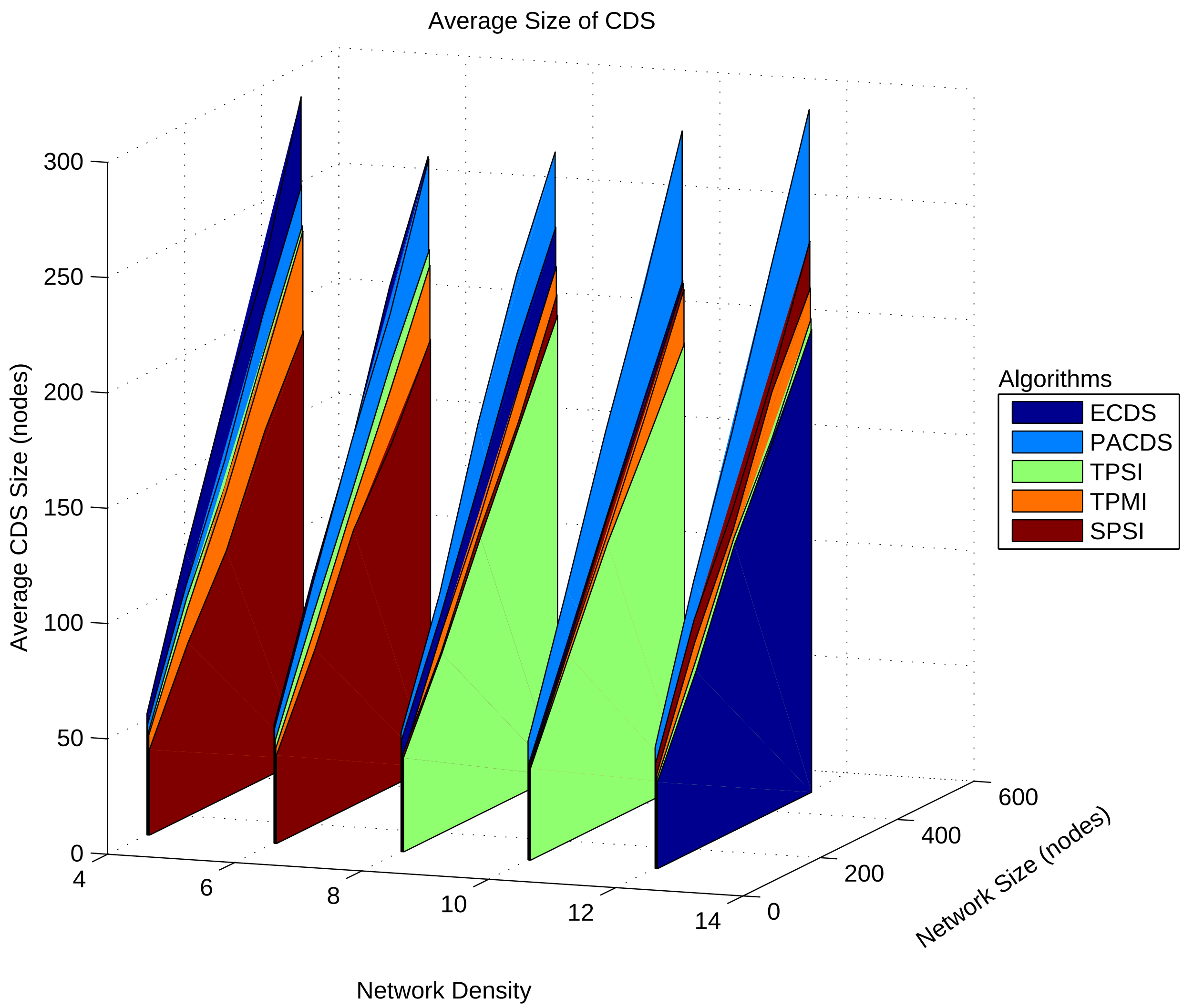


Figure 6.3: Average CDS size versus network density and network size of the proposed and leading algorithms. Network density of 4 and 6 represents sparse networks, 8 and 10 represent medium networks and 12 represents dense network.

and dense networks respectively because they created the minimum number of CDS.

An interesting finding was that the excellent performance of the ECDS claimed in [KWL+09, YJY06] is only applicable to dense networks. In contrast, TPMI's performance on the other hand was not affected by the network density. Its CDS size remains quite constant throughout all network densities. This suggests that it is more flexible to be used in any type of networks regardless of the network density. The constant performance of the TPMI can be explained by the use of multiple initiators over a single initiator when finding dominators. Because these dominators were fully distributed over the network, TPMI will consistently find those with the largest neighbor coverage. However, in the single initiator approach, this similar process was performed from one single point of view. Depending on which node was chosen as the initiator and the selection metric used for the initiator, the random selection of dominators influenced the CDS size of the network.

TPSI and ECDS both had a same trend, in which their CDS size decreased with the network densities. They produced a comparative number of CDS nodes in dense networks. This is contributed by the similarities in their CDS generation process and the CDS selection metric. It is also obvious that the benefit of using the node degree as the selection metric for choosing CDS nodes was only apparent in dense networks. The reason is that as the network becomes denser, more neighbors are covered by the CDS nodes, thus less number of CDS nodes are formed. It is also observed that in sparse and medium networks, the CDS size of the TPSI was lower than the one in the ECDS due to the pruning process. However, this pruning process had no significant impact on dense networks due to the lack of nodes to be pruned. Recall that the pruning process only prunes nodes which have one neighbor. In dense networks, nodes in the network mostly have more than one neighbors, thus there is only a few of nodes available for pruning.

In contrast to the TPSI and ECDS, the CDS size of the PACDS and SPSI kept increasing as the network density increased, with PACDS consistently forming the highest CDS size in almost all network densities. The result of the PACDS is predictable since PACDS generally creates a large number of CDS during the first phase because it uses the rule of the 'two uncovered nodes' for selecting CDS nodes. This rule did not take into account the node density, thus was less efficient in reducing the CDS size. This explains why PACDS introduced the second phase to prune the redundant CDS nodes. The reason for the increased CDS size in the SPSI with respect to the network density was contributed by the increment in the number of connectors chosen when using the MPR rule.

6.4.2 Evaluation of the Message Overhead

Figure 6.4 presents the message overhead with respect to the network density and size.

The proposed algorithms TPSI, SPSI and TPMI required a significantly low message overhead in comparison to the ECDS and PACDS, with the TPMI having the least message overhead during the CDS construction. The great performance of the proposed algorithms was obvious in the dense network. In the case of node degree 12, the SPSI, TPSI and TPMI reduced the message overhead of the ECDS up to 75%, 87% and 96% accordingly. Compared to the message overhead of the PACDS, the saving made by the SPSI, TPSI and TPMI was 71%, 84% and 95% respectively.

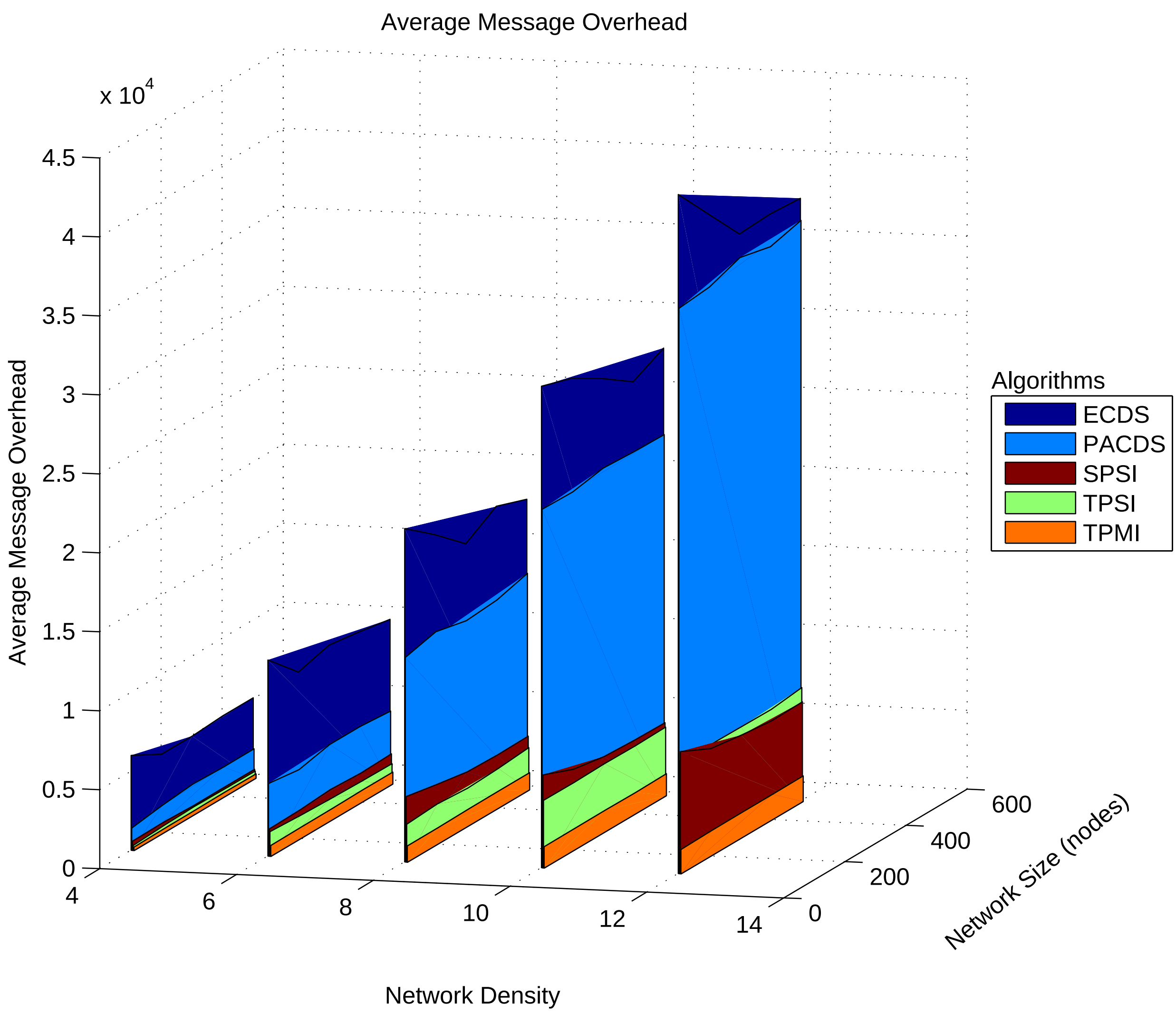


Figure 6.4: Average message overhead of five algorithms plotted against the network density and network size. Network density of 4 and 6 represent sparse networks, 8 and 10 represent medium networks and 12 represents dense network.

