COOPERATIVE MOBILITY MAINTENANCE TECHNIQUES FOR INFORMATION EXTRACTION FROM MOBILE WIRELESS SENSOR NETWORKS

ABDELRAHMAN ABUARQOUB

A thesis submitted in partial fulfilment of the requirements for the degree of
Doctor of Philosophy

Faculty of Science and Engineering

Manchester Metropolitan University

May, 2014

Abstract

Recent advances in the development of microprocessors, microsensors, ad-hoc wireless networking and information fusion algorithms led to increasingly capable Wireless Sensor Networks (WSNs). Besides severe resource constraints, sensor nodes mobility is considered a fundamental characteristic of WSNs.

Information Extraction (IE) is a key research area within WSNs that has been characterised in a variety of ways, ranging from a description of its purposes to reasonably abstract models of its processes and components. The problem of IE is a challenging task in mobile WSNs for several reasons including: the topology changes rapidly; calculation of trajectories and velocities is not a trivial task; increased data loss and data delivery delays; and other context and application specific challenges. These challenges offer fundamentally new research problems.

There is a wide body of literature about IE from static WSNs. These approaches are proved to be effective and efficient. However, there are few attempts to address the problem of IE from mobile WSNs. These attempts dealt with mobility as the need arises and do not deal with the fundamental challenges and variations introduced by mobility on the WSNs.

The aim of this thesis is to develop a solution for IE from mobile WSNs. This aim is achieved through the development of a middle-layer solution, which enables IE approaches that were designed for the static WSNs to operate in the presence of multiple mobile nodes. This thesis contributes toward the design of a new self-stabilisation algorithm that provides autonomous adaptability against nodes mobility in a transparent manner to both upper network layers and user applications. In addition, this thesis proposes a dynamic network partitioning protocol to achieve high quality of information, scalability and load balancing.

The proposed solution is flexible, may be applied to different application domains, and less complex than many existing approaches. The simplicity of the solutions neither demands great computational efforts nor large amounts of energy conservation. Intensive simulation experiments with real-life parameters provide evidence of the efficiency of the proposed solution. Performance experimentations demonstrate that the integrated DNP/SS protocol outperforms its rival in the literature in terms of timeliness (by up to 22%), packet delivery ratio (by up to 13%), network scalability (by up to 25%), network lifetime (by up to 40.6%), and energy consumption (by up to 39.5%). Furthermore, it proves that DNP/SS successfully allows the deployment of static-oriented IE approaches in hybrid networks without any modifications or adaptations.

Acknowledgments

I would like to express my sincere gratitude and appreciation to my director of studies, Dr. Mohammad Hammoudeh, for his sustained support throughout the course of conducting this research. His enthusiasm, insight and guidance were invaluable assets which encouraged me to proceed well beyond my initial ideas.

It is an honour for me to thank my wonderful parents for their unlimited love and support, their encouragement and support is a key factor in any achievement I have ever made. I would like to thank my brothers and sisters, friends, and all the people who have supported me along the way.

Finally, but not less important, I would like to thank my beloved wife, Yasmin, for her patience and continuous encouragement. I would also like to acknowledge the kind assistance offered by her, during the concluding stages of my PhD.

Above all, I thank God for giving me the perseverance required to ensure that the work I started was completed.

Contents

List of Figures

List of Tables

List of Algorithms

List of Publications

[1] **A. Abuarqoub**, & M. Hammoudeh (2013). *Self-stabilizing algorithm for information extraction from mobile wireless sensor networks. Paper presented at the Wireless and Mobile Networking Conference (WMNC), 2013 6th Joint IFIP/IEEE Wireless and Mobile Networking Conference. Dubai, UAE.*

[2] **A. Abuarqoub**, M. Hammoudeh, & T. Alsboui (2012). An Overview of Information Extraction from Mobile Wireless Sensor Networks. *Internet of Things, Smart Spaces, and Next Generation Networking* (Vol. 7469, pp. 95-106): Springer Berlin Heidelberg.

[3] **A. Abuarqoub**, F. Alfayez, M. Hammoudeh, T. Alsboui, & A. Nisbet (2012). *Simulation Issues in Wireless Sensor Networks: A Survey.* Paper presented at the SENSORCOMM '12: Proceedings of the 2012 Sixth International Conference on Sensor Technologies and Applications, Rome, Italy.

[4] T. Alsboui**, A. Abuarqoub**, M. Hammoudeh, Z. Bandar, & A. Nisbet (2012). Information Extraction from Wireless Sensor Networks: System and Approaches. *Sensors and Transducers Journal (ISSN 1726- 5479), 14*(2), 1-17.

[5] F. Al-Fayez, **A. Abuarqoub**, M. Hammoudeh, & A. Nisbet (2013). Wireless Sensor Network Simulation: The Current State and Simulation Tools. *Sensors & Transducers Journal (ISSN 1726- 5479), 18*(Special Issue), 145-155.

[6] T. Alsboui, M. Hammoudeh, & **A. Abuarqoub** (2012). A Service-Centric Stack for Collaborative Data Sharing and Processing. *Green and Smart Technology with Sensor Applications* (Vol. 338, pp. 320-327): Springer Berlin Heidelberg.

Chapter 1

Introduction

Machine-to-machine methods (M2M), intelligent embedded or smart fusion have been around in some form for many years. Wireless Sensor Networks (WSNs) goes way beyond this capability and interconnects large numbers of sensors and changes the way we interact with our environment. WSNs appear in several scientific, commercial, health, surveillance, and military applications. Unattended sensor nodes offer maintenance free operation that detects, characterises, and disseminates situational awareness continuously. After years of enthusiastic research on the opportunities it will provide and how much it will be worth, many of those looking to play in the WSNs space are starting to look at the potential practical problems, including data management. Once the WSN is up and running at full scale, it will generate large quantities of data that need to be processed and analysed in real time.

The ultimate goal of WSNs is to collect data from the environment and analyse it to extract needed information or even discover previously unknown patterns. The success of such applications is dependent on knowing that information is available, the type of information, its quality, its scope of applicability, limits to use, duration of applicability, likely return, cost to obtain, and a host of other essential details. A wide range of WSNs applications generate large volumes of data that is imperfect in nature and have considerable redundancy [1]. Depending on the type of the application the data can be valid for a short period of time, can be very complicated, and can be multidimensional, or multimodal data. Therefore, the data needs to be processed efficiently in order to extract information enfolded in that data. It is thought that it is a challenging task to extract information with user's desired quality constraints.

To aid in data collection, the use of mobile nodes has been widely suggested in the literature. Nodes movement can be controlled and optimised to improve data collection and analysis. For instance, they can be used to bridge disconnected parts of the network. Furthermore, nodes mobility can optimise energy efficiency and lifetime of WSNs. For example, moving the sink to data sources or moving the sensor nodes towards the sink is one way to avoid the communication bottlenecks.

However, the deployment of mobile nodes instigates frequent topological changes that need to be resolved before data collection can be resumed. For instance, many WSNs applications rely heavily on the node's ability to establish position information. The situation becomes more complex when the entire network population is mobile. This increases the network complexity in general, in terms of connectivity maintenance, and proliferates the cost of extracting information. This is the case in many applications such as battlefield surveillance, which is the application that will be used throughout this thesis.

Battlefield surveillance applications are primarily designed to detect predefined events that may represent danger or threat to ground forces, such as the presence of non-authorised vehicles or people in a restricted territory. In this research, this application is used to identify the challenges that need to be resolved before extracting information in such environments is possible. Battlefield surveillance sensing is multidimensional, highly dynamic and complex enough to illustrate general concepts of information extraction (IE) from hybrid (static and mobile) WSNs. Moreover, end users of this application require high quality information to be delivered in a timely manner. The application offers a comprehensive set of conditions and settings that allows the application of the proposed solution for the challenges it presents to other applications. We also use this application to evaluate the viability of the proposed methods and algorithms.

This chapter is organised as follows. Section 1.1 presents definitions and assumptions that are used throughout the thesis. It also briefly describes the general network model addressed in the research. Section 1.2 presents the motivation of this work and outlines our proposed solution. Section 1.3 provides a summary of the contributions presented in this thesis. Finally, section 1.4 presents the organisation of the thesis.

1.1 Definitions and Network Model

1.1.1 Network Types

A general purpose battlefield surveillance system has specific requirements that necessitates the use of static and mobile sensor nodes. The type of sensor nodes in this type of system depends on the conditions in which these systems are deployed [2]. For clarity, we classify WSNs into three different classes: static, mobile, and hybrid networks.

Static networks consist of stationary sensor nodes, usually also called unattended ground sensors (UGS) [3]. Commonly, such sensor nodes have limited energy, computation, and storage resources.

Mobile networks offer extended usage of WSNs, by providing sensor nodes with the capability to change their position according to the network or application requirements. Mobile sensor nodes enhance the surveillance missions in multiple ways. For instance, attaching image sensors on mobile nodes may result in the coverage of a wider geographic area. Mobile nodes can move randomly in the monitored space or their mobility can be planned or controlled to achieve specific objectives. Mobile nodes are normally equipped with a motor to move from one place to another or they can be attached to a mobile target. For instance, a sensor node could be attached to a bomb disposal suite to monitor the body conditions of the explosives expert. Thus, mobile sensor nodes can move at various speeds in different directions. For brevity, in this thesis we refer to mobile WSNs as mWSNs.

A hybrid network is often used to refer to a network consisting of a combination of static and mobile nodes [4, 5]. The percentage of each class of nodes is undefined and depends on individual deployments. Using hybrid networks is a promising network design that allows the deployment of advanced sensing systems [6, 7], by overcoming the limitation of static networks and the high costs of mobile networks.

1.1.1.1 Information Extraction

The aim of this thesis is to design a solution for Information Extraction (for brevity, IE). IE is defined as a practical multistage process of retrieving, filtering, collecting and processing the unstructured or semi-structured sensor data. IE approaches designed for static WSNs are referred to as static-oriented IE approaches. Where IE approaches designed for mWSNs are referred to as mobile-oriented IE approaches. Similarly, IE approaches designed for hybrid WSNs are referred to as hybrid-oriented IE approaches.

1.1.2 A Brief Description of the Network Model

In this research, the network is abstracted to a pyramid three-tier network model. Each tier contains different class of sensor nodes. The bottom tier hosts a static and mobile sensor nodes, which form the majority of the network population. Sensor nodes at this tier perform sensing and communication tasks. The middle tier hosts a small number of resource-rich mobile nodes called Mobile Data Collectors (for short, MDCs). MDCs have long range radio and are considered power rich devices. The top tier of the hierarchy hosts the fixed data sink(s). At this tier, information received from different sources is further fused and presented to the end users. The details of this network model are provided in Section 3.2.

1.2 Motivation & Solution Outline

Highly dynamic networks, such as mWSNs, poses new challenges to IE. In the literature, many research efforts are devoted to IE from static WSNs. In [10], a recent survey is presented to summarise static-oriented approaches to IE. The main limitation to these approaches is the disregard of the spatial and temporal properties of mWSN data. Yet, there is only a handful set of papers directly addressing IE from mWSNs, e.g.[11-15]. Contrary to expectations, these solutions dealt with individual mobility related problems as the need arises. A great number of papers in related areas, e.g., self-organisation [16], routing [17] and trajectory calculation [14], contain useful results and techniques that could be employed to IE from mWSNs. In all of the approaches cited above, it is not feasible to simply load data arriving from mobile nodes into static-oriented IE algorithms and operate on it.

To date, static-oriented IE are incapable of efficiently extracting information from a hybrid network without major modifications. Rather than developing new paradigms for IE from hybrid or mWSNs, this research is set out to provide an effective middle layer solution that enables seamless application of static-oriented IE approaches to mWSNs.

Our solution deals with mobility at different tiers of the network model separately. To address mobility at the bottom tier, we design a solution to rapidly handle network topological or structural changes in order to provide a virtually static network. To achieve this, we design and implement an autonomic self-stabilisation algorithm to abstract nodes movement from upper layers of the network stack. Self-stabilisation aims to reduce the cost of mobility maintenance as well as to reduce interruption periods to IE. To address MDCs mobility, we developed a set of algorithms to dynamically partition the network into well delimited areas.

1.3 Thesis Contributions

Aiming at improving the quality of extracted information at minimal cost, we focus our research efforts at the development of mobility management techniques that enable the deployment of static-oriented IE on mWSNs and hybrid WSNs. The main contributions of this thesis are as follows:

1. A distributed **self-stabilising algorithm** to overcome the issues of high energy consumption incurred in reconfiguring network topology due to nodes mobility. The algorithm rapidly stabilises the underlying network topology by establishing some approximation to recover from an arbitrarily corrupt communication state.
2. A **dynamic network partitioning protocol** to manage MDCs mobility and optimise their performance to increase extracted information quality. The key contributions of this protocol are summarised as follows:
    a. **Optimal MDC selection algorithm:** This algorithm defines an IE effective method to select the 'best' MDC for a sensor node to join. This algorithm can be applied without any modifications to general network deployments where multiple data collectors/sinks are present. The algorithm improves the stability of the network and reduces the topology reconfiguration frequency.

   b. **Dynamic network partitioning algorithm:** To achieve scalability and reduce mobility management overhead, we develop a distributed algorithm to dynamically partition the network. These network partitions can be used to support IE by offering free query scoping, i.e., identify 'significant' nodes. The algorithm sets up communication paths with the MDC promptly to maximise the number of messages successfully received at the sink. This offers real-time, low-cost, and high quality information.

3. As a result of the above contributions, enabling seamless application of static-oriented IE approaches to mWSNs becomes possible. This claim was thoroughly evaluated and supported by experimental study that uses real-life parameters. Experimental results show that our work outperforms its best rival in the literature.

1.4 Thesis Outline

In addition to the introduction chapter, this thesis consists of other six chapters.

Chapter 2: Background and Related Work

This chapter presents the relevant background material and surveys previous related work. The last section summarises works from the literature that we compare our solution against.

Chapter 3: Motivation Scenario and System Specifications

In the first section of chapter 3, an application scenario that motivates our research is presented and the network model is described in details. The following section describes the system’s specifications and design principles.

Chapter 4: Self Stabilization for Mobility Management

Chapter 4 presents a new self-stabilisation algorithm for mWSNs. The chapter starts by giving an overview of self-stabilisation algorithms and their suitability to WSNs. Next, a detailed description of the proposed algorithm and its proof of correctness are provided.

Chapter 5: Dynamic Network Partitioning

In chapter 5, our dynamic network partitioning protocol for mWSNs is presented. This includes the proposed MDC selection algorithm. The chapter thoroughly describes the different phases of the protocol.

Chapter 6: Implementation and Evaluation

Chapter 6 presents the performance analysis and evaluation of DNP/SS in comparison to its rival from the literature.

Chapter 7: Conclusion

Chapter 7 provides a concise summary of the thesis bringing the conclusions for each contribution. Finally, a discussion about ideas that can be used as starting point to extend and enhance this work is presented.

Chapter 2

Background and Related Work

This chapter presents an overview of the state of the art and highlights important topics that are related to the work reported in this thesis. These topics are: wireless sensor networks, IE from static and mobile WSNs, mobility benefits and challenges, attributes of mWSNs information. This overview aims to briefly present the main concepts and ideas concerning the above mentioned topics to ease the understanding of the thesis content and contributions.

2.1 Wireless Sensor Networks

Wireless sensor networks (WSNs) are distributed systems composed of spatially distributed sensor nodes to monitor physical phenomena, e.g. temperature, pressure, light, vibration, or acceleration. Sensor nodes are characterised by limited physical resources, i.e. power, memory, computation, and bandwidth. Nodes are equipped with various sensors to observe the conditions of the monitored environment. Typically, WSNs are accessed via special nodes called data sinks or access points, which can be static or mobile. The purpose of the sink node is to provide an interface between the sensor nodes and end-user application. Depending on the size of WSN and its application, multiple sinks may be present in the network.

WSNs have several advantages. The small size and low cost of sensor nodes make WSNs applicable in many application and various extends. They are self-organised and easy to deploy. They can cover large areas. They can be an economical method for long-term data gathering. They can operate untended in potentially hazardous environments, such as volcano eruptions. They may use some level of fault tolerance which keeps the network operating when one or a few nodes fail.

All these advantages and the low cost of sensor nodes and the availability of various sensors motivated a large number of applications areas; such as environmental monitoring, in the military, health care, agriculture, commerce, civil engineering, surveillance, etc. For example, in health care applications, sensor nodes can be attached to a patient to monitor his heart rate. Environmental applications may range from habitat monitoring to environment observation, e.g., forest fire monitoring. Military applications range from target tracking to battlefield surveillance.