The high message overhead in the ECDS is related to the frequent number of messages sent when nodes are competing for the CDS election. As the network density or size increases, the number of neighbors and nodes also increases. Thus, more nodes will be involved in the negotiation process to decide which nodes should be chosen as the CDS. The large number of messages required by the ECDS found in this section was verified against the number of messages measured in [WL08]. They both have a similar pattern. In the PACDS, the substantial message overhead incurred is also due to the large number of messages broadcast during both phases of the CDS generation. Although PACDS uses localized information, the pruning process is costly. Every time a node is eliminated from the CDS, it has to inform its current state to neighbors that are two hops away. To validate the message overhead of the PACDS, the number of messages of the PACDS recorded in [SSK+08, WL08] was compared to the number of messages of the PACDS measured in this section.

The good performance of the TPSI, SPSI and TPMI was due to two reasons. Firstly, these algorithms eliminate the negotiation process and based on the local information, they decide whether they are eligible for joining the CDS. Secondly, the number of exchanged messages that they use was bounded by a constant number. Nodes do not send messages unless it is necessary for example when updating node status. ECDS nevertheless requires each node to regularly send messages containing the weight information for deciding MIS and CDS nodes.

From Figure 6.4, the TPMI performs better than the TPSI and SPSI. In dense networks, its message overhead is approximately 71% and 85% lower than the message overhead of the TPSI and SPSI, respectively. This implies that the multiple initiator approach is more efficient than the single initiator approach whereby the process of finding the CDS is fully distributed with only a few nodes involved in the message transmission. Contrary to the TPMI, the CDS generation process of the TPSI and SPSI had a chain reaction, where changes in one node will trigger changes in the subsequent nodes. This contributes to the large message overhead as the number of nodes sending messages is also large.

SPSI in general has a higher message overhead than the TPSI except in dense networks. This is due to its large signaling message size. However, as the network became denser as in node degree 12, the message overhead of the TPSI was 12% higher than in SPSI. This can be explained by the increase in the number of messages sent in both phases of the CDS construction in the case of TPSI.

6.4.3 Evaluation of the Energy Consumption

Similar to Section 6.4.1 and Section 6.4.2, the energy consumption of the algorithms was analyzed with respect to the network density and size. Since the energy consumption was determined based on the energy spent for transmitting and receiving messages, it is influenced by the average number of messages used by the algorithms.

The result of Figure 6.5 indicates that the energy consumption of all algorithms remains constant with the network size.

The ECDS has the largest energy consumption followed by the PACDS. In a dense network of node degree 12, the energy spent by TPSI, SPSI and TPMI are respectively 82%, 95% and 96% less than the energy spent by the ECDS. Whereas TPSI, SPSI and TPMI consume 71%, 91% and 96% respectively less energy than in PACDS. The high energy consumption of the ECDS is due to the high message generation during the construction of the CDS that was based on a negotiation. As mentioned before in Section 6.4.2, the self-pruning process and the less efficient rule adopted for the CDS selection were the reasons for the large message created in the PACDS, which resulted in high energy consumption.

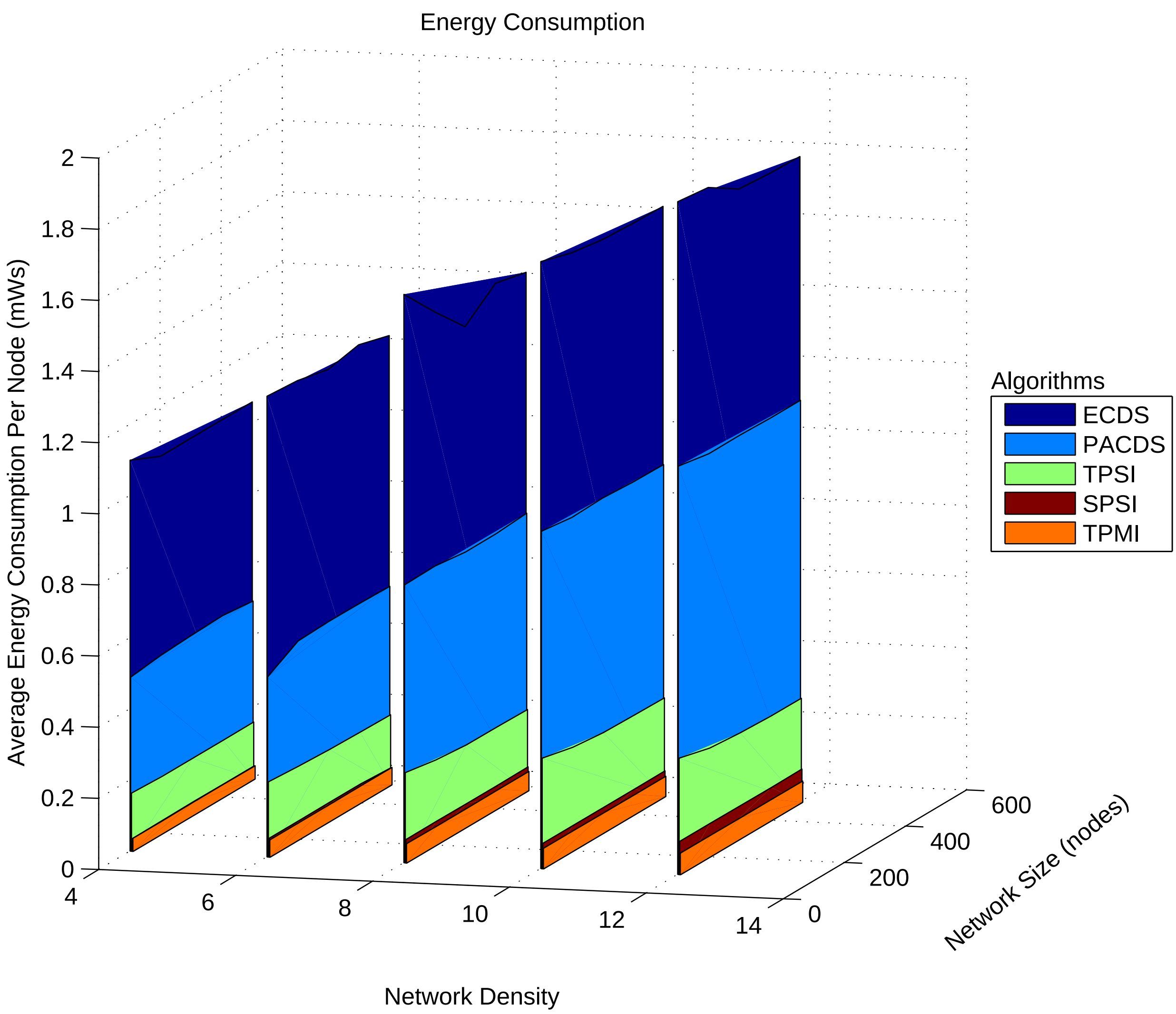


Figure 6.5: Average energy consumption of five algorithms plotted against the network density and network size. Network density of 4 and 6 represent sparse networks, 8 and 10 represent medium networks and 12 represents dense network.

The effectiveness of the TPSI, SPSI and TPMI in conserving energy consumption was due to the low number of messages sent during the CDS construction. They were designed to limit the number of messages sent and only send a constant number of messages as proven in Chapter 5.

6.4.4 Evaluation of the Network Lifetime

The network lifetime comparison among the algorithms with respect to the network density and size is presented in Figure 6.6. The network lifetime was measured based on how long the algorithm can run before the network can no longer form a CDS. The algorithm stops running when the energy level of the first CDS node becomes zero.

As shown in this figure, the network lifetime decreased as the network density increased and remained constant with increasing network size except for SPSI and TPSI. This indicates that more energy was spent in dense networks. TPMI had a longer lifetime followed by the SPSI and TPSI contributed by their low energy consumption as demonstrated in Figure 6.5. As predicted, the PACDS and ECDS both had a shorter lifetime, with the ECDS having the shortest network lifetime due to its significant amount of energy consumption.

Clearly, the TPMI, SPSI and TPSI algorithms outperformed the performance of the PACDS and ECDS because they consumed a minimum energy and utilized a low message overhead. The network lifetime of the ECDS are respectively 96%, 95% and 71% lower than the network lifetime of the TPMI, SPSI and TPSI.

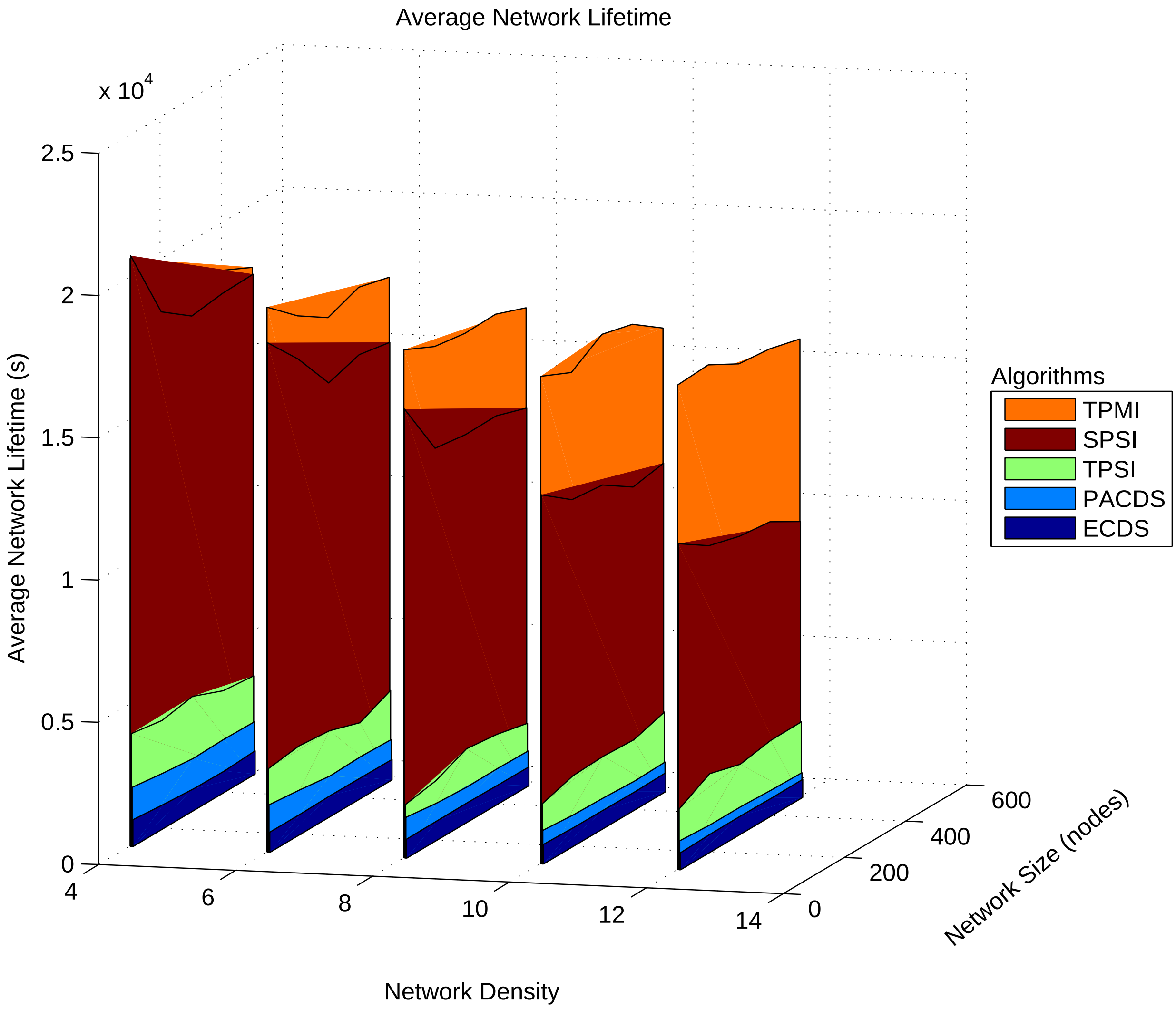


Figure 6.6: Network lifetime of five algorithms plotted against the network density and network size. Network density of 4 and 6 represent sparse networks, 8 and 10 represent medium networks and 12 represents dense network.

6.4.5 Evaluation of the Convergence Time

Figure 6.7 illustrates the relationship of the convergence time with the network density and size.

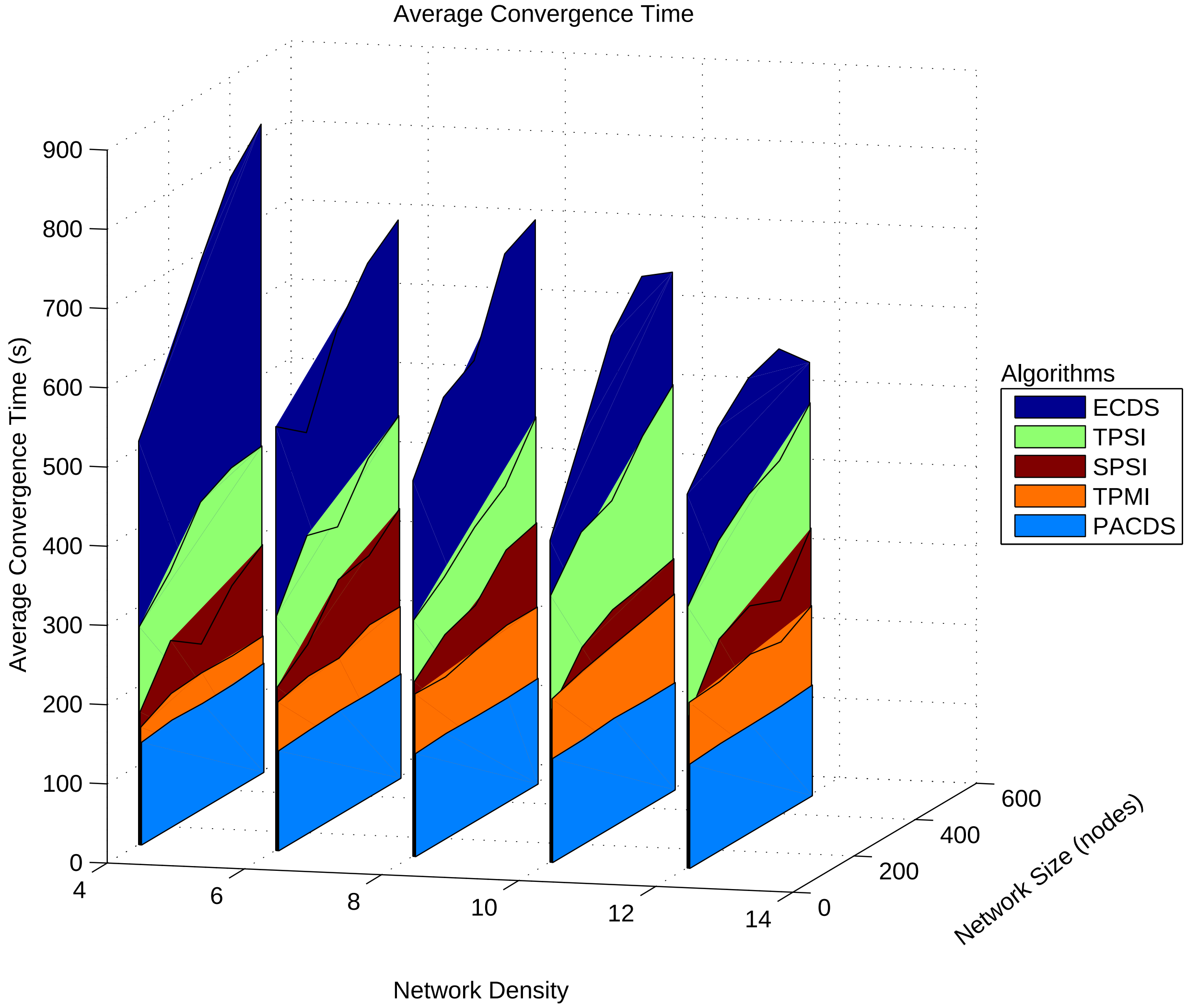


Figure 6.7: Average convergence time of five algorithms plotted against the network density and network size. Network density of 4 and 6 represent sparse networks, 8 and 10 represent medium networks and 12 represents dense network.

Apart from the PACDS and TPMI, all algorithms took a longer time to converge as the network size increased. This shows that more time was required for searching the CDS nodes when the network became larger and denser. As for the PACDS, it took the least time to build a CDS due to its distributed implementation and the simplicity in its construction. Interestingly, its convergence time was constant as the network density increased. This result confirms its superior performance in computing a CDS. The TPMI also had a relatively good performance in comparison to the PACDS where its convergence time was almost the same in any network density. It outperformed SPSI, TPSI and ECDS due to the use of distributed multiple initiators.

The ECDS on the other hand was consistently producing the largest convergence time regardless of the network size. This was due to the additional time spent on finding neighbors with the largest weight, where nodes needed to update their neighbors on their latest weight. As predicted, the TPSI required a longer time to build the CDS than the SPSI because of its extra phase used in the CDS generation process.

As the network density increased, the ECDS's performance improved. This can be explained by the fact that as nodes have more neighbors, less time is spent on computing the CDS because these neighbors are already covered by some other nodes, possibly a CDS. This behavior was not observed in the SPSI and TPSI, in which they needed similar duration to compute the CDS and were less affected by the network density. Thus, they are more robust.