The main purpose of a WSN is to provide users with access to the information of interest from data collected by spatially distributed sensors. In real-world applications, sensors are often deployed in high numbers to ensure a full exposure of the monitored physical environment. Consequently, such networks are expected to generate enormous amount of data. The desire to locate and obtain information makes the success of WSNs applications, largely, determined by the quality of the extracted information.

Sensor nodes can be static or mobile, depending on their intended usage. If the sensor nodes in a network are mobile, this network is referred as a mobile wireless sensor network (mWSNs) [8]. If the network consists of a combination of static and mobile nodes, then the term hybrid is often used for this type of WSNs [4, 5]. The presence of node mobility can enhance the WSNs performance but at the same time it has many challenges and needs a lot of management.

2.2 Information Extraction (IE)

The main goal of WSNs is to collect data from the environment and send it to end user's applications, where it is analysed to get the most information out of the data. The success of such applications is dependent on knowing that information is available, the type of information, its quality, its scope of applicability, limits to use, duration of applicability, likely return, cost to obtain, and a host of other essential details. In real-world applications, sensors are often deployed in high numbers to ensure a full exposure of the monitored physical environment. Consequently, such networks are expected to generate enormous amount of data . The desire to locate and obtain information makes the success of WSNs applications, largely, determined by the accuracy and quality of the extracted information. The principal concerns when extracting information include the timeliness, accuracy, cost, and reliability of the extracted information and the methods used for extraction. The process of IE enables unstructured data to be retrieved and filtered from sensor nodes using sophisticated techniques to discover specific patterns [9]. Information Extraction (IE) is *a practical multistage process at which end user's applications operate using a structured methodology to discover and extract the information content enfolded in data.* Every stage has a particular purpose and function.

Practical constraints on sensor node implementation such as power consumption (battery limits), computational capability, and maximum memory storage, make IE a challenging distributed processing task. To reduce communication costs, many concepts have been developed in the context of distributed WSNs such as data aggregation and fusion. The in-network processing of sense data lead to no strict, clear, undisputed distinction between data and information. For this reason there is no set and commonly agreed upon definition of information and information attributes. The variety of definitions identified in the literature stem from the highly context specific nature of data mining and databases fields and the complexity of their operationalisation and conceptualisation. Differing definitions can lead to a variance in focus and performance evaluation comparison approaches. Out of this need, we propose standard, unified definitions of information and information attributes; the particular definitions adopted by this study will depend on the WSNs discipline and level of investigation.

Addressing the need for common definitions of some of the issues surrounding the concept of IE in respect of the WSNs we define data as a collection of unstructured chunks of facts and statistics measured by sensor nodes. Information is the processed and interpreted data that provides conceptual explanations of data, i.e. it converts information encapsulated in data into a form amenable to human cognition. Data comes in a variety of forms and carry different amounts of information. For instance, scalar sensor measurements from a proximity sensor about a monitored object can be limited to just 'present' or 'not-present'. In this case the user will see less information than the 'too close', 'close', 'far', 'very far', and 'not-present'.

IE is considered as a costly task because it involves the collection and processing of often large amount of unstructured or semi-structured sensor data. There is a wide body of literature about static-oriented IE; we refer interested readers to, a recent survey that provides a comprehensive review to IE approaches. However, there has been few attempts to address the problem of IE from mWSNs, e.g. [10-15]. These approaches dealt with mobility as the need arises and do not attempt to deal with the fundamental challenges and variations introduced by mobility on the WSNs.

2.2.1 Process of Converting Data to Information

Typically, WSNs have large numbers of sensor nodes that provide full coverage of the monitored area. Consequently, sensors generate large volumes of data that is imperfect in nature and contains considerable redundancy [16]. This data needs to be processed in order to extract information relevant to a user query. Converting data to information goes into variant and nested stages of processing including: data retrieval, filtering, collection, and processing. In the following, we explain the these process in details.

2.2.1.1 Data Retrieval (DR)

The DR process begins by specifying the information needed by a user/application. Often, this takes the form of a query or event trap. DR selects a relevant subset of data or nodes carrying data that is relevant to some needed information from a larger set. In other words, it identifies the nodes that carry data that is significant to the needed information. Nodes can be classified based on their soft-state (sensor readings), e.g. [17, 18], or on the physical location of nodes when the information is spatial in nature. For example, consider a situation where the end user is interested in locations where the temperature is above 50 °C. Identifying nodes carrying data that satisfy this condition has many benefits. First, it becomes easier to choose the best path over which to transmit data - in terms of energy consumption, link reliability, and end-to-end delay. Second, if relevant data can be identified, then the others can be abandoned. This reduces the load on the rest of the system and improves information accuracy.

2.2.1.2 Data Filtering

The filtering process can serve many purposes. The DR process supplies some erroneous and redundant data, which may be passed to the IE system producing incorrect information. Data filtering has been widely used to remove redundant or unwanted data from a data stream. Data aggregation, fusion and compression are clearly candidates for this task, but other techniques from active networking could also be used. Continuous and cumulative sensory readings can be filtered instantly in the network at different levels, e.g. at cluster heads or at a sink, to avoid the expensive transmission of inappropriate data. Essentially, filtering attempts to trade off communication for computation to reduce energy consumption since communication is the most energy-expensive task. Moreover, a filter may attempt to block data, which is an artefact of the DR process.

Data fusion has been placed forward as a technique to improve bandwidth utilisation, energy consumption, and information accuracy [19]. It combines and integrate multi-sensor and multi-sourced data to produce higher accuracy and comprehensive information [20]. The fusion technique achieves high information accuracy by fusing redundant observation. To produce comprehensive information, the fusion technique fuses readings from different sensors that are related to the same event. For example, in roads monitoring applications, to discover whether there is a frost or not, a combination of temperature, humidity and solar radiation data is needed. However, since WSNs are resource limited, traditional data fusion techniques cannot function efficiently in such networks. Furthermore, WSNs are application dependent networks, hence, it is infeasible to devote one data fusion technique to work with all situations. Consequently, there has been persistent research efforts to develop data fusion algorithms that suites several applications [21, 22].

Data fusion is usually coupled with another process called data aggregation. Data aggregation is the process of combining data coming from different sources in order to eliminate redundancy and minimise the number of transmissions [23]; thereby, conserving the scarce energy resources. Data aggregation is affected by the way that data is gathered at the sensor nodes and the way that packets are routed through the network [24]. Aggregation techniques can be broadly classified into two classes, lossless aggregation and lossy aggregation [25]. The former is when multiple data segments are merged into a single packet in order to reduce bandwidth utilisation. In the later, data is combined and compressed by applying statistical processes, such as average, minimum, and maximum, before transmission in order to keep the number of transmissions as low as possible.

2.2.1.3 Data Collection

Data collection is the most energy-expensive process of IE. There are two abstract models of data collection, central at a sink, or hierarchical at cluster heads. In WSNs, the data collection model is tightly coupled with how data is routed through the network, patterns of node mobility, and the communication paradigm. Data collection from mWSNs can be classified based on the nature of mobility into three classes: flat-tier; two-tier; and three-tier [26].

A flat-tier model, consists of a set of homogeneous sensor nodes, which can be static or mobile. Sensory data is routed from the originator sensors to a central sink in a multi-hop ad-hoc fashion. Centralised data collection is not desired because it increases the delay of extracted information, it causes communication bottlenecks around the sink, it is not suitable for large networks, amongst others.

In the two-tier hierarchical model, static sensor nodes occupy the bottom tier and mobile nodes occupy the top tier. Mobile nodes act as data collectors that move towards the sink to deliver their data. However, moving data collectors all the way to the sink stops the collection process for a while causing coverage holes. This results in high information delay, which could negatively impact the validity of the extracted information. Moreover, coverage holes could result in reduction of information availability.

In the three-tier hierarchical model, static nodes occupy the bottom and the top tiers. Whereas the middle tier hosts the mobile nodes, which act as access points. The mobile nodes collect sensory data from the bottom tier and forward it to the top tier. The top tier delivers the data to the sink. This model could greatly improve the overall system performance as it reduces the delay of data delivery, and hence, increasing the information validity. Furthermore, since the distances covered by the mobile nodes become shorter, the number of coverage holes will be reduced, which results in higher information availability.

2.2.1.4 Data Processing

Data processing is the stage before passing information to end users or application. It refers to a class of programs to organise, sort, format, transform, summarise, and manipulate large amounts of data to convert it into real-world information. These programs define operations on data such as algorithmic derivations, statistical calculations, and logical deductions that exists in the user application. This stage can be followed by other processes like information visualisation and display. Data processing includes a second phase of data filtering and could include a second iteration of data fusion or integration. Data processing can be performed centrally at the data sink, hierarchically on cluster heads or in a distributed manner [27].

In centralised processing models, the data is first gathered at a sink, where data processing is applied. This approach will produce high quality information as the entire network data is used for extracting the information. However, a fully centralised data collection and processing is not always feasible. This is because centralised data processing incurs a significant data transfer cost and introduces delay to information delivery. However, some information is of a spatio-temporal nature, which requires centralised data collection and processing; extracting information for characteristics of such nature where local processing is not enough should be feasible.

To solve the apparent problems posed by the centralised model, hierarchical data processing was proposed. It exploits local processing resources to effectively reduce the amount of data transmitted across the network. In this approach, sensor nodes are divided into multiple clusters. Measurements are transferred and processed on a sparsely distributed cluster heads. These cluster heads either send processed data to a fusion centre for decision making or collaborate with each other to make decisions. The hierarchical approach helps in reducing the delay of information delivery as data processing is performed on multi processors simultaneously. Moreover, since the amount of data transferred across the network is reduced, the energy usage will be reduced leading to improved information affordability.

In some scenarios, it is desirable or necessary to process data on site and, as a result, distributed processing provides a critical solution to in-field data analysis. Distributed data processing was proposed to exploit sensor nodes computational power. Sensor nodes process the data and collaborate to transform the data into information. Therefore, processing takes place in the network and only the results are returned. This approach improves the extracted information timeliness by distributing the computational work over all the nodes in the network. However, the information accuracy could be degraded due to lack of computational resources.

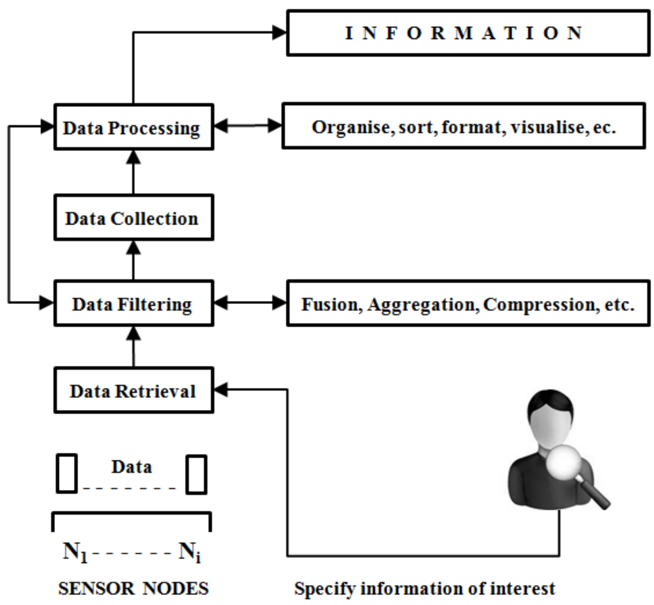


Figure 2.1: A general model of IE process

Figure 2.1 shows an illustration of the described processes. It shows the cooperation between the major entities and processes involved in the IE process.

2.3 Static-oriented IE

In terms of data delivery required by an application, IE in WSNs can be classified into four broad categories: Event-driven IE, periodic IE, on-demand IE, and hybrid. In Event-driven IE, a certain rule is defined and data will be reported continuously whenever the conditions of this rule (event) is met, while, in the periodic, data is periodically sent to a sink every constant interval of time. With on-demand IE, the data is collected according to end user's demand. Finally, the hybrid approach is a combination of two or more of the above.

IE is one of the most vital efforts to utilise the ever burgeoning amount of data returned by WSNs for achieving detailed, often costly task of finding, analysing and identifying needed information. The process of IE involves the classification of data based on the type of information they hold, and is concerned with identifying the portion of information related to a specific fact. In the context of WSNs, the notion of fact can be defined as a property or characteristic of the monitored phenomenon at a certain point in time or during a time interval. Fact can also refer to an event or action. An event is a pattern or exceptional change that occasionally appears in the observed environment [28]. Events have some distinct features that can be used as thresholds, e.g. temperature > 50, to make a distinction between usual and unusual environmental parameters.

An event may arise in many other forms. It can be a frequent, gradually occurs over time (e.g. temperature does not change instantly), and has obvious limit with normal environment parameters. In [29] complex events are defined as sequences of sensor measurements over a period of time indicating an unusual activity in the monitored environment. In WSNs, the network owners may be unaware in advance what type of events may occur. This is because one of the ultimate goals of such networks is to discover new events and interesting information about the monitored phenomenon. For this reason, threshold based event detection methods are not always efficient to identify and extract continuous IE facts. From this deficiency arise the need for periodic, on-demand, and hybrid IE approaches.

Since the number of existing IE approaches is significantly large, it will not be feasible to provide a detailed description of each approach. Instead, we have selected recent approaches that particularly represent directions of future research without focusing on the details of these approaches. However, characteristics of various approaches that are common for the approach they apply will be presented.

2.3.1 Event-driven IE

In Event-driven approaches to IE, the initiative is with the sensor node and the end user is in the position of an observer, waiting for incoming data. Any node may generate a report when a significant rule (e.g., a certain threshold) is met or an unusual event (e.g., fire) occurs. Event-driven IE is valuable for detecting events as soon as they occur over a specified region. In the simplest form, sensor nodes are preconfigured with threshold values that when exceeded indicate an event.

Event-driven IE approaches incur low power consumption and require low maintenance. Among the benefits of this class of approaches are: they reduce the amount of communication overhead by applying local filtering on collected data to determine whether to send new data or not; they implement local mechanisms to prevent multiple nodes reporting the same event; they exploit redundancy to reduce the number of false alarms; they allow timely responses to detected events; they are easy to implement and configure; they allow distributed processing at the node level or within a group of node to collaboratively detect an event; and they are suitable for time critical applications, e.g. forest fire monitoring or intrusion detection.

However, there are a number of limitations to the Event-driven IE. First, it is difficult to capture events of spatio-temporal characteristics. Second, detecting complex event may require non trivial distributed algorithms, which require the involvement of multiple sensor nodes[30]. Third, due to the fact that events occur randomly, some nodes generate higher rates of data than other nodes. This will lead to unbalanced workloads among sensor nodes. Fourth, it is not suitable for continuous monitoring applications, where sensed measurements change gradually and continuously. Finally, due to sensors measurement inaccuracies, Event-driven approaches may potentially generate false alarms.

In earlier studies, events were detected with a user-defined threshold values [31, 32]. In such approaches sensor nodes are preconfigured with a static threshold value. When the sensor node reading deviates from the pre-defined thresholds, this indicates an event, which triggers the node to convey it is data back to the sink. To overcome some of the inherent problems in the threshold-based event-detection, [28] have adopted the infrequent pattern discovery technique and developed a function for detecting an interesting event in the environment. The function is split into two phases, learning phase and event detection phase. In the learning phase, the function will learn the frequent changes (system errors) from the measurement series. In the event detection phase, a new pattern is built using the incoming measurement values in which a decision is made on whether the new pattern is a frequent (e.g., system events) or infrequent (e.g., event), then the infrequent changes will be reported to the fusion centre as a potential events. The proposed approach reduces the number of transmission, which extends the network life time. It is also characterised by low computational and time complexities. Due to its distributed nature the approach is scalable. However, high false alarms will be generated due to sensor measurement inaccuracies. Also, it is incapable of capturing complex events (e.g., herd of animals). This is because there is a spatial and temporal correlation in these types of events, which requires more efficient rules. They don’t have any mechanism to distinguish between errors and events. This will eventually affect the quality of the extracted data.

A decentralised, lightweight, and accurate event detection technique is proposed in [33]. The technique uses decision trees for distributed event detection and a reputation-based voting method for aggregating the detection results of each node. Each sensor node perform event detection using it is own decision tree-based classifier. The classification results, i.e.

detected events, from several nodes are aggregated by a higher node, e.g., a cluster head. Each node sends it is detected events, called detection value, to all other nodes in it is neighbourhood. The detection value will be stored in a table. Finally, tables are sent to the voter (e.g. cluster head), which in turns decides to make a final decision among different opinion. The decision tree approach provides accurate event detection and characterised by low computational and time complexities. However, the processing of data at the cluster head will introduce further delays in reporting an event.