6.4.6 Evaluation of the Average Route Path Length

In routing protocols, the route path length or known as the shortest path length is an important characteristic of networks. Due to the common use of the CDS in routing, the route path length property was measured on the CDS. Ideally, this property should be preserved in the induced sub-graph generated by the CDS algorithms. In this thesis, the average route path length in both the initial graph of the network and sub-graph of the network created by the proposed algorithms were measured and compared.

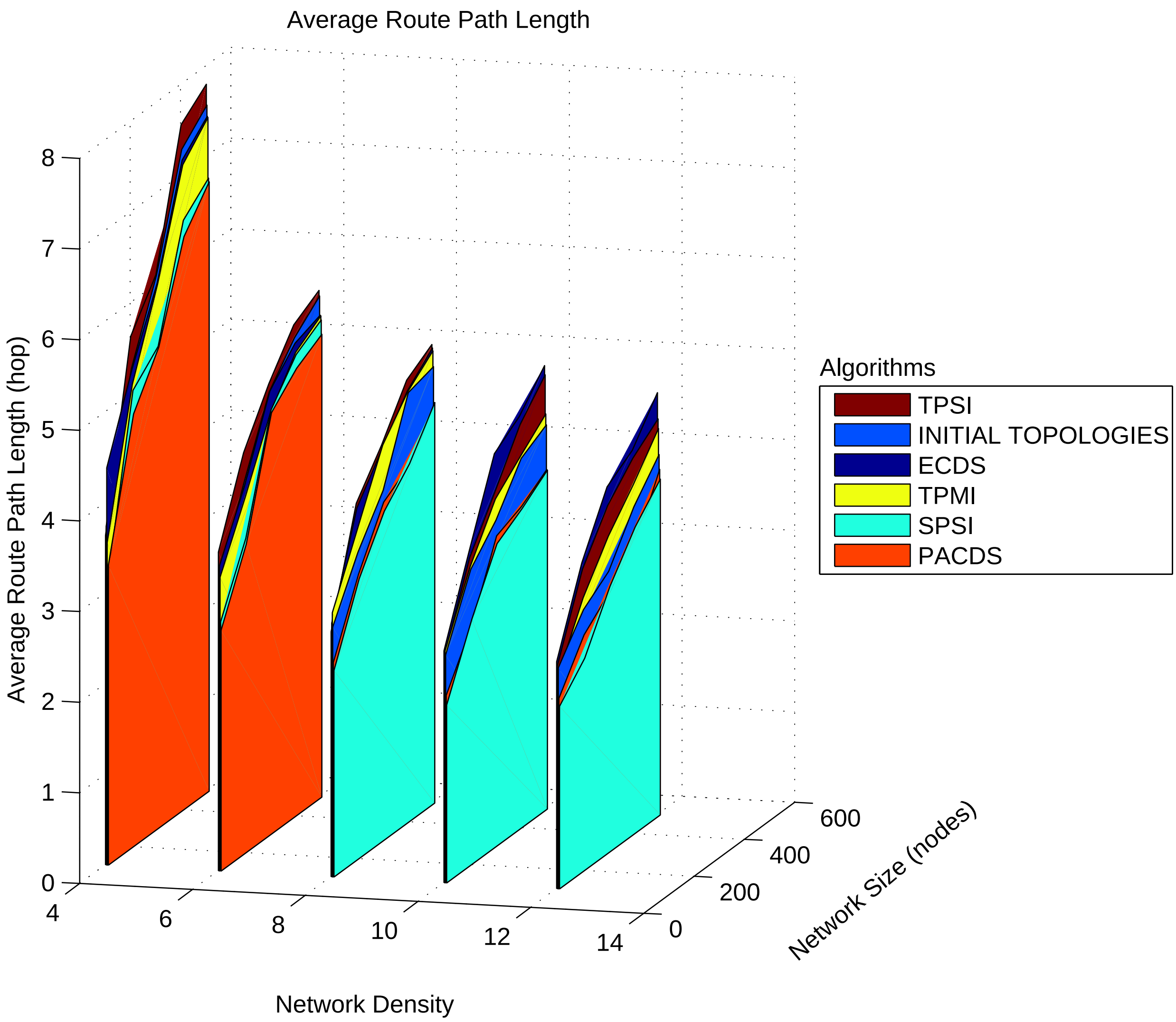


Figure 6.8: Average route path length of the network is preserved even after the application of the topology control. Network density of 4 and 6 represent sparse networks, 8 and 10 represent medium networks and 12 represents dense network.

The average route path length of initial topologies depicted in Figure 6.8 represents the route path length measured over the initial graph of the network. Detailed description on the initial topologies can be found in Section 6.2.3. Whereas the route path length of the five algorithms is the path length calculated over the CDS topologies formed by the algorithms. These CDS topologies represent the sub-graph of the network. To obtain the average route path length from a source to a target node, Dijkstra's algorithm implemented in OMNeT++ was run on initial and CDS topologies.

Figure 6.8 shows that the route path length did exist in the CDS, confirming the fact that all the proposed algorithms were capable of preserving this property. It is evident that the average route path length decreases with the increase in network density as there are more possible shorter routes available in the network. A significant improvement in the shortest path length with respect to the network density was produced by the SPSI followed by the PACDS, TPMI, ECDS and TPSI. An increase in the average route path length with respect to the increase in network size is expected due to the fact that nodes had to traverse longer to reach the target node via their CDS nodes.

6.5 Conclusions

This chapter has presented and discussed the simulation results of the proposed algorithms. These algorithms were evaluated against two competing algorithms to obtain their performance comparison. Apart from the network size, the network density parameter which was often ignored in previous studies was adopted for the performance evaluation. The simulation findings showed that the network density plays an important role in evaluating the performance of the algorithms. The proposed algorithms had successfully ensured that the route path length property measured over the constructed CDS was maintained, thus making them suitable for routing application.

The results of the simulations indicated that the TPSI, SPSI and TPMI can build a CDS quickly by using only minimum energy resources and low communication overhead. These features contribute to the network lifetime extension. With respect to the CDS size, it can be concluded that SPSI is best deployed for sparse networks, whereas, TPSI is best suited for medium and dense networks. On the other hand, TPMI is efficient for minimizing the message overhead and energy consumption, and extending the network lifetime. This demonstrates that the proposed algorithms in the thesis offer solutions for wider applications. They can be used to complement the existing CDS algorithms which are typically excellent in dense networks but perform poorly in sparse and medium networks.

Chapter 7

Concluding Remarks and Future Directions

7.1 Conclusions

As witnessed by the growth in the number of research publications to date, topology control has become one of the important strategies for conserving the power resources of nodes and handling the dynamic issues of networks. The survey conducted in Chapter 2 showed that among the techniques used for exercising topology control, a CDS offered several advantages. The survey revealed that the current CDS techniques were not efficient. The existing techniques use simplified network models, do not consider the network density for performance evaluation and require a significant number of message exchanges to build the CDS, which could adversely affect the network lifetime.

This thesis offers three distributed and localized CDS techniques that are efficient for WSNs. These techniques: (1) eliminate the use of an idealized graph such as UDG, (2) adopt a more realistic link connectivity model that considers the existence of unidirectional links, (3) take into account the interference between nodes, (4) employ a basic link layer protocol that handles packet retransmissions and (5) utilize realistic node distribution through the use of network topologies that are not uniformly distributed over an area and have variable transmission ranges. The algorithms were implemented in the simulation-based framework MiXiM, which runs on OMNeT++ engine with the core functionality written in C++. The efficiency of the proposed algorithms were analyzed against the two competing CDS algorithms ECDS and PACDS in two important aspects: network density and network size.

In Chapter 3, two distributed CDS techniques TPSI and SPSI that were based on a single initiator construction were proposed. Due to the importance of extending the network lifetime, they were designed in a distributed fashion using only localized information to create a small CDS and reduce the communication overhead. The theoretical analysis conducted in Chapter 5 and the simulation results of Chapter 6 show that both TPSI and SPSI have low computational overheads with linear time and message complexities compared to the two leading algorithms. From the simulation results, the benefit of the SPSI is apparent in sparse networks while the TPSI is more efficient for medium and sparse networks.

Chapter 4 proposed the TPMI algorithm which is capable of handling frequent topology link changes in the network. This is achieved through the use of multiple initiators for the CDS construction. Similar to the objectives of the TPSI and SPSI, the TPMI aims to create a small number of CDS while reducing the message overhead of the network, which then leads to energy-efficiency and network lifetime extension. The theoretical analysis was performed on the TPMI in Chapter 5 to verify its time and message complexities as well its approximation factor, showing its bounded CDS size. Simulation results in Chapter 6 show that the TPMI generates a relatively constant CDS size independent of network density. It is capable of creating a smaller CDS size compared to the leading algorithm ECDS in sparse and medium networks and always generates a lower CDS size than the one in PACDS. Although TPMI does not create the smallest size of a CDS as in TPSI and SPSI, it is very efficient in minimizing the message overhead, reducing the energy resources and prolonging the network longevity.

7.2 Summary of Contributions

The contributions of this thesis in the area of topology control can be summarized as follows:

- A detailed survey on topology control providing an overview of various existing techniques and their classification [ASFI12].
- Three CDS algorithms TPSI, SPSI and TPMI were proposed for the realistic implementation of WSNs. They outperformed the two leading CDS algorithms implemented in this thesis in terms of the CDS size, message overhead, energy consumption, network lifetime and convergence time. SPSI in particular is excellent for sparse networks [AS12a] and TPSI is best suited for the medium and dense networks [AS12b]. TPMI on the other hand is an energy-efficient and fast convergence algorithm.

Figures 7.1, 7.2 and 7.3 provide the performance comparison of TPSI, SPSI and TPMI with the two leading algorithms PACDS and ECDS as benchmarks.

7.3 Future Directions

There are various issues that require further investigation within this research topic. Recently, there has been growing interest in algorithms for non-planar topologies such as deployments in underwater environments or multi-level buildings. These topologies are modeled in three-dimensional (3D) space rather than

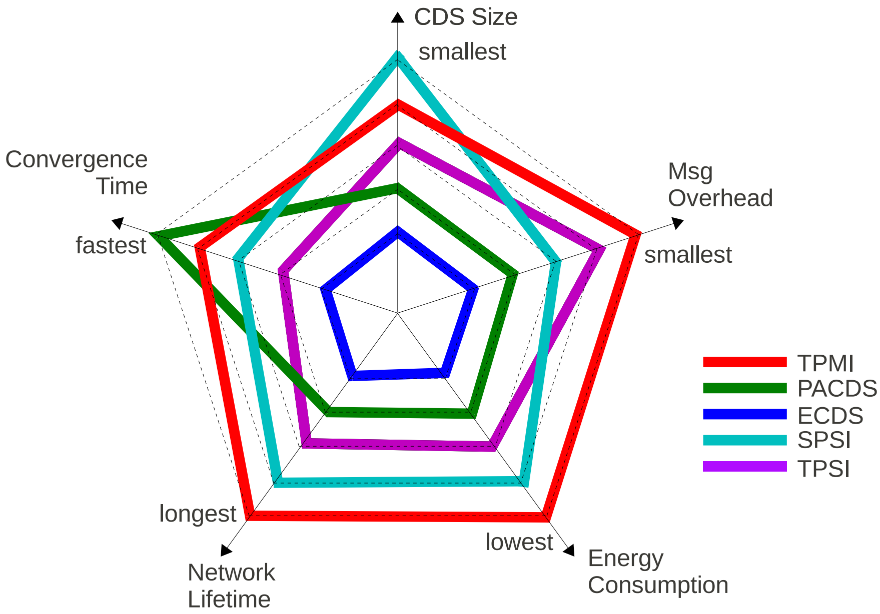


Figure 7.1: Comparative performance of the algorithms in sparse networks (average node degree is less than 6 [Sto08]).

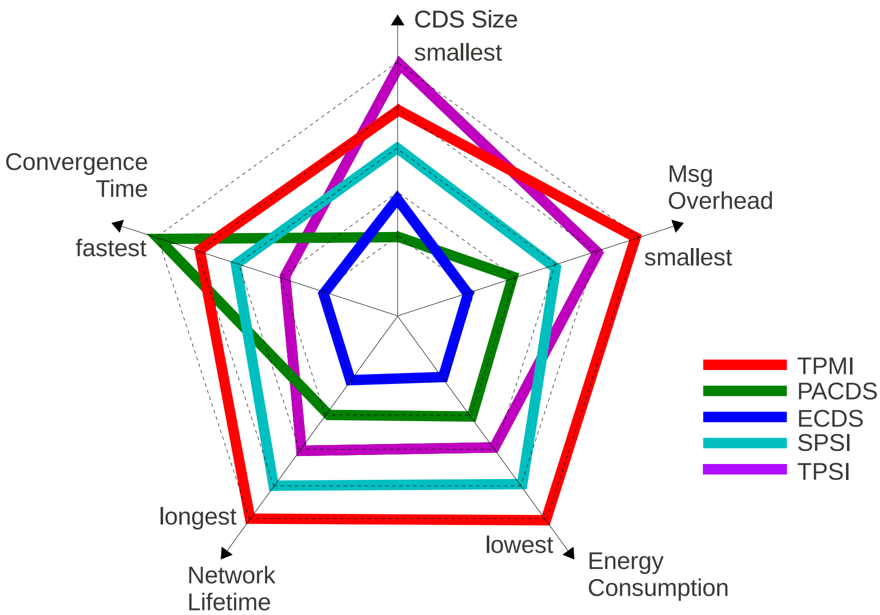


Figure 7.2: Comparative performance of the algorithms in medium networks (average node degree is between 8 and 10 [Sto08]).

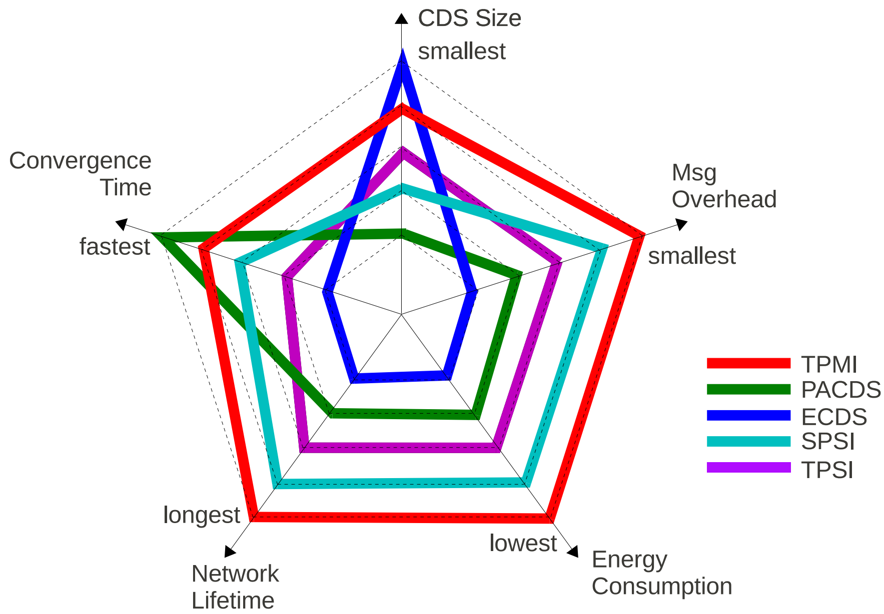


Figure 7.3: Comparative performance of the algorithms in dense networks (average node degree is above 12 [Sto08]).

two-dimensional (2D) space. The proposed algorithms can be extended to 3D topologies and their performance can be investigated.

There have been a number of WSN applications that assume a sink or sensor nodes to be mobile such as in battlefields. Therefore, it is important to evaluate the impact of mobility on the performance of the algorithms, particularly with respect to the message overhead, energy consumption and network lifetime. A mobility model that reflects the actual users' movement should be considered over the random waypoint mobility model.

One of the common applications of the CDS is broadcasting. The proposed algorithms can be implemented to support routing and their impact on various routing protocols can be studied.

Appendix A

Graph Theory Definitions

The chapter introduces some definitions from graph theory for modeling the WSNs.

Definition A.1. *Symmetrical Links*

A graph G is said to exhibit symmetrical links if node u is able to send messages to node v as well as receiving messages from v with no interference. This property is also known as bidirectional links. Symmetrical links are assumed in the network since routing and connectivity both rely on them.

Definition A.2. *Connectivity*

A communication graph G is connected if two vertices u and v are joined by at least one edge (u, v) in the G.

Definition A.3. *Node Degree*

A node degree $ND(v)$ is defined as the maximum number of edges at node v, assuming that the edges are symmetrical. It is also equivalent to the number of neighbors of node v.

Definition A.4. *Independent Set*

An independent set (IS) of $G = (V, E)$ is a set of $S \subseteq V$ such that no two nodes are adjacent in the set.

Definition A.5. *Maximal Independent Set*

A maximal independent set (MIS) of $G = (V, E)$ is an independent set that is not a subset of any other independent set such that adding the vertex not in the set violates the independence property of the set.

Definition A.6. *Dominating Set*

A dominating set (DS) of $G = (V, E)$ is a subset of V such that each node not in the set is adjacent to at least one node in the set.

Definition A.7. *Connected Dominating Set*

A connected dominating set (CDS) of $G = (V, E)$ is a dominating set of G which connects the dominating set to form a connected subgraph of G.

Appendix B

Symmetrical Neighbor Discovery and Maintenance Mechanism

A symmetrical neighbor discovery and maintenance mechanism is responsible for discovering symmetrical neighbors and updating neighbor information in the network. It ensures up-to-date neighbor information is used during the construction of a CDS.

This mechanism consists of two phases: neighbor discovery and neighbor maintenance. The first phase finds symmetrical neighbors while the second phase regularly updates the neighbor information.

B.1 Neighbor Discovery

Each node stores two types of neighbors: (1) symmetrical neighbors and (2) broken neighbors. The symmetrical neighbors are nodes with symmetrical links whereas the broken neighbors are nodes with asymmetrical links. Each node u uses the rule described by Algorithm 10 to discover its symmetrical and asymmetrical neighbors. The symmetrical neighbors of node u are considered as neighbors and they are adopted for computing the CDS but the broken neighbors are not the neighbors of node u. Node u monitors the incoming number of beacon packets N_{pkt} from its neighbors v for a duration of time $T_{arrival}$. Two conditions specified in step 2 and step 3 of the rule are used to find the neighbors. Node u considers node v as its symmetrical neighbor if: (1) N_{pkt} of node v exceeds the $N_{threshold}$, which is set to the value of 3 and (2) node u is the potential neighbor of node v.

Algorithm 10 Neighbor Reception Rule.

1. Record the number of consecutive beacon packets, N_{pkt} received within a timeout period, $T_{arrival}$.
2. If node u has $N_{pkt} > N_{threshold}$ during $T_{arrival}$, it is a potential neighbor. Otherwise, reset the timeout counter and continue listening for incoming packets.
3. For every potential neighbor of node u, if neighbor v has node u in its potential neighbor list, it is added to the symmetrical neighbor list. Otherwise, it is added to the broken neighbor list of node u. The symmetrical neighbors are the neighbors of node u.