A recent approach called (Event-Driven data Aggregation) EDAG is presented in [34]. EDAG is a clustering event-based approach that exploits the occurrence of events to form the clusters. The approach combines the advantages of cluster-based approaches with the structure-free approaches using two mechanisms. The first one is called a multi-level data-aware aggregation scheme. This mechanism increases the probability of spatial and temporal convergence of packets necessary for aggregation. The second mechanism is called an event-driven cluster head election which adapt the formation of the cluster according to the change in event sensing. The EDAG approach conserve energy combining packets and sending them as one packet. However, combining packets may result in reducing the information accuracy.

Another recent approach [35] introduced collaborative event-driven energy efficient Protocol (CEDEEP). CEDEEP takes into consideration the spatiotemporally correlated properties of event driven WSN. It assume that large number of nodes are being triggered by the same external event. The proposed approach reduces the number of nodes' transmissions according to the correlation among the sensed values in one-hop neighbours. During the network initialization, correlated areas are established. Also, the nodes memberships are defined based on calculation of the cross correlation coefficients. One node in each collaborative area is chosen to be a collaborative head (CH). When an external event occur, CH requests the node that has the highest energy in its area to send its sensing data. Upon receiving the report by the rest of sensor nodes, each node compares the received sensing value with its own. If the calculated difference is within the predefined limit, the node turns off its radio modules. Otherwise, the node sends its sensing data to the CH. CEDEEP incurs energy efficient operation of one-hop event driven WSN. However, the approach is still not applicable to multi-hop event-driven WSN.

2.3.2 Periodic IE

In periodic approaches to IE, a sensor node periodically generates a report from the physical environment to give the end-user its current status. The reporting period may be preconfigured or set by the end-user depending on the nature of the monitored environment and applications requirements.

Periodic approaches have the ability to enable arbitrary data analysis, they provide continuous monitoring of the sensor network to reflect environmental changes, they scale to handle millions of nodes (through aggregation), they extend network life time by sending nodes to sleep between transmissions, they can reduce congestion and improve system reliability by scheduling nodes to transmit at different times, they explicitly incorporate resource capacity, and highlights unused resources. However, there are a number of limitations to the periodic approaches. First, they are limited to specific set of applications where consistent changes occur across the network, e.g. agricultural applications. Second, a large portion of the returned data might be redundant and not useful for the end-user thereby resulting in wastage of resources. Third, nodes have to maintain global clock and deal with synchronisation issues. Finally, it is extremely difficult to define optimal time intervals.

In periodic IE, most of the published work in the literature is based on probabilistic models that attempt predict the next value that the sensor is expected to acquire. For example, Ken's [36] model exploits the spatio-temporal data correlations while guaranteeing correctness. It involves placing a dynamic probabilistic model on the sensor node and on the sink, and these models are always kept in synchronisation for periodic updates. An approach similar to Ken is proposed in [37]. In contrast to Ken, the approach exploits only the temporal correlation of sensed data and is based on the ARIMA (Auto Regressive Integrated Moving Average) prediction model. It places the model on the sensor nodes and the aggregator nodes (cluster head) to predict the next values. These models are always kept in synchronization for periodical updates. The approach is energy efficient since the number of transmitted massages is reduced. However, the forecasting can be badly distorted by outliers leading to wrong prediction. Also, the authors claim that further research will exploit the spatio-temporal correlation of sensed data.

Similar approaches to [36, 37] have been suggested in [30, 38]. In contrast to Ken, these approach uses dynamically changing subset of the nodes as samplers where the sensor readings of the sampler nodes are directly collected, while, the values of non sampler nodes are predicated through probabilistic models that are locally and periodically constructed. All approaches in [30, 36, 38] save energy by reducing the number of transmitted messages. However, the additional cost to maintain models synchronised is not negligible.

The authors in [39] propose a distributed algorithm to compute data-gathering schedule that aim to improve the lifetime of WSN by suitably selecting energy efficient data flow paths from various sensors to the base station. In order to gather and fuse sensory data, the algorithm performs the data gathering in rounds fashion. The base station re-computes the data gathering schedule and update the routing trees whenever half of the expected lifetime of the WSN is finished. This periodic rescheduling reduces the number of reschedule operation, leading to enhancing the WSN lifetime. This approach does not consider the situation where a forwarding node in the routing tree drains its energy and disconnect part of the routing tree. Furthermore, the approach does not consider the energy balancing among nodes, which limits its applications in large scale WSNs.

Another recent approach called DGA-EBCDS is presented in [40]. It is an energy-balanced connected dominating set (CDS) scheme aiming to effectively preserve the network energy and extend its life time. The approach assigns different roles to the network nodes to act as dominators and connectors. The assignment of the roles is based on their weights and the state of their neighbours' connections. Finally, the dominators and the connectors form the CDS. DGA-EBCDS reduces the energy consumption used to build the CDS as well as prolonging the network lifetime. However, the assignment of dominators should be more dynamic to avoid draining the dominators energy before the rest of the nodes.

2.3.3 On-demand IE

On-demand approaches to IE, typically involve request-response interactions (query) between the end-user or application components and sensor nodes. End users issue queries in an appropriate language, and then each query is disseminated to the network to retrieve the desired data from the sensors based on the description in the query. On-demand approaches provide a high level interface that hides the network topology as well as radio communication from end users.

However, there are a number of limitations to the on-demand approaches. First, most of existing query languages do not provide suitable constructs to easily articulate spatio-temporal sense data characteristics. Second, it is difficult to formulate queries using current languages that represent higher level behaviour, or specify a subset of nodes that have significant effect on the query answer. This may result in generating large amount of data of which big portion is not useful for the end user. Third, to the best of our knowledge, there is no published work that fully exploits all the potentials of different heterogeneous resources in WSN applications in a context-aware manner. Forth, approaches that take a database view of the network are inclined more towards the extraction of the reactive behaviour of the WSN and suggestions were made that the active database should be viewed as two end-points of the range of rule-based languages in databases [41]. Finally, though declarative languages are came into view in WSNs settings, the trigger that are the fundamental means for specifying the reactive behaviour in a database have not yet been maturely developed.

On-demand IE systems, applies techniques used in traditional database systems to implement IE. A query is sent to the network and data is collected according to the description in the query. COUGAR [42] was the first project that attempted to introduce the concept of WSN as a distributed database. It allows the end user to issue a declarative query (SQL) for retrieving information. The authors introduced a query layer between the application layer and the network layer. The query layer comprises a query proxy, which is placed on each sensor node to interact with both the application layer and the networking layer. The goal of the query proxy is to perform in-network processing. In-network processing increases efficiency in terms of power consumption, and reduces the amount of data that needs to be sent to the gateway node. The user does not need to have knowledge about the network, or how the data is retrieved or processed. However, COUGAR is incapable of capturing complex events, e.g. of spatio-temporal nature, or a produce queries that targets only a subset of the network [43].

A similar approach to COUGAR is proposed in [44]. TinyDB is a query processing system, which extracts information from the data collected by the WSN using the TinyOS operating system. TinyDB maintains a virtual database table called SENSORS. It disseminates the queries throughout the network by maintaining a routing tree (spanning tree) rooted at the end point (usually the user's physical location). Every sensor node has its own query processor that processes and aggregates the sensor data and maintains the routing information. TinyDB is extensible and complete framework with effective declarative queries. In-network processing reduces the amount of data that is required to be sent to the sink, thus, energy consumption is reduced. However, data does not include the georeferencing of sensor nodes for spatial queries, and tight correlation among routing and queries.

A new type of queries called region-based queries was proposed in [45]. The authors proposed a framework called REQUEST+ to process this type of queries. They divide the monitored area into regions covering the entire area. The SEC (Smallest Enclosing Circle) index is used to construct the regions for every possible combination that can be located in a circle of a given diameter. REQUEST+ can find some interest regions satisfying several conditions. Consequently, the query could deal with every possible region where sensor nodes are deployed. This makes it possible to have a macro view of the monitored area. However, REQUEST+ works only with static WSNs.

2.3.4 Hybrid IE

A hybrid approach is an approach that combines the functionality of two or more algorithms from different IE categories. Hybrid approaches aims to minimise the effect of the disadvantages of individual IE categories described above.

Many hybrid approaches to IE have been recently proposed in the literature. In [46], the authors proposed a hybrid protocol that adaptively switches between periodic and event-driven data collection. A sensor node is triggered to detect an event of interest, and from the point when an event detected to the point when the event becomes no longer valid, the protocol switches to behave as a periodic protocol. During this period sensor nodes continuously report data to the sink. This protocol reduces unnecessary data transmission and minimise event notification time. However, it is not guaranteed to work well for all applications due to limitations of the PAD algorithm, such as if sensor nodes detecting an event are located at the border between clusters, those nodes in other clusters can be included only when clusters at the same level have used time-driven data dissemination.

More recently, in [47], the authors proposed a hybrid framework similar to [30], [36], which deploy both of event-driven and on-demand approaches to IE. The idea is to process continuous group-by aggregate queries, and to allow each sensor node to check whether sensor readings satisfy local predicates based on predefined thresholds. Then, nodes send only data that satisfy local predicates to their cluster heads, which in turns process the data to answer the query as accurate as possible. The proposed hybrid framework is able to target a subset of the network by using the group-by clause. It reduces communication cost by using one dimensional haar wavelets. However, it introduces a delay in reporting events since the data is processed at the cluster head.

In a recent publication [48], the authors proposed a hybrid data gathering approach that dynamically switches between event driven and periodic IE techniques. Their aim is to extract more accurate information than event driven and save more energy than periodic techniques when deployed individually. The basic idea is that as soon as an event is detected, the approach switches to periodic data reporting. Nodes in close proximity to the node(s) detecting the event are notified. After being notified about an event, neighbouring nodes proactively engage in periodic data reporting. The approach provides continuous monitoring of the monitored conditions. However, the choice of nodes to be engaged in the information gathering is simply based on a predefined number of hops. This often results in involving irrelevant nodes, which leads to wastage of resources and potentially less accurate information.

2.4 Mobile-oriented IE

Recent applications of WSNs (e.g. in medical care) make use of mobile sensor nodes to improve their performance. However, mobility poses new challenges to IE researchers including: increased data loss and delivery delay due to intermittent connectivity; lower throughput due to low channel utilisation; frequent topological changes; amongst others. Few IE solutions have been proposed to deal with some of these challenges. These solutions can be categorised according to the type of the mobile entity as follows:

2.4.1 Mobile Nodes

Researchers developed approaches specifically designed to extract information from WSNs where sensor nodes are mobile. These approaches can be classified according to the purpose of mobility as follows:

Coverage: Some approaches, e.g. [49-51] move nodes to provide better coverage by filling in holes in sensing coverage. They relocate redundant nodes to areas where node density is low to improve the accuracy of extracted information. However, this type of algorithms needs complex relocating models to calculate the moving nodes' trajectories and their new locations.

Mobile Environments: Approaches such as [52-54] move mobile nodes to monitor moving objects, e.g. wildlife monitoring or offline monitoring of vehicle fleets. Mobile nodes are mounted on the monitored objects to log the sensed information on their memories for later analysis. When the mobile nodes move within radio range of the sink, they upload the logged information to it. However, these algorithms have several drawbacks. They are designed for specific types of applications that are not time critical. Moreover, in most situations animals or vehicles move in groups, the nodes density will not be disseminated in an effective way. Consequently, sinks located in dense areas will be overloaded, leading to increase latency and data loss.

Relay Nodes: In some approaches, mobile nodes are used as relay nodes besides their sensing duties. Mobile nodes can be used to carry information from the sensing field and deliver it to a fixed sink. In these algorithms, mobile nodes send data over a short range communication (from a sensor to the relay node) that necessitates less transmission power. Mobile nodes can move randomly, like in [13, 55-57] on fixed trajectories, as in [58-61] or based on occurrence of an event of interest, as in [62, 63]. However, these approaches introduce considerable delays on data delivery and may potentially miss some important information in case of frequently changing phenomena. To overcome some of these issues, some recent researches have proposed moving the base station itself rather than moving relay nodes.

2.4.2 Mobile Data-sinks

The largest set of IE approaches that has been proposed in the literature suggests using mobile sink for data collection and analysis. The mobile sink moves towards isolated nodes according to a particular trajectory to collect their data. Based on the nature of the trajectory, the sink mobility approaches to IE can be further classified into three classes:

Fixed Trajectory, e.g. [11, 64, 65] assumes that the trajectory is fixed such as in roads. In these approaches data or information is conveyed to rendezvous nodes, which are closer to the trajectory, which is then cached until the mobile sink passes by and picks it up. The mobile sink can perform further processing on the received information or data. Hence, IE is achieved periodically.

Dynamic Trajectory, e.g. [66, 67], assumes that the trajectory is dynamic, different algorithms are used to calculate the trajectory. In these approaches IE is performed according to a pre-computed schedule, e.g. [68], or according to event occurrence, e.g. [69]. Approaches such as [69], propose mobility models that moves the sink node according to the evolution of the current events.

Random Trajectories, e.g. [10, 70, 71] assume that mobile sink moves randomly in the sensor field. Mobile sinks are mounted on people, vehicles, or animals moving chaotically to collect information of interest around the network.

Although mobile sink strategies are desirable due to their simplicity, they suffer from some drawbacks. As the sink moves through the sensor field, it causes high control overhead to find a route to the sink and send packets to it. This may possibly dissipate the energy saved by using the mobile sink strategy. Moreover, constantly relocating the sink introduces significant delays on data or information delivery. Despite the extended coverage, these approaches lack scalability; as the network grows in size, the nodes located close to the mobile sink's trajectory get overloaded leading to energy depletion in the network, disconnections, and bandwidth bottlenecks. Finally, the trajectory calculation is a complex problem.

2.5 Attributes of mWSNs Information

In this section, different attributes for mWSNs information are identified and defined. Most of these attributes have been used constantly in the literature. Inconsistencies in the definitions have led to problems in measuring the quality of extracted information. Hence, standard definitions are needed in order to facilitate the comparison of different approaches. However, we only choose the attributes that are relevant and useful in evaluating the quality of extracted information from mWSNs.

Accuracy: Accurate information allows the end user to take correct actions by providing a realistic reflection of the actual sensed environment. The term *accuracy*, also known as *correctness* has been widely used in the Quality of Information field [88]. The authors in [89] define accuracy as the level of detail (precision) in the sensed data. Similarly, in [88], accuracy is the degree of correctness, which provides the level of detail in the network. However, the above definitions do not differentiate between data accuracy and information accuracy. This problem has evolved from the confusion of the terms data and information. Moreover, the level of details is controlled by the user and missing a detail does not necessarily affect the reported information accuracy. The accuracy of information is not only achieved by the accuracy of data, data processing models can significantly affect the information accuracy. We adopt the definition in [90] as it covers exactly what we mean by information accuracy in this work. Information accuracy is *the degree of deviation of the extracted information from the actual current state of the monitored environment.*

Completeness: In the literature, the definition of information completeness is linked to data integrity, which is the absence of accidental/malicious changes or errors in some data. In [88], information completeness is defined as the characteristic of information that provides all needed facts for the user/application during the process of information construction. The authors in [91] define information completeness as a measure of the fraction of all generated reports that arrive to the end user. Each of these definitions refer to the information as complete information when the delivered information represents all the sensed data without any diminution. In other words, they define completeness as the ratio of the received reports over the sent reports. However, if part of the environment is not covered by sensor nodes, then this part is not represented in the extracted information. Therefore, it is important to incorporate the sensing converge in the definition of information completeness. To include sensing coverage, we re-define completeness as the *degree of obtaining all desired information that represent the actual current state of the complete monitored environment.*

Affordability: Affordability refers to the cost of collecting sensed data [89], i.e., it is the expensiveness of information. In [88], affordability is the characteristic of information associated to the cost of measuring, collecting and transporting of data/information. *We define affordability as the ability to afford the cost of information in terms of resource utilisation from the stage of sensing the environment to the stage of extracting the required information.*

Timeliness: Information timeliness is a crucial and decisive criteria in time critical applications. In [89], timeliness describes how timely the data is provided to be useful to the end users or applications. To incorporate the scale of a multi-hop network, timeliness is measured as the time normalised against the average time for a single-hop along the shortest path from a sensor to the sink [92]. In the above definitions, timeliness is accommodates different types of delay including: loading, propagation, queuing, and processing delay. However, in mobile-oriented approaches, the mobile sink has to travel to a specific point to collect information. This introduces considerable delays on data delivery; therefore, the time the sink spends travelling toward sensor nodes should be also considered. We extend the definition in [88], *timeliness is an indicator for the total time required when the first data sample generated in the network until the information reaches the user for decision making. This includes the time that the mobile node spends travelling towards the target nodes.*

Availability: The term *availability* has been widely used in computer networks as a primary QoS measure. Network availability refers to the overall up-time of the network, or the probability that the network is available to use [93]. In [94], availability is defined as the fraction of time that a network is able to provide communications services. However, we are not only concerned about the availability of the nodes, communication links or the network in total; but also in the availability of information. The network generated data could contain the desired information but the inability of the user or the lack of the powerful IE tools leads to incomplete information. Moreover, nodes mobility is an important factor that impacts availability of information. If the node that carries the desired information is not in the vicinity of the mobile sink, information from that node will be unavailable. Therefore, many factors impact the information availability including: sensor nodes; communication links; sensors generated data; and IE techniques. An absence of one of the previous factors leads to unavailable information. We define information availability as *the fraction of time the network is able to acquire and deliver the end user's desired information.*

Validity: information validity refers to whether the information is useful to the end user or not. There are many factors that could result with invalid information. For instance, information based on un-calibrated sensor readings, corrupted packets, or noisy data is unbeneficial and even confusing to the end user. Furthermore, in time critical applications, delaying the information invalidates it. For instance, in target tracking application, information could be received indicating that the target is in location x, but when the information was received the target has moved to location y. *The extracted information is valid if its content is entailed by its data.*

2.6 Mobility Benefits

Although mobility requires a lot of management, it has advantages over static WSN such as: better energy efficiency [8], improved coverage [49], enhanced target tracking[53], greater channel capacity [72], and enhanced information fidelity [73].

In most of WSNs applications, nodes location is important as its information is useful for coverage, routing, location services, and target tracking [74]. An appropriate node deployment strategy can effectively reduce the network topology management complexity and the communication cost. Sensor nodes can be placed on a grid, randomly, or surrounding an object of interest [75]. In applications where nodes need to be deployed in harsh or remote environments, nodes deployment can not be performed manually or accurately. Therefore, if a node runs out of battery, the data from the dead nodes would be lost, which introduces a new problem by negatively affecting the accuracy and completeness of the extracted information. Some approaches tried to solve the problem by exploiting node redundancy. This class of approaches requires dense node deployment, which increases the system cost and management complexity. Node mobility presents effective solution to the above problem at low cost. Mobile nodes can redeploy the network by connecting disjointed areas created by dead nodes without the need of very dense deployment. Some approaches, e.g. [49, 50, 76], move nodes to provide better coverage by filling in holes in sensing coverage. They relocate redundant nodes to areas where node density is low. A complete coverage results in high information accuracy and completeness as every point in the environment has data to represent it. Furthermore, relocating nodes to substitute dead nodes helps in tolerating node failure. That maintains high information availability and completeness.

Unfair coverage caused by random nodes distribution results with high traffic load in some parts of the network. In traditional static networks, the nodes located around the sink become bottlenecks due to the many-to-one multi-hop communication. Bottlenecks introduce information delivery delay and causes energy depletion in some parts of the network or could even lead to the network partitioning problem [77]. This decreases the level of information completeness and availability. Furthermore, the probability of error increases with the number of hops that a packet travels over [26], which lower the information accuracy. mWSNs are believed to provide more balanced energy consumption than static networks [26, 78, 79]. Nodes mobility offers a solution by moving nodes as needed to optimise the network performance.

Moving the sink to data sources or moving the sensor nodes towards the sink is one way to avoid the communication bottlenecks. Approaches such as [10, 11, 69, 80] suggest moving the sink close to data sources to perform data collection and analysis. This has been shown to be an effective way of reducing network congestion levels and relaying information in partitioned networks. Keeping the network connected leads to better sensing coverage, and hence maintains the higher information completeness level. Furthermore, moving the sink closer to sensor nodes helps conserve power by reducing the bridging distance between the node and the sink [81]. This also increases the performance of the network by saving retransmission bandwidth [75]. Moreover, information accuracy also increases due to the fact that the probability of error in the data decreases when decreasing the number of hops [26]. Other approaches, e.g. [62, 63, 82], suggest using mobile nodes to collect data from the monitored field and deliver it to a fixed sink. In these approaches, mobile nodes send data over short range communication, which involves less transmission power. This leads to reduced energy consumption and communication overhead. Since the cost of transporting data is reduced due to using single hop communication, the total cost will be reduced, resulting in more affordable information. Moreover, introducing mobility leads to balance the data load to be transmitted from sensor nodes to the sink, which helps in buffer overflow prevention [82, 83].

However, the above mentioned approaches have some drawbacks: First, some nodes could have data to send but the mobile sink or data collector is not around, this negatively impact the timeliness and validity of information. Third, moving nodes usually consumes more energy than sensing, computation, and communication. The mobile sink or data collector could move towards some nodes which have no data to send, this would be waste of energy and time. Therefore, if nodes' movements are not planned in an efficient way, they could deplete the limited node's energy; which can diminish the gains in quality of information.

In some mWSNs applications [52-54], e.g. wildlife monitoring or offline monitoring of vehicle fleets, nodes mobility is provided by external source, i.e., the monitored object carries the node. Often, the node is mounted onto monitored object in order to collect sensor measurements and store it into their memories for later analysis. When the moving object runs across the sink’s coverage area, it transmits the information to the sink. When the sensor node is attached to the monitored object, that object stays always within the sensing coverage of the sensor node. This protects the level of information accuracy. However, these algorithms do not work efficiently with time critical applications.

2.7 Mobility Challenges

WSNs are characterised by dynamic topologies due to the insertion of new nodes and death or failure of existing nodes. Mobility makes the topology more dynamic due to continuous node movement. This introduces new challenges including intermittent connectivity, data latency, and high devices cost and size. In some situations where links disconnections cannot be recovered quickly, the information become unavailable for the failure duration. Also, nodes buffers may become overloaded and hence information can be lost. Consequently, the level of completeness and validity of extracted information will be reduced.

At the same time , nodes mobility often increases the end-to-end delay on data delivery [73]. In mobile data collectors approaches, the collecting node moves towards isolated nodes to collect their data, while other nodes cannot transmit their data until a collecting node is within their transmission range. Consequently, the information about the sensed environment becomes outdated, which possibly lead to extract invalid information.

Localisation: Many WSNs applications rely heavily on node's ability to establish position information. Location can be defined in multiple ways: Geographical location, which is the spatial coordinates (longitude, latitude); Relative location, which is the location of a subject in relation to another node (e.g. between node x and node y); Semantical location also known as the symbolic location, which represents nodes relationships in a space. It carries semantics about locations that have a certain meaning for users or applications, e.g. soldier near a tank. The process of obtaining the position of a sensor node is referred to as localisation. Localisation has been identified as an important research issue in the field of WSNs. Localisation algorithms use various available information from the network in order to calculate the correct position of each sensor node. The location information is a key enabler for many WSN applications, e.g. target tracking, and it is useful for managing deployment, coverage, and routing [53]. Location information enables binding between extracted information and physical world entities. If the positions of sensor nodes can be determined more accurately, it will leverage the achievement of meaningful use of extracted information. The location of an event can be determined by knowing the location of nodes that report it. Thus, the locations of nodes that carry information of spatio-temporal nature need to be considered. Obtaining the nodes' locations helps in identifying nodes that carry data relevant to a certain piece of information.

In mobile environments, locations of nodes keep changing over the time. This introduces additional challenges that need to be addressed. (1) Localisation latency: the localisation algorithm should take minimal time to cope with mobility speed. For instance, if a node is moving at speed of 10 meter per second and the localisation algorithm needs 3 seconds to complete execution, the node would be 30 meters away from the calculated location. In the previous example, if the radio range needed to keep the node connected less that distance that the node has travelled, the node would be lost; The information from that node will be inaccurate or even might be unavailable. (2) Increased control messaging: managing nodes location information requires communications and transmission of control packets. When node location changes frequently, the control packet overhead will be increased leading to higher energy consumption. This negatively affect the affordability of extracted information.

Trajectory Calculation: In mWSNs, the trajectories of nodes motion can be random, fixed, or dynamic. Some approaches, e.g. [10], assume that mobile nodes are mounted on people, vehicles, or animals moving chaotically around the network. Due to the fact that nodes cannot communicate unless they are in the radio range of each other, all nodes in the network need to keep sending periodic discovery messages to keep their routing tables updated. Transmitting a large amount of discovery and control messages consumes more energy. Furthermore, As the nodes needs to be aware of all changes in the network topology, they cannot switch their transceivers to sleep mode in order to conserve energy.

Approaches such as [11] propose mobility models that moves the sink or data collector in a fixed trajectory. Data or information is conveyed to rendezvous nodes that are closer to the data collector trajectory, where it is cached until the mobile data collector passes by and picks it up. Sensor nodes can turn their transceivers off when the mobile data collector is away.

However, in these approaches the fixed trajectory need to be defined. This needs a complex algorithm to calculate the most appropriate route that the node should follow.

When the trajectory is dynamic as in [69], nodes motion can be according to pre-computed schedule, or based on occurrence of an event of interest. However, calculating a dynamic trajectory is a complex problem, since it should satisfy the spatial and temporal constraints of the monitored phenomena. Knowing the trajectory of mobile nodes is very important as it helps to predict the nodes' locations. Therefore, this helps to plan for more efficient data collection leading to save more energy and maximising the network lifetime. Sensor nodes could be pre-configured with a sleep-wake cycle that is based on the location of the mobile node; a node goes to sleep when the mobile node is out of its radio range.

Velocity Control: Commonly, in mWSNs, nodes move in constant speed [84, 85]. Velocity of the mobile node effects the information delivery time. However, some data collection approaches, e.g. [12, 86, 87], assume that the speed of the mobile nodes is variable and also has different accelerations in order to optimise the movement of mobile nodes to reduce the information delivery time. The velocity is controlled by the task that the mobile node performs. If a mobile node performs data collection task, its velocity should be low compared to a mobile node that performs fire sensing task. Controlling velocity helps in utilising the available resources and results in more efficient WSN system. For instance, consider that there is a senor node that generates a reading every one minute, and there is data collector visits that node every 15 seconds; in this case, a lot of the data collector's energy is being wasted. However, by optimising the speed of the mobile node to best match the data generation rate, the data collector visits that node every one minute; hence, the data transmission of the network will be more efficient.

Moreover, determining the velocity of mobile nodes is crucial in many of mWSNs applications. For instance, in tracking moving targets, the mobile sensor node should stay close to the target in order to maintain constant coverage.

Another important mWSN feature is the cost of the system. Large-scale WSN applications comprises hundreds or thousands of low cost nodes spread across a wide geographical area; hence, per-node monetary cost is important. Nodes mobility, greatly increases the cost of each node as they are equipped with an electric motor to control their motion. The cost of the electric motor is non-negligible. To power these motors, energy can be obtained either from the node's battery (which has undesirable consequences) or from additional battery, which further increases the cost and size of the device.

2.8 Description of the Approaches Used in Performance Evaluation

This subsection provides the details of a mobile data collection protocol called MDC/PEQ [95] and a static-oriented hybrid data gathering protocol [48]. We use these two protocols, as the closest rivals to our proposed solution, as a baseline for performance comparison. The reasons for the choice of these protocols are discussed in the evaluation Chapter 6.

2.8.1 An Overview of MDC/PEQ

This subsection summarises the main features of the MDC/PEQ algorithm [95]. MDC/PEQ is a mobile data collection algorithm designed for applications that require fast response time. It aims for a low-latency and reliable mobile data gathering for delay-sensitive applications. Their network structure consists of three layers: The bottom layer contains a randomly scattered static sensor nodes that perform the sensing tasks. The middle layers formed by mobile data collectors (MDCs) for collecting data from the bottom layer and pass it to the static sink located the top layer.

In the MDC/PEQ protocol, MDCs broadcast configuration beacons periodically. Initially, when the beacon is received by a sensor node, it joins the MDC's cluster and update its routing information to relay data packets to the corresponding MDC. For connection reconfigurations (handoffs), sensor nodes use the signal strength of the beacon as well as the hop level. The hop level is the number of hops a node is away from an MDC. Sensor nodes can relay data to MDCs in multi-hop fashion, they do not need to wait until an MDC is nearby.

After network deployment, MDCs start broadcasting $\mathrm{INIT\_SETUP}$ messages with hop level $\mathrm{h} = 1$. Upon receiving the $\mathrm{INIT\_SETUP}$ message, a sensor node compares the field h with its hop level h_s. If $h < h_s$, the node will connect to the MDC that sent the $\mathrm{INIT\_SETUP}$ message and assign $h_s = h$. The node rebroadcasts the $\mathrm{INIT\_SETUP}$ message with hop level incremented by 1. This results in tree routed clusters and each node will have routing information to relay data packets to the sink.

Cluster Configuration

This phase starts when an MDC sends a beacon message to advertise its presence. Beacon packets contain $\mathrm{MDC\,ID}$, TTL , and hop level h.. Initially, all sensor nodes have their hop level is set to -1. When a sensor node, which has no active connection to an MDC receives a beacon message, it establishes a route to the corresponding MDC and assigns $\mathrm{h_m} = \mathrm{h}$. Next, the sensor node forward the original beacon message. By the end of this phase, the network clusters are configured and the network is ready for operation.

Route Maintenance (handoff)

MDCs broadcast beacon messages every $\mathrm{T_b}$ seconds. A reception threshold Rx_{thresh} is set on nodes based on their communication range R. Each beacon packet is received at a specific signal strength SS. According to the SS of the beacon packet, the sensor node responds according to Algorithms 1:

Algorithm 1: MDC/PEQ Algorithm

```
Begin
If (SS ≥ Rx_thresh)
     IF(sender_MDC == current_MDC)
          SS = received_SS
     ElseIf h_m ≤ h then:
          //node drops the packet
     else
          curr MDC= sender_MDC
Else
     If (sender_MDC== current_MDC)
          //disconnect from the current cluster
          //Broadcast LSS message
     else
          //Node drops the packet
End
```

Data Transmission

Each node holds two paths to the data sink, one is direct multi-hop to the sink, and the other is through an MDC. When a node has data to be transmitted, it uses the shorter path, i.e., the path with the smaller number of hops. In case of both paths have the same number of hops, the node relays its data through the direct path to the sink. If the node does not belong to any cluster, it uses the direct route to the sink to transfer its data. If an intermediate node has a route to an MDC, it forwards the data to that MDC. This approach achieves good timeliness as nodes do not wait for an MDC to move nearby.

2.8.2 A Hybrid Data Gathering Protocol

The authors of [48] puts forward a static-oriented, hybrid IE approach that dynamically switches between event-driven and periodic data gathering techniques. The design of [48] considers the spatio-temporal domains for adapting a suitable IE technique. When an event is detected, [48] switches from event-driven to periodic data reporting. The node(s) that detected an event notifies its neighbours about it. When an event becomes invalid, [48] reverts back to the event-driven operation mode. This method of operation has proven to improve the QoI at low energy consumption. This is due to the fact that only nodes carrying significant data participate in the IE. Hence, the authors claim that their work extracts more accurate information than traditional event-driven approaches and conserve more resources than periodic-driven techniques when deployed separately.

The operation of [48] is based on two algorithms, namely, parameter-based event detection (PED), and parameter-based area detection (PAD). The former is to determine in a timely manner when to switch between the event-driven data reporting technique and the periodic-driven data reporting technique, and vice versa. The later is to determine the sensor nodes in close proximity to the event and to engage them in the data reporting process.

PED Algorithm

The algorithm uses multiple variables to control the 'aggressiveness' level for changes in the data reporting technique. These variables include a threshold value, a threshold variable, and two counter-variables (p and q). The threshold value is the point that must be exceeded to be reported to the sink. The threshold variable is the average of a sensed value over a defined time interval. The counter-variable p is the number of consecutive time intervals with increasing (for starting the periodic data reporting technique) or decreasing (for stopping the periodic data reporting technique) slope of the threshold variable. The counter-variable q is the number of consecutive time intervals that the threshold variable is above (for starting the periodic data reporting technique) or below (for stopping the periodic data reporting technique) the threshold value, regardless of slope. Two pairs of parameters are provided to the algorithm, P_{start} and Q_{start}, and P_{stop} and Q_{stop}, such that $P_{start} \leq P_{stop}$ and $P_{stop} \leq Q_{stop}$.