B.2 Neighbor Maintenance

A neighbor maintenance process maintains and updates the neighbor information of nodes. Node u broadcasts its neighbor information to all neighbors when the state of neighbors or the list of neighbors changes.

Appendix C

MiXiM and OMNeT++ Simulation Frameworks

The simulation frameworks, OMNeT++ Version 4.1 and MiXiM Version 1.2 running on Linux platform were used for simulations. Section C.1 briefly provides the overview of OMNeT++ and Section C.2 describes the architecture of MiXiM.

C.1 OMNeT++ Framework

OMNeT++ [Var] is an open source discrete event simulation tool written in C++ programming language. It is used for modeling communication networks, multiprocessors and other distributed or parallel systems [VH08]. OMNeT++ provides basic tools and libraries that allow users to build a large-scale network of hierarchical structure. In order to support larger applications, OMNet++ is designed to be as general as possible. To suit specific applications, users can choose simulation models and frameworks that have been independently developed. MiXiM [WSKW] is one of the supported frameworks. The list of updated frameworks and simulation models is available in [Var].

C.2 MiXiM Framework

MiXiM is an OMNeT++ modeling framework that merges several existing simulation frameworks to support detailed models and protocols for mobile or static wireless networks. Details of the relevant frameworks can be found in [WSKW]. It offers accurate lower layer models of the protocol stack to realistically simulate the behaviors of the wireless networks. The wireless channel effects of the physical layer, namely signal fading and attenuation can be implemented and several MAC protocols such as CSMA, 802.11 and 802.15.4 can be chosen. The energy consumption activities of various devices such as radio, sensor nodes and CPU, and the host failure state are also provided. This energy model is ported from the Energy Framework [Fee] and was used for modeling the energy consumption of the algorithms described in Section 6.2.2.

To create a network in MiXiM, three components must be present:

1. Network architecture. It is built using modules. A larger module (a WSN) can be formed by combining several simple modules (sensor nodes).

2. Network functionality. The functionality of the modules is programmed in C++.

3. A configuration file. This file specifies the simulation parameters for the performance analysis.

Each node has a basic structure almost similar to the OSI/OSI protocol stack as shown in Figure C.1. The structure consists of three protocol layers and three other auxiliary modules. The modules are:

- **appl** – Application layer that sends broadcast packets.

- **net** – Network layer that performs routing protocols.

- **nic** – Network interface card implements physical and media access control (MAC) layers. The MAC layer in this simulation uses the CSMA/MAC protocol and the physical layer adopts a simple path loss model.
- **utility** – Blackboard module handles the communication between two nodes and provides a general interface for collecting statistical data.
- **arp** – Address resolution protocol maps IP addresses to MAC addresses.
- **mobility** – Mobility module provides mobility patterns and handles nodes' location and movement.

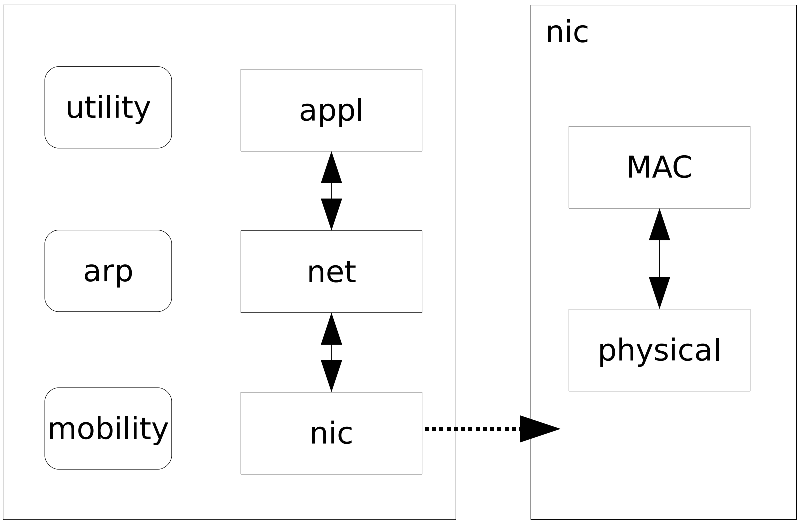


Figure C.1: Basic structure of sensor nodes in MiXiM.

The nic module tightly couples the physical and MAC layers as depicted in Figure C.1. It models the wireless channel effects and connectivity between nodes. Two nodes are connected only if they are located within a certain distance called *maximal interference distance* calculated by

$$\textit{maximal interference distance}(\mathrm{m}) = \left(\frac{\lambda^2 P_{\mathrm{Tmax}}}{(4\pi)^2 P_{\mathrm{Rmin}}}\right)^{\left(\frac{1}{\alpha}\right)}, \qquad \text{(C.2.1)}$$

where λ is the wavelength in meter, P_{Tmax} is the maximum transmit power of a node in mW, P_{Rmin} is the minimum receiving power threshold for a packet and α is the path loss coefficient of value 3.

It is worth mentioning that the packets sent between two connected nodes may not always be correctly received by receivers due to signal interference or fading effects. Hence, the physical layer models are responsible for deciding the successful reception of the incoming packets. They compute the interference level and bit error rates of the received packets to decide whether the packets are correctly received. If the signal strength of the received packets is below than a predefined threshold, the packets are discarded and then they will be scheduled for retransmission. Else, the packets will be further sent to the upper layers.

Bibliography

[200S8] Sensors Magazine 2004. Editorial: this changes everything-market observers quantify the rapid escalation of wireless sensing and explain its effects. Wireless for Industry, Supplement to Sensor Magazines, Summer Issue, pages S6-S8.

[802] IEEE 802.15. IEEE 802.15.4: a physical layer and media access standard for low-rate wireless personal area networks. `http://www.ieee802.org/15/pub/TG4.html`.

[AIM10] L. Atzori, A. Iera, and G. Morabito. The internet of things: A survey. *Computer Networks*, 54(15):2787–2805, 2010.

[All] Zigbee Alliance. Zigbee: a low-cost, low-power, wireless mesh network standard. `http://www.zigbee.org`.

[AS12a] A. A. Aziz and Y. A. Sekercioglu. A distributed energy aware connected dominating set technique for wireless sensor networks. In *Proceedings of the 4th International Conference on Intelligent and Advanced Systems (ICIAS 2012)*, Kuala Lumpur, Malaysia, June 2012.

[AS12b] A. A. Aziz and Y. A. Sekercioglu. A distributed topology control method for improving energy efficiency of wireless sensor networks.

In *Proceedings of the 4th International Conference on Intelligent and Advanced Systems (ICIAS 2012)*, Kuala Lumpur, Malaysia, June 2012.

[ASFI12] A. A. Aziz, Y. A. Sekercioglu, P. Fitzpatrick, and M. Ivanovich. A survey on distributed topology control techniques for extending the lifetime of battery powered wireless sensor networks. *IEEE Communications Surveys and Tutorials*, PP(99):1–24, 2012.

[ASSC02a] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, 2002.

[ASSC02b] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, 2002.

[Awe87] B. Awerbuch. Optimal distributed algorithms for minimum weight spanning tree, counting, leader election, and related problems. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing (STOC 1987)*, pages 230–240, New York, NY, USA, May 1987.

[AWF02] K. M. Alzoubi, P. J. Wan, and O. Frieder. Message-optimal connected dominating sets in mobile ad hoc networks. In *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc 2002)*, pages 157–164, Lausanne, Switzerland, June 2002.

[BCDP03] S. Butenko, X. Cheng, D. Z. Du, and P. M. Pardalos. On the construction of virtual backbone for ad hoc wireless networks. *Cooperative Control: Models, Applications and Algorithms*, 1:3–43, 2003.

[BDTC04] J. Blum, M. Ding, A. Thaeler, and X. Cheng. Connected dominating set in sensor networks and MANETs. *Handbook of Combinatorial Optimization*, pages 329–369, 2004.

[BF54] S. Blumsack and A. Fernandez. Ready or not, here comes the smart grid! *Energy*, 2011, doi:10.1016/j.energy.2011.07.054.

[BGLA03] L. Bao and J. J. Garcia-Luna-Aceves. Topology management in ad hoc networks. In *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc 2003)*, pages 129–140, New York, NY, USA, June 2003.

[BJ02] S. A. Borbash and E. H. Jennings. Distributed topology control algorithm for multihop wireless networks. In *Proceedings of 2002 World Congress on Computational Intelligence (WCCI 2002)*, pages 355–360, Honolulu, HI, USA, May 2002.

[Bou09] A. Boukerche. *Algorithms and Protocols for Wireless Sensor Networks*. John Wiley and Sons, 2009.

[bus99] 21 ideas for the 21st century. *Business Week*, pages 78–167, Aug. 30 1999.

[BWZ04] M. Burkhart, R. Wattenhofer, and A. Zollinger. Does topology control reduce interference? In *Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc 2004)*, pages 9–19, New York, NY, USA, May 2004.

[CD05] D. Cook and S. Das. *Smart Environments: Technology, Protocols and Applications*. Wiley-Interscience, 2005.

[CE04] A. Cerpa and D. Estrin. Ascent: Adaptive self-configuring sensor networks topologies. *IEEE Transactions on Mobile Computing*, 3(3):272–285, 2004.

[CJBM02] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *Wireless Networks*, 8(5):481–494, 2002.