Initially all nodes run event-driven data reporting. Every sensor node executes the PED algorithm and calculates the average of the sensed values and updates the counter variables p and q, accourdingly. If $(p \geq P_{start})$ or $(q \geq Q_{start})$, the sensor node switches to periodic data reporting. If $(p \geq P_{stop})$ or $(q \geq Q_{stop})$, the sensor node switches back to event-driven data reporting. The P_{start} and P_{stop} change their values only when the sensed data increases or decreases monotonically. This enables a rapid response to a change in the environment, and prevents a response to a transient change due to, e.g., sensor malfunctioning. In case the P test fails, the Q test is applied which does not require the monotonic increase or decrease in the sensed values.

PAD Algorithm

After switching to periodic data gathering, the sensor node broadcasts a message to engage neighbouring nodes in the IE process. The algorithm is based on a pair of configurable parameters, time-to-live (TTL) and valid time (VT). TTL is the number of hops that the broadcast message can be conveyed to before it is dropped. Hence, the PAD is based on the logical assumption that the closer a sensor nodes to the location of an event the more likely it is to carry relevant data. The VT determines the period of time that the neighbouring sensor nodes should run periodic data gathering. It is important to point out that VT is only used with nodes that were switched to periodic data gathering by TTL. It is necessary to use VT with these nodes as they may not yet detect the event, and hence, they would switch back to event-driven data reporting and reduce the readiness to extract information.

2.9 Summary

IE is considered as a costly task because it involves the collection and processing of often large amount of unstructured or semi-structured sensor data. There is a wide body of literature on static-oriented IE; we refer interested readers to [96], for a recent survey that provides a comprehensive review to IE approaches. However, there has been few attempts to address the problem of IE from mWSNs, e.g. [10-12]. These approaches dealt with mobility as the need arises and do not attempt to deal with the fundamental challenges and variations introduced by mobility on the WSNs. We believe that there is a need to adapt the mobile environment to work with static-oriented IE approaches. By achieving this, the benefits offered by nodes mobility can be exploited to further increase the performance of the static-oriented IE approaches.

Chapter 3

Motivation Scenario and System Specifications

3.1 Introduction

This chapter describes a real-life application scenario, battlefield surveillance, where environmental monitoring is carried out using a combination of static and mobile sensor nodes. Using this application, we aim to demonstrate the practicality and reliability of our presented network model. Moreover, since surveillance applications impose strict requirements on the quality of the returned information; this application allows us to extract and analyse the challenges posed by such a complex system. The findings from this stage are used to derive the desired system's specifications and its design principles.

The rest of the chapter is organised as follows: In Section 3.2, introduces the application scenario and the network model. Section 3.3 presents system specifications and its design principals. Finally, Section 3.4 concludes the chapter.

3.2 Application Scenario

The research in this project is driven by real-world application scenarios. This section presents a motivating scenario for which a decentralised model of IE from WSNs with mobile nodes is essential. The chosen application scenario is battlefields surveillance. This class of applications includes collaboratively interacting and detecting various entities in the monitored environment such as: enemies movement, militants activities, target acquisition, soldier localisation, healthcare monitoring, etc. Battlefield surveillance is considered one of the most important force multipliers which are essential for winning the future battles. This scenario place complex demands on the IE service and yields more than one of the goals and benefits of mWSNs outlined in Section 2.5.

Battlefields surveillance is selected because it is general enough to be applied in other applications with mobile nodes, e.g., wildlife monitoring, border security, etc. WSNs can provide real time meaningful location and status of various enemy assets and related infrastructure. An effective usage of WSNs in battlefields can save many soldiers' lives and it is essential for gaining success in operation of war. Battlefields surveillance is based on a dynamic network topology aiming to provide the ability to extract high quality information. Furthermore, this application includes a mixture of static and mobile nodes, which is considered a practical approach for the deployment of advanced surveillance systems [6, 7]. It also has many different mobility entities such as: soldiers, armoured vehicles, and tanks. Battlefield surveillance presents real-world examples that instantiate this scenario.

Battlefield surveillance systems are complex, multidimensional and highly dynamic. This offers an comprehensive set of conditions and settings to apply any proposed solution for the challenges it presents. Its complexity also provide a suitable environment to test mobile-oriented IE approaches. The presented scenario highlights the motivations for deploying mobile nodes and the challenges that need to be addressed to enable rapid and energy efficient IE over dynamic environments. The requirements identified through this scenario will be used to design a general purpose IE framework for hybrid WSNs. On a later stage, the framework will be tested in this application environment to measure its performance and applicability to real world data sets and settings.

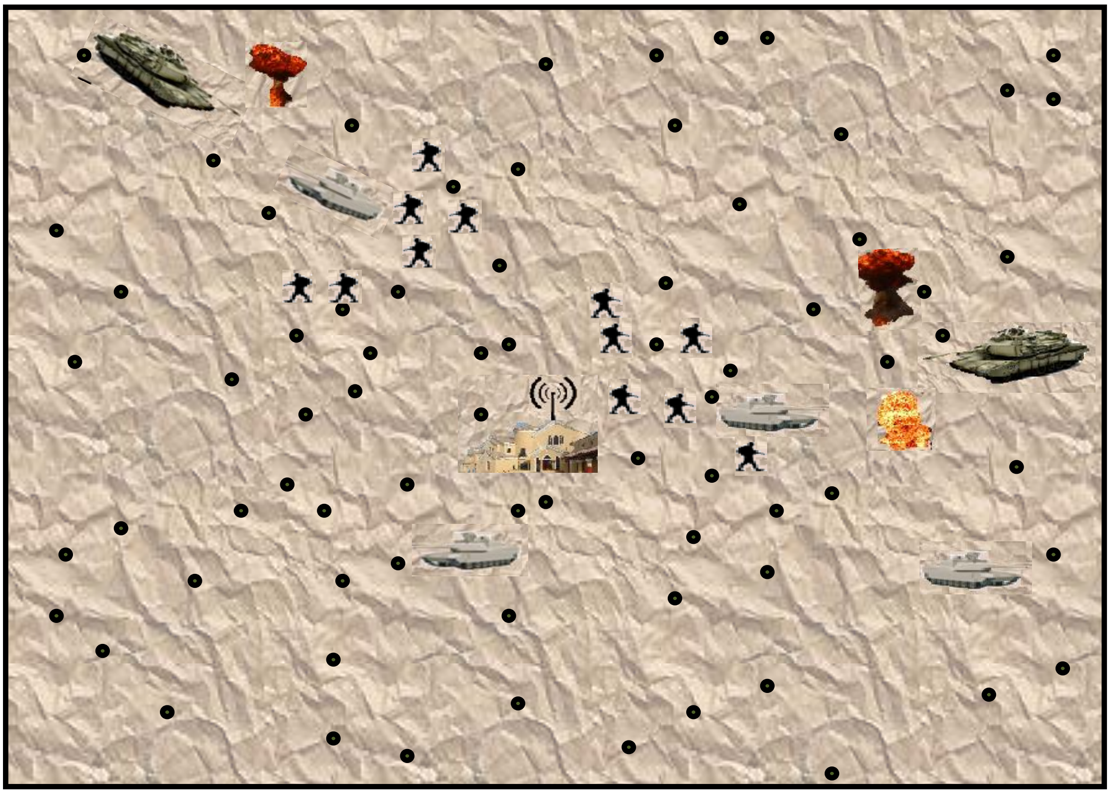


Figure 3.1: Battlefield surveillance scenario

We consider the hypothetical battlefield surveillance scenario as depicted in Figure 3.1. The figure shows a battlefield that has different entities including: armoured vehicle soldiers, command and control centre, and static sensor nodes deployed in the battlefield, all belonging to one military force. The area around the command and control centre is covered by a sparse WSN used for target detection and surveillance. A group of militants started an attack against this centre. Accordingly, two special forces, each consisting of an armoured vehicle and a group of soldiers, were despatch to engage with the militants. In addition, each soldier body armour is equipped with a sensor node used for determining his location, monitoring his health conditions and collecting information about the environment. Moreover, the armoured vehicles are equipped with sensor node and long range wireless communication facility to communicate with the command and control centre. This type of sensor node has unlimited power supply, which is provided by the carrying vehicle.

The aim of deploying the WSN is to provide high quality information in the battlefield, and hence, the ally commanders will successfully achieve their mission goal. Ally soldiers need to be informed not only about their positions but real time information on what is behind an obstacle and what are the adversary's behaviours and movements. Furthermore, the commanders in the headquarter need accurate information about everything happening in the battlefield in order to make the right decisions and give the appropriate orders to combatants.

The above scenario combines the advantages of static and mobile entities of WSNs. Static nodes are deployed from the air in the area around the command and control centre. There are two groups of mobile nodes, these can travel to anywhere at any time with variable speeds. The first group of mobile nodes are the ones that are attached to soldiers. These nodes can travel with speeds between 0 mph and 10 mph [97]. The second group, called data collectors, consist of the powerful nodes attached to the armoured vehicles, their speed varies between 0 mph and 40 mph [98]. The collected data from the all nodes, i.e., static nodes on the ground and mobile nodes attached to soldiers, is sent to the data collector nodes. Data collectors communicate with the base station in the command and control centre to provide real time information. Moreover, they may carry special sensors, e.g., cameras, to sense additional modalities.

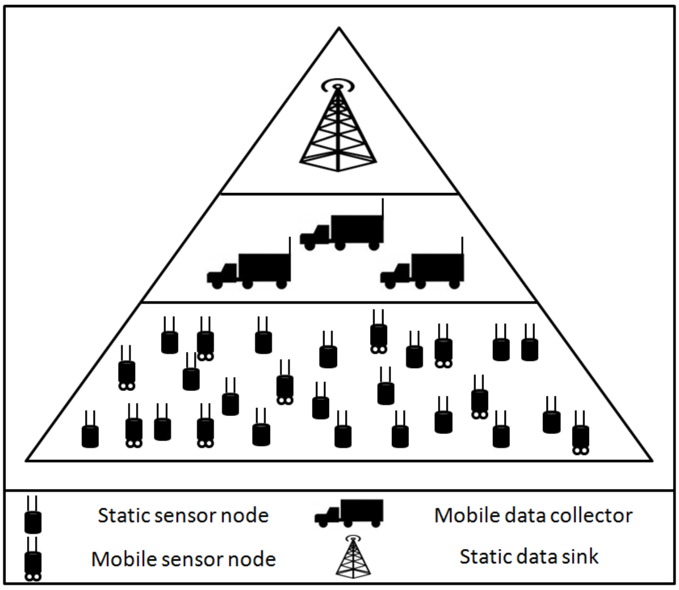


Figure 3.2: Three-tier network structure

In the above scenario, node mobility appears at various extents. Some parts of the network are purely mobile, other parts are mixture of mobile and static nodes. Therefore, the deployed sensor network can be illustrated as a three-tier network structure, Figure 3.2. The static sensor nodes on the ground and the mobile sensor nodes attached on soldiers are grouped in one category, called tier 1. The data collectors attached on tanks are grouped to form tier 2. The headquarter base station represents tier 3.

In an abstract logical model of a distributed computing application, we map different devices to a three-tier network structure, see Figure 3.2, based on their physical capabilities. Static nodes deployed on the ground and mobile nodes carried by soldiers are allocated to the bottom tier of the network, tier 1. Data collectors are form tier 2. Tier 3 contains the base station at the command and control centre.

The bottom tier hosts the majority of the network nodes. At this tier, static nodes perform sensing tasks and act as a communication architecture that mobile nodes join as they travel across the network. Mobile nodes increase the sensing coverage in the areas of high interest, where the soldiers are, as well as collect and forward data from static nodes over high speed, low delay links. The middle tier hosts a small number of mobile data collectors, which have long range communication radio and are considered power rich devices. The physical location of the data collectors in the monitored environment determines the area of the 'highest priority' to end users. Therefore, their powerful physical capabilities is exploited in extracting information with high quality at a low cost. Finally, the top tier of the hierarchy hosts the fixed data station. At this tier, information received from different sources is fused and presented to the end users at the command and control centre who can in turn return feedback to the troops in the battlefield.

3.3 System Specifications

One step towards achieving the ultimate aim of this research, was to identify and analyse a complex application scenario, where IE from mWSNs is mission critical. A battlefield surveillance application scenario that incorporates different mobility models and types is presented in Section 3.2. This application identify and analyse a complex application scenario. This section presents the main specifications that has to be present in the desired system.

Quality of Information

In battlefield surveillance, timely and accurate target intelligence is vital for producing effective responses to battlefield events. For instance, a short delay in receiving information about the current location of the enemy rocket-launch system means less chance to intercept and destroy this system. The same consequences will be reached if timely, yet inaccurate, information about the enemy's location is received by the ground control station. These two simple examples demonstrates that the QoI is one of the core factors that determines the success or failure of any WSN system. Typically, WSNs are error prone and various applications impose different restrictions on the quality of the information presented to end users. Generally, the QoI can be controlled, a higher QoI is possible at extra cost due to the additional data collection and processing. In this work, QoI is measured using a set of information attributes, these are presented in Section 2.7.

This research investigate two methods to achieve the required QoI:

1. The application of efficient and effective IE approaches, which suit the application and its immediate environment. As shown in Section 2.4, there are a small number of IE approaches that are specifically designed for mWSNs. These approaches have many inherent problems that makes them inefficient outside the application domain they are designed for. On the other hand, researchers have proposed hundreds of static-oriented IE approaches, these approaches have been reviewed and proved to be effective and efficient. In [99], we provide a comprehensive review of static-oriented IE approaches categorisation. One solution to IE from mWSNs would be to apply static-oriented IE approaches to mWSNs by making mobility transparent to the IE layer of the WSN protocol stack. The challenge would be how to abstract mobility from the IE layer while still harnessing the benefits that mobile nodes offer to the WSN system. The reusability of efficient IE approaches has several benefits including reducing development time and costs as well as reducing the number of failures. Reusing approaches that has been previously tried and tested in

working systems is usually more reliable and dependable than designing and implementing new approaches [100]. The initial use of a new approach reveals any design and implementation faults. These faults require more development time and costs to be fixed.

2. The exploiting of nodes mobility to achieve higher QoI at a lower cost. In the presented application scenario, mobility is beneficial in several ways. Firstly, MDCs move closer to event of interest to collect data from significant sensor nodes. This reduces the bridging distance between the data sources and the data collector. Consequently, information timeliness and affordability are improved. Moreover, since mobile sensor nodes are attached to the moving phenomena (i.e, moving vehicles and soldiers), the phenomena is always covered by the sensing coverage. This contributes to higher information accuracy, completeness, availability and validity. Finally, soldiers move in troops close to or inside armoured vehicles. This relative movement increases the density of nodes near the data collectors. Therefore, achieving better energy efficiency and reducing information delivery delay.

Cost of Extracting Information

The cost of information is generally the cost involved in data retrieval and processing in addition to the cost of moving mobile nodes towards the event of interest. Collecting data from the entire network may contain redundant or irrelevant data. Redundant data provides no more information than the currently existed, and irrelevant data provides no useful information in any context. Irrelevant data could be, for instance, data collected from nodes that are away from the event of interest. Therefore, collecting data from the entire network could lead to unnecessary increase in cost of information. Moreover, it can cause degradation in the QoI as the filtering process may not always removes 100% of irrelevant data. Some of the cost of moving MDCs can be recovered through localised retrieval and processing of information. This is possible whenever nodes are within close proximity to the monitored event. Local processing of sensor data results in higher QoI (e.g., lower end-to-end delay), less overall communication overhead, lower energy consumption and better scalability. Given all of its benefits, the required system will be based on local data retrieval and processing to provide its intended functionality.

Mobility, nevertheless, introduces new challenges that has to be resolved in the desired system. Nodes mobility results in frequent topological changes, these has to be carefully managed to reduce any potential interruptions to IE. Frequent topological changes result in intermittent connectivity. This may lead to loss of significant data. For example, when a node moves to a new position, it requires time to establish routing paths; during this period of time this node is not participating in sensing. Furthermore, the topology management traffic consumes a large portion of the valuable bandwidth resources. During major topology update periods, the network availability will be highly affected. Network congestion may delay data delivery, which results in invalidating this data. In conclusion, due to its great impact on the IE process, effective mobility management will be one of the essential design goals of the desired system.