[Con] Harbor Research Consultant. 2009-2013 M2M/Pervasive Internet Market Forecast Report. `http://www.harborresearch.com`.

[CS04] X. Chen and J. Shen. Reducing connected dominating set size with multipoint relays in ad hoc wireless networks. In *Proceedings of the 7th International Symposium on Parallel Architectures, Algorithms and Networks (I-SPAN 2004)*, pages 539–543, Hong Kong, SAR, China, May 2004.

[DB97] B. Das and V. Bharghavan. Routing in ad-hoc networks using minimum connected dominating sets. In *Proceedings of IEEE International Conference on Communications (ICC 97)*, volume 1, pages 376–380, Montreal, Canada, June 1997.

[DHSR08] G. Demiris, B. Hensel, M. Skubic, and M. Rantz. Senior residents' perceived need of and preferences for "smart home" sensor technologies. *International Journal of Technology Assessment in Health Care*, 24(1):120, 2008.

[DMSL11] W. Dargie, R. Mochaourab, A. Schill, and G. Lin. A topology control protocol based on eligibility and efficiency metrics. *Journal of Systems and Software*, 84(1):2 – 11, 2011.

[DP04] D. Z. Du and P. Pardalos. *Handbook of Combinatorial Optimization*, page 332. Kluwer Academic Publishers, 2004.

[ECV+05] S. C. Ergen, S. Y. Cheung, P. Varaiya, R. Kavaler, and A. Haoui. Wireless sensor networks for traffic monitoring. In *Proceedings of the 4th*

International Symposium on Information Processing in Sensor Networks (IPSN 2005), UCLA, Los Angeles, April 2005.

[EG01] W. Edwards and R. Grinter. At home with ubiquitous computing: Seven challenges. In *Proceedings of the 3rd International Conference on Ubiquitous Computing (Ubicomp 2001)*, pages 256–272, Atlanta, GA, USA, October 2001.

[eme03] 10 emerging technologies that will change the world. *MIT Technology Review*, 106(1):33–49, Jan 2003.

[Fee] L. M. Feeney. Energy Framework: An Extensible Framework for Simulating Battery Consumption in Wireless Networks. `http://www.sics.se/~lmfeeney/software/energyframework-0.9.html`.

[FN01] L. M. Feeney and M. Nilsson. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM 2001)*, volume 3, pages 1548–1557, Alaska, USA, April 2001.

[FTS11] Z. Fan, S. K. Tan, and M. Sooriyabandara. M2M communications in the smart grid: Applications, standards, enabling technologies, and research challenges. *International Journal of Digital Multimedia Broadcasting*, 2011. `doi:10.1155/2011/289015`.

[GJ79] M. R. Garey and D. S. Johnson. Computers and intractability. *A Guide to the Theory of NP-Completeness*, 1979.

[GK98] S. Guha and S. Khuller. Approximation algorithms for connected dominating sets. *Algorithmica*, 20(4):374–387, 1998.

[GT95] M. Gerla and J. T. C. Tsai. Multicluster, mobile, multimedia radio network. *Wireless networks*, 1(3):255–265, 1995.

[GTJ+11] W. Geng, S. Talwar, K. Johnsson, N. Himayat, and K. D. Johnson. M2M: From mobile to embedded internet. *IEEE Communications Magazine*, 49(4):36 – 43, 2011.

[HCB00] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS 2000)*, volume 8, page 8020, Maui, Hawaii, USA, January 2000.

[HCG09] E. B. Hamida, G. Chelius, and J. M. Gorce. Impact of the physical layer modeling on the accuracy and scalability of wireless network simulation. *Simulation*, 85(9):574–588, 2009.

[HDB04] B. Hohlt, L. Doherty, and E. Brewer. Flexible power scheduling for sensor networks. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN 2004)*, pages 205–214, Berkeley, California, USA, April 2004.

[HKS+04] T. He, S. Krishnamurthy, J. A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh. Energy-efficient surveillance system using wireless sensor networks. In *Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services (Mobisys 2004)*, pages 270–283, Boston, MA, USA, June 2004.

[Hu93] L. Hu. Topology control for multihop packet radio networks. *IEEE Transactions on Communications*, 41:1474–1481, 1993.

[JO02] E. Jennings and C. Okino. Topology control for efficient information dissemination in ad-hoc networks. In *Proceedings of International Symposium on Performance Evaluation of Computer and Telecommunication*

Systems (SPECTS 2002), volume 2002, San Diego, California, USA, July 2002.

[JSAC01] C. E. Jones, K. M. Sivalingam, P. Agrawal, and J. C. Chen. A survey of energy efficient network protocols for wireless networks. *Wireless Networks*, 7(4):343–358, 2001.

[KK03] Vikas Kawadia and P.R. Kumar. Power control and clustering in ad hoc networks. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM 2003)*, volume 1, pages 459–469, San Francisco, USA, March 2003.

[KNG+04] D. Kotz, C. Newport, R. S. Gray, J. Liu, Y. Yuan, and C. Elliott. Experimental evaluation of wireless simulation assumptions. In *Proceedings of the 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2004)*, pages 78–82. Venice, Italy, October 2004.

[KWL+09] D. Kim, Y. Wu, Y. Li, F. Zou, and D. Z. Du. Constructing minimum connected dominating sets with bounded diameters in wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 20(2):147–157, 2009.

[KYZS05] B. Kim, J. Yang, D. Zhou, and M. T. Sun. Energy-aware connected dominating set construction in mobile ad hoc networks. In *Proceedings of the 14th International Conference on Computer Communications and Network (ICCCN 2005)*, pages 229–234, San Diego, California, USA, October 2005.

[LCD03] J. Lundquist, D. Cayan, and M. Dettinger. Meteorology and hydrology in yosemite national park: A sensor network application. In *Information Processing in Sensor Networks*, pages 553–553, 2003.

[LH04] L. Li and J. Y. Halpern. A minimum-energy path-preserving topology-control algorithm. *IEEE Transactions on Wireless Communications*, 3(3):910–921, May 2004.

[LHB+05] L. Li, J. Y. Halpern, P. Bahl, Y. M. Wang, and R. Wattenhofer. A cone-based distributed topology-control algorithm for wireless multi-hop networks. *IEEE/ACM Transactions on Networking (TON)*, 13(1):159, 2005.

[LLM+05] E. L. Lloyd, R. Liu, M. V. Marathe, R. Ramanathan, and S. S. Ravi. Algorithmic aspects of topology control problems for ad hoc networks. *Mobile Networks and Applications*, 10(1):19–34, 2005.

[LMFJ+04] K. Lorincz, D. J. Malan, T. R. F. Fulford-Jones, A. Nawoj, A. Clavel, V. Shnayder, G. Mainland, M. Welsh, and S. Moulton. Sensor networks for emergency response: Challenges and opportunities. *Pervasive Computing, IEEE*, 3(4):16–23, 2004.

[LSM06] O. Liang, Y. A. Sekercioglu, and N. Mani. A survey of multipoint relay based broadcast schemes in wireless ad hoc networks. *IEEE Communications Surveys and Tutorials*, 8(4):30–46, 2006.

[LTW+05] 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. *Wireless Communications and Mobile Computing*, 5(8):927–932, 2005.

[LW01] X. Y. Li and P. J. Wan. Minimum energy mobile wireless networks revisited. In *Proceedings of the IEEE International Conference on Communications (ICC 2001)*, Helsinki, Finland, June 2001.

[LW09] M. A. Labrador and P. M. Wightman. *Topology Control in Wireless Sensor Networks: with a Companion Simulation Tool for Teaching and Research*. Springer, 2009.

[MDJ+06] 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*, 35(1):111–119, 2006.

[MF10] F. Mattern and C. Floerkemeier. The internet of things. In *Proceedings of the 1st International Conference on the Internet of Things (IOT 2008)*, pages 242 – 259, Zurich, Switzerland, March 2010.

[MGG10] Y. Ma, Y. Guo, and M. Ghanem. RECA: Referenced energy-based CDS algorithm in wireless sensor networks. *International Journal of Communication Systems*, 23(1):125–138, 2010.

[MM09] P. McDaniel and S. McLaughlin. Security and privacy challenges in the smart grid. *IEEE Security & Privacy*, pages 75–77, 2009.

[MR04] V. Mhatre and C. Rosenberg. Homogeneous vs heterogeneous clustered sensor networks: a comparative study. In *Proceedings of IEEE International Conference on Communications (ICC 2004)*, volume 6, pages 3646–3651, Paris, France, June 2004.

[Mul10] G. Mulligan. The internet of things: Here now and coming soon. *IEEE Internet Computing*, 14(1):35 – 36, 2010.

[NKSK02] S. Narayanaswamy, V. Kawadia, R. S. Sreenivas, and P. R. Kumar. Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the COMPOW protocol. In *Proceedings of European Wireless Conference (EW 2002)*, pages 156–162, Florence, Italy, February 2002.

[NLP11] D. Niyato, X. Lu, and W. Ping. Machine-to-machine communications for home energy management system in smart grid. *IEEE Communications Magazine*, 49(4):53 –59, April 2011.

[Pos09] S. Poslad. *Ubiquitous Computing Smart Devices, Smart Environments and Smart Interaction*. Wiley-Interscience, 2009.

[PS03] S. J. Park and R. Sivakumar. Adaptive topology control for wireless ad hoc networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 7(3):37–38, 2003.

[QVL02] A. Qayyum, L. Viennot, and A. Laouiti. Multipoint relaying for flooding broadcast messages in mobile wireless networks. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS 2002)*, pages 3866–3875, Hawaii, USA, January 2002.