Design Principles

In the design of the desired system, we focus our research efforts on minimising the cost of data retrieval and processing as well as minimising the cost associated with mobility and its management. The main objective of the system is to abstract nodes mobility from the IE layer to enable the deployment of static-oriented IE approaches to mobile or hybrid environments. The desired system will be designed to deal with various mobility models and types using two main design principles:

1. **Transparent and Mobility-aware Adaptation**

   In our attempt to design solutions to support node's mobility in delivering effective IE, we face two conflicting design choices: mobility-transparent or mobility-aware. The coexistence of the two seems to be a right choice to design a system that fits into a wide range of user requirements. The mobility-transparent choice provides transparent mobility support to IE and higher level application functions. In this choice, IE can not perceive any connection changes events; therefore, there must be some mechanisms in the desired system to cope with the changing network context and keep the IE unaware. Mobility-transparent IE do not mean they are fully unaware of nodes mobility. It is important in pervasive IE applications to make applications aware of and adapt to the changing network conditions due to e.g. mobility. On the other hand, a mobility-aware IE approach is aware of the special events happening on network connections. Such events may include e.g. the arrival or departure of a node, varying availability of connectivity, new overlay node due to mobility, network QoS status changes, etc. These applications do not necessitate a transparent management on mobility. Alternatively, they attempt to deal with the changes of location and connection by themselves and exploit the new network status in improving IE and applications performance. One way to achieve this is to keep the mobile's local movements transparent to all entities that are outside of the local scope.

Ideally, a system that makes full use of the mobility benefits is needed. Simultaneously, this system should impose no or minor modifications to, possibly, static-oriented IE approaches used in the upper layers. In the current IE approaches specifically designed for mWSNs, information about the mobility is fed to upper level layers, i.e., they are mobility-awareness. In contrast, our design aims at abstracting mobility from the IE layer. Yet, the deployed IE approach will interact with the underlying network environment and collects data about the actual location of a node, its velocity, direction, etc. This will be achieved by inserting a new layer below the IE layer, which will be responsible for handling transparent and mandatory mobility. For example, a topological change in certain portion of the network is detected by the proposed middle-layer, then, it redirect requests to access data to a replica residing on another part of the network without informing the IE approach about this decision. In other cases, the topological change, or parts of it, is passed up to the IE layer, which is now responsible of taking strategic decisions. It would, for example, be the IE layer to choose which replica to contact in the stable portion of the network. The IE requirements should be carefully taken into account during both the design and the operation of a certain mobility support mechanism.

In the design of our system, the following mobility design principles are considered:

1. It should be remain feasible to decide whether an optional mobility support mechanism should be applied or not, and if it is, to what extent and in what form it will be utilised. This can be decided at design- or run-time, autonomously by the IE layer or explicitly by the users.
2. Since mobility management and support comes at significant cost and complexity, it should remain an option to utilise any potential benefit offered by node mobility. There should be interfaces where IE may become aware of mobility if they need to. As our literature review suggests, many solutions for mobility support are already exist, and it is almost for sure that the mobility support systems in the future are going to be heterogeneous. However, as of today, the interoperation between different protocols is still problematic; and by this work we hope to contribute to solving these problems.
3. The mobility management mechanisms should be effective, i.e., reduces the changes on current network topology. Since mobility means that the location of a mobile may change at any time, how to secure such dynamic location updates is a very important consideration for all mobility support solutions.

In the our presented network model, there are two different types of mobile nodes: Mobile nodes that attaches on soldiers (Layer 1) and mobile data collectors attached on the armoured vehicles (Layer 2). The management of mobility at each layer requires special treatment. In Layer 1, the nature of the application requires relevant soldiers to move in troops or be transported using an armoured vehicle. Hence, the relative position of a node does not change significantly over time. Relative mobility does not impose frequent topological updates as connectivity is mostly maintained as the nodes move. Effectively, this means that the network can be viewed by the IE layer as a temporarily fixed network. Therefore, as far as nodes data and their location are received accurately at the IE layer, nodes mobility at this layer can be abstracted. In contrary, the mobility at Layer 2 of the network has to be passed to the IE layer to realise its benefits. The benefit of this design is that an MDC does not lose the ability to communicate with legacy nodes while roaming around, i.e., the mobile can benefit from unilateral deployment of mobility support. Another potential advantage is that the static nodes do not need to be disturbed by the mobility of the mobiles, which saves resources and reduce interruptions to IE. This design accommodate the inertness of the network by maintaining the current conditions from the IE layer point of view. It is like the 3G network with the smart network and dumb terminals. The gain is that the static nodes can benefit from the mobility support without upgrade. The final benefit is that the IE layer might also be mobility-capable and thus does not provide backward compatibility at all; however, as a trade-off, the system design becomes much simpler, and the data path is always the shortest one.

2. **Minimal Topology Maintenance and Update**

A major design principle is to provide efficient local mobility support. Local mobility management is designed to function together with global mobility management and hence focuses more on performance issues. Because it is usually used on a small scale with a limited number of mobile nodes, fine-tuned mechanisms that are not suitable for large networks can be used to improve performance. As a result of local mobility management, the number of location updates is considerably reduced. Accordingly, local mobility management also contributes to the scalability of global mobility management. To implement local mobility management, nodes have to been assigned to logical groups that define their scope within the global network. Logical grouping of nodes has been proved to improve the QoI and reduce its extraction cost [101]. One problem of local mobility management is that it often requires infrastructure support, i.e. logical groups construction and maintenance. These resource intensive tasks can be assigned to the resource-rich MDCs.

3.4 Summary

This chapter presents our motivational scenario. Battlefields surveillance application was chosen due to its structural complexity and the strict constraints it imposes on the quality of the extracted information. Importantly, battlefield surveillance application provide their functionality over a combination of static and mobile nodes, which allows the analysis of the studied problem under different settings and conditions. The motivation scenario was used to determine the desired system specification and its design principles.

Chapter 4

Self-Stabilisation for Mobility Management

4.1 Introduction

In this chapter, we describe a solution that enables the application of the static-oriented IE approaches, to operate in the presence of multiple mobile nodes. Unlike static networks, mobile WSN topology depends on uncontrollable factors such as mobile nodes velocity and direction. The main challenges is to accommodate the changes in the ad hoc network structure caused by nodes mobility, by maintaining connectivity and performing data retrieval efficiently. Since data retrieval is the main building block and the most energy-expensive process of IE, adaptation to the topological changes is necessary to design any IE approach for WSNs. Self-stabilisation scheme can serve this purpose as it provides adaptivity in a proactive and distributed manner. Topological changes in mobile WSN can be treated as a transient fault and thus self-stabilisation has the ability of adaptation in a proactive manner to topological changes.

Self-stabilising algorithms are constructed in such a way that a given node will execute the same operations in both stable and unstable states. A topological change puts the system into a unstable state, the next steps of normally continued execution will put it back into a correct state. In this work, we focus on the design and analysis of self-stabilising graph distributed algorithms for identifying sets of nodes or sets of edges which satisfy a given property P. The aim of such algorithms is to consistently label links between neighbouring nodes to maintain connectivity and reduce interruption to IE tasks. The desired algorithm adapts to the changes in the network topology due to the mobility of nodes; a state following a topology changes is seen as an inconsistent state from which the system will converge to a state consistent with the new topology.

The basic idea of our self-stabilising algorithm is that every node has a set of local variables whose contents specify the local state of the node, i.e. node membership and connections. Every node only knows about its immediate neighbourhood (a partial view of the global state), and this depends on the connectivity of the system, the link quality (e.g. Expected Transmission Count (ETX)), and other IE requirements (region/group membership). Nevertheless, the purpose of the distributed self-stabilising algorithm is to arrive at a desirable global final state. Often, node mobility brings the system to some illegitimate state, and self-stabilisation brings it back to the legitimate state by executing atomic algorithms in an infinite loop. In this chapter, we extend the distributed self-stabilising algorithms for constructing spanning tree to capture the IE requirements and link quality.

The rest of the chapter is organised as follows: In Section 4.2, we introduce some notation and formulate the problem as a graph. In Section 4.3, we present an overview of self-stabilisation in the context of distributed systems. Section 4.4 discusses the suitability of self-stabilisation for WSNs. In section 4.5, the self-stabilising algorithm is presented with its proof. Section 4.6, concludes the chapter.

4.2 System Model and Problem Formulation

The topology of ad hoc networks can be modelled with an undirected graph $G = (V, E)$, where V is the finite set of nodes and E is the finite set of edges, i.e. links between neighbouring nodes. Two nodes are connected with an edge if each of them is within the radio transmission range of the other, provided no other node in the network transmits at the same time. If i is a node, then $N(i)$, its open neighbourhood, denotes the set of nodes to which i is adjacent, and $N\,[i] = N\,(i) \cup \{i\}$ denotes its closed neighbourhood. Every node $j \in N\,(i)$ is called a neighbour of node i. Throughout this chapter we assume that G of n number of nodes is connected, i.e. the constituents of G have no isolated nodes, and $n > 1$.

The existence of an edge between two nodes depends on the transmission power allocated to the nodes, noise variances at the nodes, as well as modulation and coding schemes used at the nodes. We assume that the links between two adjacent nodes are always bidirectional. Our emphasis on bidirectional edges originates from the fact that most network and transport layer protocols assume bidirectional edges between the nodes. The work developed here can easily be extended to settings where directed edges are allowed between the nodes.

Since nodes can be static or mobile, the network topology changes with time. For modelling simplicity, we assume that no node leaves the system and no new node joins the system. We also assume that link failures are handled by the link layer protocol, e.g. by using time-outs, retransmissions and per-hop acknowledgments. Moreover, we assume that all transmissions from a node are carried out at the same power level. Therefore, the network graph will always have the same node set but different edge sets. These assumptions hold in most mobile ad hoc networks, where the movement of nodes is coordinated to ensure that the network topology remains connected.

4.3 An Overview of Self-Stabilisation

Self-stabilisation is the property of an autonomous distributed systems to achieve correct system behaviour in the presence of arbitrary changes to the state of any constituent node. These algorithms are designed to guarantee convergence to some desired stable state from arbitrary initial states arising out of an arbitrarily large number of faults. A distributed system that is self-stabilising will end up in a correct state regardless what state it is initialized with. A fundamental concept of self-stabilising algorithms is that correct system state will be reached after a finite number of execution steps in a finite amount of time. Therefore, self-stabilisation automatically corrects following arbitrary transient faults/state-changes that corrupt the system state. If correct system behaviour and applications assumptions are such that a process should run on a number of different network topologies, then self-stabilisation implies that the system topology and state can be asynchronously changed, without notifying the process of this change, yet the process will eventually self-correct its behaviour to the new topology. In other words, the global behaviour of the system should ideally remain in the legitimate state desirable by the applications.

Every node has a set of local variables whose contents specify the local state of the node. The state of the entire system, called the global state, is the union of the local states of all the nodes in the system. Each node is allowed to have only a partial view of the global state, and this depends on the connectivity of the system and the propagation delay of different messages.

A node i may change its local state by making a move, which involves changing the value of at least one of its local variables. Self-stabilising algorithms are often given as a set of rules of the form $p\,(i) \Rightarrow M$, where $p\,(i)$ is a predicate (Boolean condition) and M is a move or action. The predicate $p\,(i)$ is defined in terms of the local state of i and the local states of its neighbours $j \in N\,(i)$. A node i becomes privileged if at least one predicate $p\,(i)$ is true. When a node becomes privileged, it may execute the corresponding move. A privileged node that makes a move is called an active node. We say the system has stabilized if no nodes are privileged.

We assume a parallel model in which two nodes in different subgraphs may move simultaneously and asynchronously. A central node, e.g. cluster head, selects among all privileged nodes, the next node to move. If two or more nodes in the same subgraph are privileged, it will be selected non-deterministically which node will move next. One can transform the algorithm to work using complex techniques to recover from instability in an effective manner, i.e. with the minimum overhead and resource consumption. An execution of self-stabilising algorithm is represented by a sequence of moves $M1; M2; \ldots$, in which M_s denotes the $s-th$ move. The system's initial state is denoted by s_0, and for $t > 0$, the state resulting from M_t is denoted by s_t . In distributed systems si expresses a global state, which is a concatenation of the local states of every node and the contents of every communication channel. We refer the reader to [102] for a general treatment of self-stabilising algorithms.

4.4 Suitability for Ad Hoc Networks

A self-stabilising algorithm has several properties that are highly desirable for any distributed ad hoc system. Generally, self-stabilising algorithms are distributed in nature, i.e. they can be used to design efficient light-weight network algorithms where each node has only a local view of the system. In this context, the term local refers to a part of a computer network, specifically, node's immediate neighbourhood. Local algorithms reduce data communication across the network and avoid the computation of the global network state where local adaptivity enables the system to reach a correct state in a finite number of moves. WSN systems can benefit from this advantage. For instance, routing protocols that operate primarily on local information are more efficient, since these can be reactive to local changes, while not requiring energy expensive global transfers of routing information.

Self-stabilising algorithms do not require correct initialisation, because the initial states of nodes does not have to be legal. This property is particularly beneficial to WSNs where nodes start to behave correctly regardless of their initial state. For instance, a newly deployed WSN is unstructured (e.g., nodes do not know their neighbours) and lacks a reliable and efficient communication scheme (e.g. absence of MAC layer support). Yet, at the end of the initialisation phase, communication links will be established and nodes will cooperatively achieve the application functions. The network can reach convergence by having each node communicate only with its immediate neighbours.

Self-stabilising algorithms can recover from arbitrary transient failures rapidly and at low cost. Effective self-healing improves the Quality of Service (QoS) levels by minimising service interruption times. It also increases the network life time by reducing the amount of communication and computation to converge to a stable state. Usually, self-stabilising algorithms do not need to know anything about the failure, e.g. its type, duration, scope, or even whether it actually occurred or not. The only condition is that the failure is not permanent and will eventually stop affecting the system, enabling it to finally reach a correct state in a finite number of steps. Local adaptivity of these algorithms to dynamic changes of the system configuration, e.g. faults or network topology reconfiguration, makes self-stabilisation an interesting solution for a large number of WSN applications, e.g. routing and communication protocols ([103, 104]). This property is highly desirable for any distributed system, since without having a global memory, global stable state is achieved in finite time and thus the system can correct itself automatically from transient failures. Many such algorithms have recently appeared in the literature, a good survey of self-stabilising algorithms can be found in [102].

Finally, self-stabilising algorithms do not rely on strong assumptions on the system, which makes it suitable for a wide range of network deployment and applications. This property simplified the task of designing self-stabilising algorithms and their applications considerably.

4.5 The SS Algorithm

The self-stabilising algorithm proposed here dynamically groups sensor nodes into homogeneous network groups with respect to their topological relationships and their sensing-states. This setting allows users to manipulate groups of spatially distributed data streams instead of individual nodes, which is a more efficient (in terms of communication and computation) IE method. For each data value (e.g., local sensor reading) of node i we seek the nearest (closest value) element in $N(i)$ in terms of absolute difference. Consider the following predicate for any node i:

$$\Phi_i = (|d(i) - d(k)| < |d(i) - d(l)|)\,(\{k;\, l\} \in N(i))$$

This predicate is true when the absolute difference between data value $d(i)$ of node i and the data value $\mathrm{d}(\mathrm{k})$ of its new parent node k is smaller than the absolute difference $\mathrm{d}(\mathrm{i}) - \mathrm{d}(\mathrm{l})$ where $\mathrm{d}(\mathrm{l})$ is the data value of node l, which is the current parent of node. For each sense modality, a node has exactly one parent, except for the root. In the SS algorithm, the parent is chosen based on the link reliability and the soft-state (sensor data) of the node. Using the soft-state will naturally lead to connecting nodes with homogenous data to the same tree branch. This is expected to make the task of data retrieval more efficient in terms of identifying nodes carrying the required data.

The use of the sensor data value to build a graph that assigns nodes to distinct paths, results in a hierarchical description of the data in the form of a tree. Furthermore, it can offer a path over which quires can be sent to sensor nodes. For instance a path can be used by queries to explore sensors data starting at the root and going only as deep as is necessary. The capability to rapidly identify 'data significant' nodes or limit the scope of a query to arbitrary branches of the tree results in substantial energy savings, lower bandwidth utilisation and lower end-to-end delay.

To achieve self-stabilisation, the proposed algorithm uses four variables. The first variable is a binary variable $x\,(i)$ indicating membership in an IE group. The second variable is a predecessor pointer of node i, pointing to one of the nodes in V . By pointing to a neighbour j, written $i \rightarrow j$, a node i communicates to j. The third variable is $L(i)$, where $0 \leq L(i) \leq n$, which is the level of node i in a graph of n number of nodes. The level of a node is the depth of that node with respect to the root node 0. It is calculated as 1 + the number of connections between the node and the root. The fourth variable is the Expected Transmission Count (ETX), proposed by De Couto et al. [104], which is a link reliability-based metric.

In most self-stabilising algorithms, a node can read only the variables of its immediate neighbours, i.e. those nodes that are within distance one from it. In the chapter, we assume a more general model where nodes can read within distance h. Getting access to all node states up to distance h is required for finding high-quality paths in the presence of lossy wireless links. This makes the proposed solution scalable and suitable for multi-hop communication networks.