[RKH00] R. Ramanathan and R. Kosales-Hain. Topology control of multihop wireless networks using transmit power adjustment. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM 2000)*, volume 2, pages 404–413, Tel Aviv, Israel, March 2000.

[RM99] V. Rodoplu and T. H. Meng. Minimum energy mobile wireless networks. *IEEE Journal on Selected Areas in Communications*, 17(8):1333–1344, Aug 1999.

[RSPS02] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava. Energy-aware wireless microsensor networks. *IEEE Signal Processing Magazine*, 19(2):40–50, 2002.

[San05a] P. Santi. *Topology Control in Wireless Ad Hoc and Sensor Networks*. John Wiley and Sons, 2005.

[San05b] P. Santi. Topology control in wireless ad hoc and sensor networks. *ACM Computing Surveys*, 37(2):164–194, 2005.

[SBC03] G. Srivastava, P. Boustead, and J. F. Chicharo. A comparison of topology control algorithms for ad-hoc networks. In *Australian Telecommunications, Networks and Applications Conference (ATNAC 2003)*, Melbourne, Australia, December 2003.

[SCCZ07] Z. Shen, Y. Chang, C. Cui, and X. Zhang. A topology control algorithm for preserving minimum-energy paths in wireless ad hoc networks. *Frontiers of Electrical and Electronic Engineering in China*, 2(1):63–67, 2007.

[SDB98] R. Sivakumar, B. Das, and V. Bharghavan. An improved spine-based infrastructure for routing in ad hoc networks. In *Proceedings of the 3rd IEEE Symposium on Computers and Communications (ISCC 1998)*, Athens, Greece, June 1998.

[SFEB09] V. K. Sood, D. Fischer, J. M. Eklund, and T. Brown. Developing a communication infrastructure for the smart grid. In *Electrical Power Energy Conference (EPEC), 2009 IEEE*, pages 1 –7, October 2009.

[SSK+08] K. Sakai, F. Shen, K. M. Kim, M. T. Sun, and H. Okada. Multi-initiator connected dominating set construction for mobile ad hoc networks.

In *Proceedings of IEEE International Conference on Communications (ICC 2008)*, pages 2431–2436, Beijing, China, May 2008.

[SSKO08] Kazuya Sakai, Min-Te Sun, Wei-Shinn Ku, and Hiromi Okada. Maintaining cds in mobile ad hoc networks. In *Wireless Algorithms, Systems, and Applications*, pages 141–153. Springer, 2008.

[SSZ02] I. Stojmenovic, M. Seddigh, and J. Zunic. Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 13(1):14–25, 2002.

[Ste05] D. Steel. Smart dust. In *ISRC Technology Briefing*, 2005.

[Sto05] I. Stojmenovic. *Handbook of Sensor Networks: Algorithms and Architectures*. John Wiley and Sons, 2005.

[Sto08] I. Stojmenovic. Simulations in wireless sensor and ad hoc networks: matching and advancing models, metrics, and solutions. *IEEE Communications Magazine*, 46(12):102–107, 2008.

[STS02] C. Schurgers, V. Tsiatsis, and M. B. Srivastava. STEM- topology management for energy efficient sensor networks. In *Proceedings of IEEE Aerospace Conference*, volume 3, pages 3–1099, Montana, USA, March 2002.

[THH02] Y. C. Tseng, C. S. Hsu, and T. Y. Hsieh. Power-saving protocols for IEEE 802.11-based multi-hop ad hoc networks. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM 2002)*, volume 1, pages 200–209, New York, NY, USA, June 2002.

[TMB01] M. Takai, J. Martin, and R. Bagrodia. Effects of wireless physical layer modeling in mobile ad hoc networks. In *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiCom 2001)*, pages 87–94, Rome, Italy, July 2001.

[TNCS02] Y. C. Tseng, S. Y. Ni, Y. S. Chen, and J. P. Sheu. The broadcast storm problem in a mobile ad hoc network. *Wireless networks*, 8(2):153–167, 2002.

[Var] A. Varga. OMNeT++ Object-Oriented Discrete Event Simulation Framework. `http://www.omnetpp.org`.

[VH08] A. Varga and R. Hornig. An overview of the OMNeT++ simulation environment. In *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops (SIMUTools 2008)*, pages 1–10, Marseille, France, March 2008.

[Vie98] L. Viennot. *Complexity results on election of multipoint relays in wireless networks*, 1998.

[WAF04] P. J. Wan, K. M. Alzoubi, and O. Frieder. Distributed construction of connected dominating set in wireless ad hoc networks. *Mobile Networks and Applications*, 9(2):141–149, 2004.

[WD03] J. Wu and F. Dai. Broadcasting in ad hoc networks based on self-pruning. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM 2003)*, volume 3, pages 2240–2250, San Francisco, USA, January 2003.

[WDJ+06] 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*, 352(1-3):1–7, 2006.

[WGS01] J. Wu, M. Gao, and I. Stojmenovic. On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks. In *Proceedings of the 30th International Workshops on Parallel Processing (ICPP 2001)*, pages 346–356, Valencia, Spain, September 2001.

[WL99] J. Wu and H. Li. On calculating connected dominating set for efficient routing in ad hoc wireless networks. In *Proceedings of the 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM 1999)*, page 14, Seattle, WA, USA, August 1999.

[WL08] P. M. Wightman and M. A. Labrador. A3: A topology construction algorithm for wireless sensor networks. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM 2008)*, pages 1–6, New Orleans, LA, USA, November 2008.

[WLBW01] R. Wattenhofer, L. Li, P. Bahl, and Y. M. Wang. Distributed topology control for power efficient operation in multihop wireless ad hoc networks. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM 2001)*, volume 3, pages 1388–1397, Anchorage, Alaska, USA, April 2001.

[WLD06] J. Wu, W. Lou, and F. Dai. Extended multipoint relays to determine connected dominating sets in MANETs. *IEEE Transactions on Computers*, pages 334–347, 2006.

[WPMR07] A. Warrier, S. Park, J. Min, and I. Rhee. How much energy saving does topology control offer for wireless sensor networks? – A practical study. *Computer Communications*, 30(14-15):2867–2879, 2007.

[WSKW] K. Wessel, M. Swigulski, A. Kopke, and D. Willkomm. A Mixed Simulator (MiXiM) for Mobile and Wireless Networks. http://mixim.sourceforge.net.

[WST] WST: Application areas: Condition based monitoring. http://wins.rockwellscientific.com/WST_CBM.html.

[Wu02] J. Wu. Extended dominating-set-based routing in ad hoc wireless networks with unidirectional links. *IEEE Transactions on Parallel and Distributed Systems*, 13(9):866–881, 2002.

[WW03] J. Wu and B. Wu. A transmission range reduction scheme for power-aware broadcasting in ad hoc networks using connected dominating sets. In *Proceedings of the IEEE Semiannual Vehicular Technology Conference (VTC 2003)*, volume 5, pages 2906–2909, Orlando, Florida, USA, April 2003.

[WWK03] S. C. Wang, D. S. L. Wei, and S. Y. Kuo. A topology control algorithm for constructing power efficient wireless ad hoc networks. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM 2003)*, pages 1290–1294, San Francisco, USA, December 2003.

[WWS03] J. Wu, B. Wu, and I. Stojmenovic. Power-aware broadcasting and activity scheduling in ad hoc wireless networks using connected dominating sets. *Wireless Communications and Mobile Computing*, 3(4):425–438, 2003.

[WX06] L. Wang and Y. Xiao. A survey of energy-efficient scheduling mechanisms in sensor networks. *Mobile Networks and Applications*, 11(5):723–740, 2006.

[YF04] O. Younis and S. Fahmy. HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Transactions on Mobile Computing*, pages 366–379, 2004.

[YHE01] X. Ya, J. Heidemann, and D. Estrin. Geography-informed energy conservation for ad hoc routing. In *Proceedings of the 7th International Conference Mobile Computing & Networking (MobiCom 2001)*, pages 70–84, Rome, Italy, July 2001.

[YJY06] Z. Yuanyuan, X. Jia, and H. Yanxiang. Energy efficient distributed connected dominating sets construction in wireless sensor networks. In *Proceedings of the International Conference on Wireless Communications and Mobile Computing (IWCMC 2006)*, pages 792–802, Vancouver, British Columbia, Canada, July 2006.

[YRS+11] Z. Yan, Y. Rong, X. Shengli, Y. Wenqing, X. Yang, and M. Guizani. Home M2M networks: Architectures, standards, and QoS improvement. *IEEE Communications Magazine*, 49(4):44 – 52, 2011.

[YSS11] B. Yin, H. Shi, and Y. Shang. An efficient algorithm for constructing a connected dominating set in mobile ad hoc networks. *Journal of Parallel and Distributed Computing*, 71(1):27–39, 2011.

[YWD09] R. Yu, X. Wang, and S. K. Das. EEDTC: Energy-efficient dominating tree construction in multi-hop wireless networks. *Pervasive and Mobile Computing*, 5(4):318–333, 2009.

[ZSL05] D. Zhou, M. T. Sun, and T. H. Lai. A timer-based protocol for connected dominating set construction in ieee 802.11 multihop mobile ad hoc networks. In *Proceedings of the Symposium on Applications and the Internet*, pages 2–8. IEEE, 2005.

[ZVS11] A. Zaballos, A. Vallejo, and J. M. Selga. Heterogeneous communication architecture for the smart grid. *IEEE Network*, 25(5):30–37, 2011.