Consider the following predicate for any node i:

$$\theta_i = ((i \rightarrow j)\,(j \in N\,(i)) \wedge L(i) = (L(j) + 1))$$

This predicate is true when the parent of the node i is one of its immediate neighbours and the level of the node i is greater than the level of its parent by 1. Now we can define our legitimate state.

Definition 1 The system is in a stable state iff for the root node r

$$L(r) = 0 \wedge (r \rightarrow null) \wedge (\forall_i \neq r)\,(\theta_i)$$

We introduce another predicate for any node i

$$\Theta_i = (\exists j\,(j \in N\,(i)) \wedge L(j) < L(i))$$

Θ_i is true for a node i iff there exists at least one neighbour of node i with a level less than that of node i. We can now state the algorithms a set of rules for each node in the graph. The rules on any node are presented in Algorithm 2.

Algorithm 2: The self-stabilizing algorithm on all nodes

R1: if $\mathrm{i} = \mathrm{r} \wedge \mathrm{i} \nrightarrow \mathrm{r} \vee (\mathrm{L(i)} \neq 0)$

then $\mathrm{i} \rightarrow \mathrm{null}, \mathrm{L(i)} = 0$

R2: if $(\mathrm{x}(\mathrm{i}) = 0) \wedge \mathrm{L(i)} \neq 0 \wedge (\forall \mathrm{j} \in \mathrm{N}(\mathrm{i}))(\mathrm{x}(\mathrm{j}) = 1)$

then $\mathrm{x}(\mathrm{i}) = 1, \mathrm{L(i)} = (\mathrm{L(j)} + 1), (\mathrm{i} \rightarrow \mathrm{j})(\mathrm{ETX}(\mathrm{i} \rightarrow \mathrm{j})_{\mathrm{max}})$

R3: if $(\mathrm{x}(\mathrm{i}) = 1) \wedge (\mathrm{i} \rightarrow \mathrm{j})(\mathrm{j} \in \mathrm{N}(\mathrm{i})) \wedge (\exists \mathrm{l} \in \mathrm{N}(\mathrm{i}))(\mathrm{x}(\mathrm{l}) = 0)(\mathrm{ETX}(\mathrm{i} \rightarrow \mathrm{j}) < ETX(i \rightarrow l) \wedge \Phi)$

then $\mathrm{x}(\mathrm{i}) = 0, \mathrm{L(i)} = (\mathrm{L(l)} + 1), (\mathrm{i} \nrightarrow \mathrm{j})(\mathrm{i} \rightarrow \mathrm{l})(\mathrm{ETX}(\mathrm{i} \rightarrow \mathrm{l}) > ETX(\mathrm{i} \rightarrow \mathrm{k})_{\mathrm{max}})$

P1: if $(\mathrm{x}(\mathrm{i}) = 0) \wedge (\mathrm{i} \rightarrow \mathrm{j})(\exists \mathrm{k} \in \mathrm{N}(\mathrm{i}))(\mathrm{x}(\mathrm{k}) = 0) \wedge (\mathrm{ETX}(\mathrm{i} \rightarrow \mathrm{j}) > ETX(\mathrm{i} \rightarrow \mathrm{k})\, \forall \mathrm{k} \in \mathrm{N}(\mathrm{i}))$

then $\mathrm{i} \nrightarrow \mathrm{j}, \mathrm{i} \rightarrow \mathrm{k}$

In Algorithm 1, rules **R1, R2** and **R3** are membership moves that allows nodes to change IE group membership. By executing rule **R1** the node becomes a root node. The root node may be privileged in an illegitimate state, but once it makes a move it becomes un privileged and can never be privileged again. If a node travels into a location where it is surrounded by nodes that all belong to one group, then it joins the group by executing rule **R2**. If there exists more than one node j, i will connect to the node with the best link quality $ETX\ (i \rightarrow j)_{max}$. If two or more links have equal ETX, then any one is chosen at random; the choice does not affect the correctness of the algorithm. A mobile node is already a member of an IE group, but it has a neighbour l that is a member of a different IE group, may use rule **R3** to leave its current group and join the other group provided that l offers a better ETX and the new group is more related to its data state. The rule **P1** is a pointer move that do not modify the membership in the dominating set. It is used to adjust the pointer from one node to another node that offers a better ETX. Any node $i\ (i \neq r)$ is privileged

iff $\neg\theta_i \wedge (\Theta_i \vee (\neg\Theta_i \wedge L(i) < n) \vee \neg (ETX\ (i \rightarrow j)_{max}) \vee \neg\Phi)$.

If a node $i \neq r$ is not privileged then the predicate

$\theta_i \vee (\neg\Theta_i \wedge L(i) = n) \vee \Phi$ is true.

Definition 2 If a privileged node changes both its level and pointer, it becomes unprivileged and remains so until the new predecessor node makes some move.

4.5.1 Proof of Correctness

Lemma 1 In a stable state, no node is privileged.

Proof: Obvious from definitions 1 and 2.

Remark 1 If a node $i(i \neq r)$ is not privileged then the predicate $\theta_i \vee (\neg\Theta_i \wedge L(i) = n)$ is true.

Lemma 2 In a stable state, each node in the graph has a level less than n.

Proof: We will prove by contradiction. Assume that $S(l) = \{i|L(i) = l\}$ is the set of nodes with level $l \in [0; n]$ and $|S(n)| \geq 1$. Since the root node r is not privileged, $L(r) = 0$ and hence $r \notin S(n)$. In a connected graph, at least one node $i \in S(n)$ must have a neighbour outside of $S(n)$. Since this neighbour has a level less than n, θ_i is true ($\neg\Theta_i$ is false) and hence there exists a node j such that $i \rightarrow j$ and $L(i) = L(j) + 1$. That is, $L(j) = n - 1$ and hence,$|S(n-1)| \geq 1$. For any node $l \in S(n-1)$,θ_i l must be true since node l has a level less than n and is not privileged. Proceeding as before, $|S(n-2)| \geq 1$. Using a similar argument, we observe that $|S(l)| \geq 1$ for all $l \in [0; n]$. Since these $S(l)$'s are mutually disjoint, union of all these sets has at least $(n+1)$ nodes, contradicting our assumption that $l \in [0; n]$. Hence, $S(n)$ must be null, i.e., each node in the graph has a level less than n.

Lemma 3 In a stable state, the global system state represents a valid spanning tree of the graph rooted at node r.

Proof: By Lemma 2, every node i $(i \neq r)$ has a level less than n and hence by Remark 1 the predicate θ_i is true for node i. The parents pointers define a subgraph of G whose edges are $(i \rightarrow j)$ $(i \neq r)$. This subgraph is acyclic since θ_i is true for each node i and is connected since every node is connected to r. Hence, the global system state represents a valid spanning tree of the graph rooted at node r.

Remark 2 The local state of a node is defined by the tuple $(L(i), i \rightarrow j, x\,(i))$. Since the levels and parents can be finitely many different values, the system state space is finite.

Definition 3 For any system state, we define the set $T = \{r, i\}((\forall i|\theta_i = T) \wedge (j \in T))$ where j is the parent of i.

Definition 4 When r is privileged, T is said to be in the initial state or T_I . When r is not privileged, T is said to be in the final state or T_F .

Lemma 4 For any node $i \neq r\ in\ T_I$, the system state will remain in the T_I for all following system states until r makes a move.

Proof: In T_I only r is privileged. Other nodes will only be privileged when their parents makes a move. Hence, all nodes in T_I will remain in the same state until the r is privileged. When r in an initial state makes a move, the T for the resulting system state may consist of only r if the level of r is changed by the move.

Lemma 5 Any node i in the T_F for a system state will always remain in the T_F for all following system states.

Proof: Similar to the proof of the previous lemma.

Definition 5 Let $\Gamma \subset \{V - r\}$ where $\forall i \in \Gamma\ (x\,(i) = 1)$. We call a move by a node $i \in \Gamma\ a\ \Gamma$ –local action iff either the action does not change the node's parent pointer or the pointer points to another point in Γ. Let $l_{min} = min\,\{L(i) \in \Gamma\}, \Gamma_{min} = \{i|L(i) = l\_min\}$ and $\Gamma' = \Gamma - \Gamma_{min}$.

Lemma 6 For any Γ, there can not exist an infinite sequence of Γ-local actions.

Proof: We prove the above lemma using induction. If $\Gamma = 1$, then, the node in **Γ** can only make n times move using rule **P1**. Hence, the claim is true for $|\Gamma| = 1$. Let the claim be true for $|\Gamma| = k$ where $k \in (1, n)$. Assume that for $|\Gamma| = k$ there exists a finite sequence of Γ-local actions. A move by a node in Γ', if not Γ'-local, has to be done using rule **P1**, i.e. the new parent will be in Γ_{min}; the move is Γ-local by assumption, hence, the node remains unprivileged until some node in Γ_{min} makes a move. Since $|\Gamma'| < |\Gamma|$, by induction, there is no infinite sequence of Γ'- local moves. Thus, a node in Γ_{min} may make a move in finite time to become Γ'-local. If all nodes in Γ_{min} make concurrent move by rule **P1**, l_{min}will increase by 1 resulting a new Γ_{min} or $|\Gamma_{min}|$ will decrease. Proceeding as before, we observe that l_{min} will increase by 1 in finite time until it becomes n and $|\Gamma'| = 0$ and no Γ moves will be possible.

Lemma 7 In any illegitimate state with a T_F , the set T_F grows in size after finitely many moves.

Proof: Assume that $\Gamma = V - T_F$. Any node $i \in \Gamma$ may make a move that is not Γ-local, then i enters T_F . Since an infinite sequence of Γ-local moves is not possible, T_F grows in size after finitely many moves.

Theorem: In any network having no isolated nodes, Algorithm 1 brings the network back to a stable state starting from any arbitrary illegitimate state in finite time.

4.6 Using Weighted Criteria to Plan Moves

One significant WSN design challenge is meeting application requirements, such as end-to-end delay and network lifetime, for many application domains. In Algorithm 2, the link quality (in terms of ETX) and the node's soft-state (in terms of Φ) are weighted equally. Often, the system owner has to prioritise these two self-stabilisation criteria to meet the requirements of application performance. This type of WSN optimisation is referred to as static optimisation [105]. In static optimisation, the WSN is optimized permanently at the deployment time. This type of optimisation is suitable for stable/predictable applications. However, static optimization is inappropriate when the operating environment and the user requirements change continuously. The optimisation of the WSN during runtime to adapt to application requirements is referred to as dynamic optimisation [105].

We position the presented self-stabilisation algorithm in the dynamic optimisation category. This is due to its ability to automatically detect path degradation and adapt by choosing resources appropriately to provide reliable communication. In addition, Algorithm 2 addresses data retrieval problems by allowing the node to intelligently choose a group that meets emerging IE demands.

To offer user greater flexibility in setting up the network to meet emerging application performance demands, we assign importance (weights) to the two criteria (ETX and Φ) used in Algorithm 2. We define a new value D, which is the weighted average for each possible parent, to allow nodes to plan potential moves. The first step in planning a new move is to use the same scale to measure both criteria. Proportional scoring is one technique to measure all criteria on the same scales. Next, the system owner have to decide the relative importance of the two criteria. However, it is crucial to consider the ranges of the criteria; the weights should reflect the relative value of going from best to worst on each scale. For example, if ETX is three times more important than Φ, then this implies that ETX weight is 0.75 and Φ weight is 0.25. Finally, the weighted averages to get the overall score of each parent is calculate as:

$$D = 0.75 \times ETX + 0.25 \times \Phi$$

Optimising WSN to meet both performance and IE requirements depends on a number of other factors, including the system topology, hardware architecture, routing protocols, IE approaches, the jitter requirements of the end application, etc. D can be extended to include any number of independent variables:

$$D = a + b_1X + b_2X_2 + \cdots + b_kX_k + e$$

D is an observed score on the dependent variables, a is the intercept, b is the slope, X is the observed score on the independent variable, and e is an error or residual.

4.7 Summary

This chapter presents the first steps towards enabling the deployment of IE approaches designed for static networks to run on networks with mobile nodes. The proposed solution makes node mobility transparent to upper layer protocols by rapidly stabilising the underlying network topology. Self-stabilisation is achieved in a distributed manner based on the node's local variables.

Chapter 5

Dynamic Network partitioning

5.1 Introduction

The delimitation of the problems to be addressed before the application of static-oriented IE approaches to mobile networks is possible, can be formulated in view of the scenario presented in Chapter 3, Section3.2. Chapter 4 presented a solution from the notion of self-stabilisation to manage nodes mobility in the bottom layer of our network model. Using self-stabilisation, some approximation of fixed topology is established to recover from an arbitrarily corrupt communication state. Thus, the presented self-stabilisation algorithm provides autonomous adaptability against topology changes in a transparent manner to both upper network layers and to user applications. The benefits of mobility at the bottom layer, e.g., enhanced coverage, are achieved before data is conveyed to upper layers. Therefore, mobility can remain abstract and transparent to higher level network entities.

This chapter aims to present a solution for the management of MDCs mobility. The mobility management strategy at this layer of the network has special requirements and design goals. MDC mobility management is designed to realise the benefits introduced to the network and applications by this type of nodes. It aims to balance reliable collection of large amounts of data, while reducing the amount of signalling and management overhead. MDCs provide mobility aware data collection as available contact with neighbouring nodes is scarce and short depending on its movement speed. Therefore, it is essential to setup communication paths with the MDC promptly to maximise the number of messages correctly transferred to the sink. Moreover, since MDCs move during data transfer, message exchange must be mobility-aware. In fact, the knowledge on the MDC mobility pattern can be used to further optimise the detection of mobile nodes. If it is possible to predict with certain accuracy the MDC visiting times, sensor nodes can be pre-configured to when they expect the MDC to be in their transmission range. From an IE perspective, MDCs are expected to follow the phenomenon of interest, and hence, the awareness of their locations and movement is desirable.

In this chapter, we propose an ensemble solution to cope with nodes mobility in the middle layer. The main focus here is to manage the mobility of the MDCs to achieve continues sensing coverage and reduce the cost of topology updates. Higher IE efficiency can be achieved by reducing connectivity disconnections times. Less and shorter disconnections results in reduced signalling overhead and packets loss; therefore, increasing the sensing coverage at a lower cost. From users point of view, information quality in military applications is very critical. Such systems need to be fast, accurate and provide continuous coverage. Therefore, our goal here is to organise MDCs in such a way that interruptions to IE is reduced, while keeping energy consumption and resource usage to the minimum.

Utilising MDC nodes to collect data has many benefits. It helps decreasing the bridging distances between data sources and data sinks. It also avoids problems associated with bottlenecks, especially in areas around the sink, such as packet loss, increased end-to-end delay and energy depletion. The existence of multiple data collectors eludes the breakdown of interconnections; meaning that if one data collector fails, data can be transmitted through another data collector. This guarantees high information availability, completeness, and validity. Furthermore, by nature of the military application, MDCs move towards the location of an event or area of interest close to sensor nodes holding user-needed information. This places the power-rich MDCs closer to the information-rich (called *significant*) nodes; consequently, it minimises end-to-end communication delays. Logically, data held by nodes that are further away from an event is less significant to the user. Therefore, tracking an event insures that only relevant data is used in extracting information requested by the user. This leads to increased information accuracy due to the fuller coverage and exclusion of irrelevant data from a query response.

There is an obvious trade-off between the cost of mobility and the benefits it offers to IE. In here, the aim is to reduce mobility management cost and optimise the performance of MDCs to increase information quality. The performance of distributed algorithms in WSNs is strongly influenced by the connectivity of the network [106]. Therefore, the network has to be interconnected to efficiently handle MDCs mobility. For scaling, robustness, and load-balancing reasons, it is desirable to divide the network into multiple logical groups. In this way, it becomes plausible to guide the dissemination of user queries and constrain control messages to avoid the ripple effect that may be caused by MDCs movement.

The remainder of this chapter present the technical details of our contributions towards MDC mobility management at the middle network layer. The first part of this chapter provides an illustration of the features of our proposed solutions. Then, we describe our algorithms of establishing logical entities, called collection zones, and the MDC selection. The following section brings important definitions and assumptions, we provide a detailed description of collection zones reconciliation and service zones construction. Followed by descriptions of updating and expanding service zones and avoiding orphaned nodes problem.

5.2 An Overview of MDC Mobility Management

We begin this section by introducing the terminology used throughout the chapter, then we briefly introduce the basic elements of our logical network partitions. Then, we provide an overview of the MDC mobility management using the defined logical partitions. In this section, we only aim to provide the high level picture of our proposed mobility management solution; the details of how to form various logical entities and how they are used is provided in the next section.

MDCs are resource-rich mobile nodes that lie at the middle layer of the network. They collect data sensed by sensor nodes either directly (e.g., by visiting sensors and collecting data from each of them) or indirectly (e.g., through intermediate nodes). They are not sources nor final destinations of messages, but exploit mobility and powerful resources to support network operation or data collection. MDCs can process the collected data autonomously or send it to the sink for central processing. For its interaction with the sink, an MDC may use some forward path (either direct or multi-hop) to the sink. In most cases, the interaction uses direct connectivity with the sink, as MDCs has long radio range compared to conventional sensor nodes. Based on estimates from real-life applications, MDCs are assumed to form 5% of the total network population. They can move at variable speeds in any direction within the monitored area.

The data retrieval process requires stable network topology to select nodes carrying desired data, e.g, [107, 108]. Moreover, spatial events are commonly collaboratively detected by neighbouring sensor nodes. For instance, in target tracking applications, multiple nodes share their location data to determine the position of a nearby target. However, in the presence of MDCs mobility, topology updates are needed frequently and topology maintenance incurs high energy consumption. This presents a challenge that has to be resolved before the deployment of a successful IE system is possible.

Aiming at reducing mobility-caused signalling overhead and increasing the quality of returned information, we design and implement a logical network structure based on convex hulls. We propose a dynamic partitioning algorithm that results in a set of well delimited network partitions called Service Zones (SZs). An MDC and neighbouring sensor nodes collaborate to establish logical SZs. Dynamic partitioning of the network into SZs have many benefits. Firstly, partitioning reduces signalling overhead, consequently bandwidth utilisation, by localising mobility management traffic. It is well established that less congestion reduces queuing delays, which in turn improves the overall information timeliness. Importantly, reducing control messages in the network conserves energy, which leads to better information affordability and longer network life. Secondly, network partitioning is a well-tested solution to achieve scalability and load balancing. Grouping nodes into smaller logical sets makes buffer overflows and energy depletion less of a problem. This is expected to support achieving high information completeness and availability as well as maximising the network lifetime. Finally, maintaining continuous connectivity with the MDC when it is in the communication range of sensor nodes increases information availability and validity.

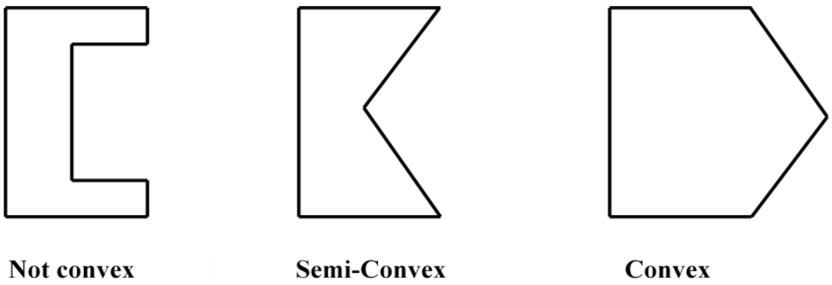


Figure 5.1: Degrees of convexity in point sets

In here, we introduce the terminology used and briefly introduce the basics of convex hulls. Throughout this thesis, the terms *convex hull* and *SZ* are interchangeable. In mathematics, the convex hull of a set Q of points in the Euclidean plane is defined as the intersection of all convex sets containing Q or as the set of all convex combinations of points in Q. A set is said to be convex if for every pair of points within the set, every point on the line segment that joins the pair of points is also within the object. Figure 5.1 gives some examples of sets that are not convex, semi-convex, and convex, respectively.

Let $\mathrm{Q} = \{\mathrm{q}_0, \mathrm{q}_1, \ldots., \mathrm{q}_\mathrm{n}\}$ be a set of sensor nodes, where $|\mathrm{Q}| \geq 3$ is the minimum number of nodes to create a valid convex hull. Each node q is represented as a pair $(\mathrm{q}(\mathrm{x}), \mathrm{q}(\mathrm{y}))$. Let the function $\mathrm{CH}(\mathrm{Q}) = \mathrm{H}$ returns a set of nodes $\mathrm{H} \subseteq \mathrm{Q}$ that encloses the nodes in set Q (i.e., border nodes). H is the smallest convex region that contains Q and is called the convex hull of the nodes set Q. If $|\mathrm{H}|$ is the number of nodes in H, then $|\mathrm{H}| \leq |\mathrm{Q}|$. The set H stores the list of vertices of the convex hull in counter clockwise order.

In dynamic network partitioning, the frequent boundary updates presents a significant challenge. Collection Zones (CZs) are introduced to tackle this challenge. CZs are designed such that their maintenance is quick and efficient. CZ maintenance cost is reduced by performing efficient neighbour discovery and localised computation. The CZ of an MDC is defined by the set of nodes directly connected to that MDC. The movement of an MDC within its defined CZ does not require connectivity or neighbourhood update. This enables nodes within a CZ view their MDC as a virtually static node for a certain period of time. Depending on its speed, sensor nodes can easily predict the connectivity period with their present MDC.

The result of the network partitioning process is based on the number of MDCs, their positions, and their movements. For each MDC, we define a convex group of nodes that will form the MDC's Service Zone(SZ). The construction of convex groups does not exhibit high computational complexity [109]. While an MDC is moving inside its SZ, it performs several operations to keep the network topology up to date. These operations include, updating its SZ members list, connecting new nodes or disconnecting existing nodes from its SZ. These operations ensure that sensor nodes can be allocated to the best MDC that can forward its data to the sink more effectively. Simultaneously, sensor nodes should be fairly allocated to MDCs to load balance their workload.

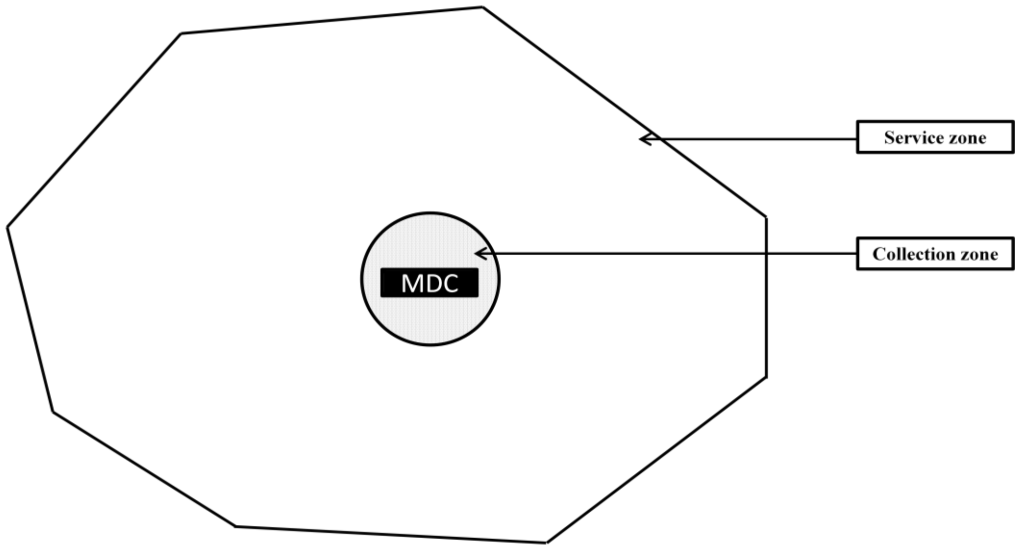


Figure 5.2: The logical structures defined around an MDC

Figure 5.2 shows the conceptual relationship between the SZ and the CZ. A SZ is a logical designated geographical zone around an MDC, containing a set of nearby sensor nodes. It is forming a convex group of nodes constructed by the MDC. The MDC is responsible for all communication in that SZ. MDCs exchange control messages with each other and only with sensor nodes that belong to their SZ. This way, every sensor node receives control messages only from the MDC that it belongs to. As a result, flooding problems from MDCs to sensor nodes in the bottom layer is evaded.

A CZ of an MDC, is the circular area around an MDC bounded by a circle. The circle's radius is equal to half of the radio range of sensor nodes. In Cartesian coordinates, the centre of the CZ is the physical position of the corresponding MDC before it moves. An MDC communicates directly with nodes that are inside its CZ. An MDC can move inside its CZ and stay directly connected to the same set of nodes. This design exploits the fact that the active radio coverage of MDCs is wider than their CZs. Therefore, sensor nodes can consider the MDC static until it moves out of its CZ. Hence, it does not need to issue any neighbour discovery or update message during this period of time. SZs are assumed to be larger than the CZs. The SZ and the CZ change dynamically depending on the mobile MDCs speed and direction information. As mentioned previously, the position, speed, and direction of the mobile nodes are obtained by the GPS information (i.e., latitude, longitude, altitude, speed, track).

5.2.1 Network Setup

This section goes through the detailed process of setting up the network to prepare it for operation. This process is the stage after nodes deployment. It consists of three phases: neighbourhood discovery phase, CZ creation phase, and network formation phase, respectively.

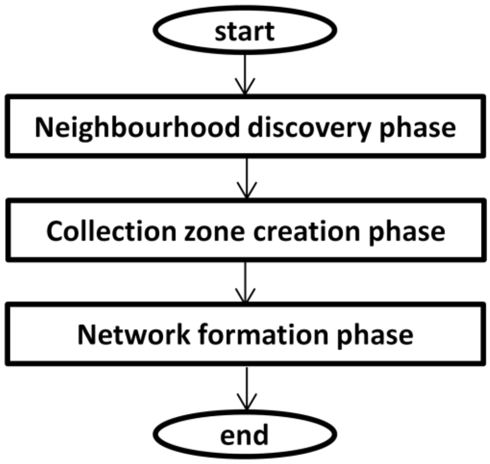


Figure 5.3: Network setup procedure

Figure 5.3 illustrates the various phases of the network setup process. During the neighbourhood discovery phase, the MDCs create binding tables for storing nodes' information. MDCs proactively discover their neighbours, and then, sensor nodes receive advertisements from MDCs, then, each sensor node chooses the optimal MDC and connect to it. During the CZ creation phase, each MDC constructs its CZ boundaries and determine which sensor nodes are located inside it. These sensor nodes will have direct communication with the MDC. During the network formation phase, reconciliation of the overlapped CZs occur and SZs are constructed, and thereby the network is ready for operation.

5.2.2 Network Discovery Phase

In the literature, neighbourhood discovery can be classified into the following main categories: proactive, e.g. [110], reactive, e.g. [111], and hybrid, e.g., [112]. In the proactive approach, the data collector periodically broadcasts advertisement message. When the message is received by a sensor nodes, that node creates a route to the data collectors and relays the message to its neighbours. This results in many duplicated messages consuming valuable bandwidth and energy. In contrast, in the reactive approach, discovery messages to initialise or update connections are initiated by sensor nodes. The sensor node broadcasts a connection request message in the network. When a data collector receives the message, it unicasts a reply message containing its details, such as, its address, location, and available resources or services. This approach saves bandwidth and energy as it sends requests only when information is needed. However, the main drawbacks of this approach is the high latency in data collector discovery and causing bottlenecks on nodes close to the data collectors. The hybrid approach uses a combination of the two above approaches by considering the disadvantages of two of them.

Node ID	X position	Y position	Track	Battery level	Last update time
1	$54°11'18.0"N$	$07°52'18.4"E$	$37\ degrees$	$92\ Joule$	$2014.01.18\ T\ 16{:}02{:}53$

Figure 5.4: Binding table entry example

Since convex hulls partitioning divides the network into many partitions, we use hybrid discovery approach. The proactive approach is used before constructing the SZs, and the reactive approach is used by orphaned nodes (described in Subsection 5.3.1) after constructing the SZs. After deploying the network, every MDC in the network creates a binding table, as in Figure 5.4. Binding tables contain the following information: nodes IDs, locations obtained from GPS, battery level, depth level, and the update time. The binding table is used to store information about sensor nodes that are connected to the MDC.

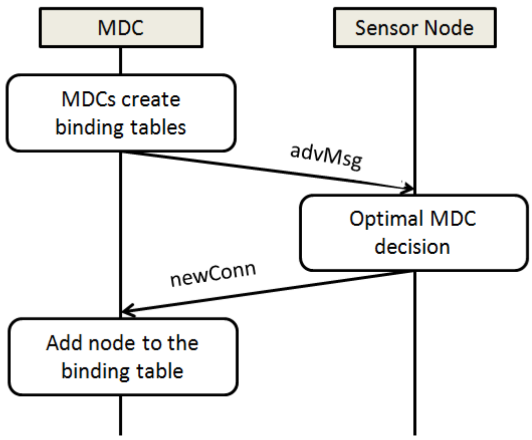


Figure 5.5: Network discovery messages diagram

As it is illustrated in Figure 5.5, every MDC proactively discovers its neighbours. It broadcasts advertisement messages containing its location information. Intuitively, the traditional way of broadcast in WSNs is flooding. When a sensor node receives a packet, it sends it to all of its neighbours, which results in serious redundancy, collisions, and contention. To reduce the impact of such consequences, MDCs broadcasts advertisement messages to all nodes that are less than d away from it (where d is the maximum distance that an advertisement message can be transmitted over). Upon receiving the advertisement messages by sensor nodes, each sensor node makes a decision about the best MDC to connect to. The decision is based on our proposed MDC selection algorithm (described in Subsection 5.2.2.1). After making the decision, each sensor node replies with a $newConn$ message to the chosen MDC. Finally, MDCs receive replies from different nodes and adds them to its binding table.

5.2.2.1 Optimal MDC Selection Algorithm

During the network discovery phase, sensor nodes may receive advertisement messages from multiple nearby MDCs. In this subsection, we consider the selection process of the best MDC among several alternatives. The choice of an MDC has direct impact on the frequency of network reconfiguration, which significantly impact the energy conservation, load balancing, and QoI. Joining an appropriate MDC is a very important but sophisticated task. Various factors need to be considered for selecting an MDC. Some of these factors include distance between the MDC and the selecting node, transmission latency, mobility speed and movement direction of the MDC.

Our optimal MDC selection scheme aims to improve the stability of the network and reduce topology reconfiguration frequency. For this, the optimal MDC selection scheme is based on the principle that the sensor node will be able to predict the future disconnection time. If the mobility parameters are known (speed, direction) as well as the positions of the sensor node and the MDC, sensor node will be able to determine a parameter called the Connection Expiration Time (CET). CET is the duration of time that a sensor node would remain connected to an MDC before it loses connectivity due to the lack of a communication route (disconnection). Thus, lower CET means stable network for longer time.

Let n be a sensor node that received an advertisement message from an MDC. The MDC moves in θ direction in two-dimensional space (with respect to the positive X-axis). Let their initial position be $\mathrm{x_n, y_n}$ and $\mathrm{x_{MDC}, y_{MDC}}$. Suppose that the MDC travels at the speed of $\mathrm{v\ m/s}$. The velocity of the MDC on the x and y axis can be can be calculated using the following formula:

$$v_x = v * \sin\theta$$

$$v_y = v * cos\,\theta$$

To calculate CET

$$CET = \frac{-\left(v_y * |x_{MDC} - x_n|\right) + \sqrt{\left(v_x^2 + v_y^2\right)R^2 - \left(v_y * |y_{MDC} - y_n| - v_x * |x_{MDC} - x_n|\right)^2}}{\left(v_x^2 + v_y^2\right)}$$

Following is the algorithm of optimal MDC selection where $maxConnection$ is the connectivity duration of the optimal MDC, $membersNo$ is the number of connected nodes to an MDC.

Algorithm 3: Optimal MDC selection algorithm

```
Input: MDCs details, sensor node location
Begin
MDC_ID=MDC1; maxConnection= 0 ; MembersNo= 0;
for every MDC MDCi do
   if (CET_MDC i > maxConnection) then
     maxConnection = CET_MDC i;
     MDC_ID= MDCi ;
else if (CET_MDC i= maxConnection) then
   if (MembersNo _MDC i < MDC_ID.MembersNo) then
     MDC_ID= MDCi  ;
return MDC_ID;
End
Output: ID of the selected optimal MDC
```

The MDC which has the maximum CET then it is elected as a serving MDC. Once the MDCs selection is over, the node transmits a newConn message to the selected MDC. Then the MDC adds that node to its binding table. There are two cases for applying the algorithm. In case one, during the network setup phase, through sensor nodes that receive advertisement messages from MDCs. In this case, nodes set R value to d, the maximum distance that the MDC forward advertisement message over. In case two, orphaned nodes (described in Subsection 5.3.1) apply the algorithm assuming that R is equal to the distance between the MDC's CZ and the intersection point with the convex hull boundary.

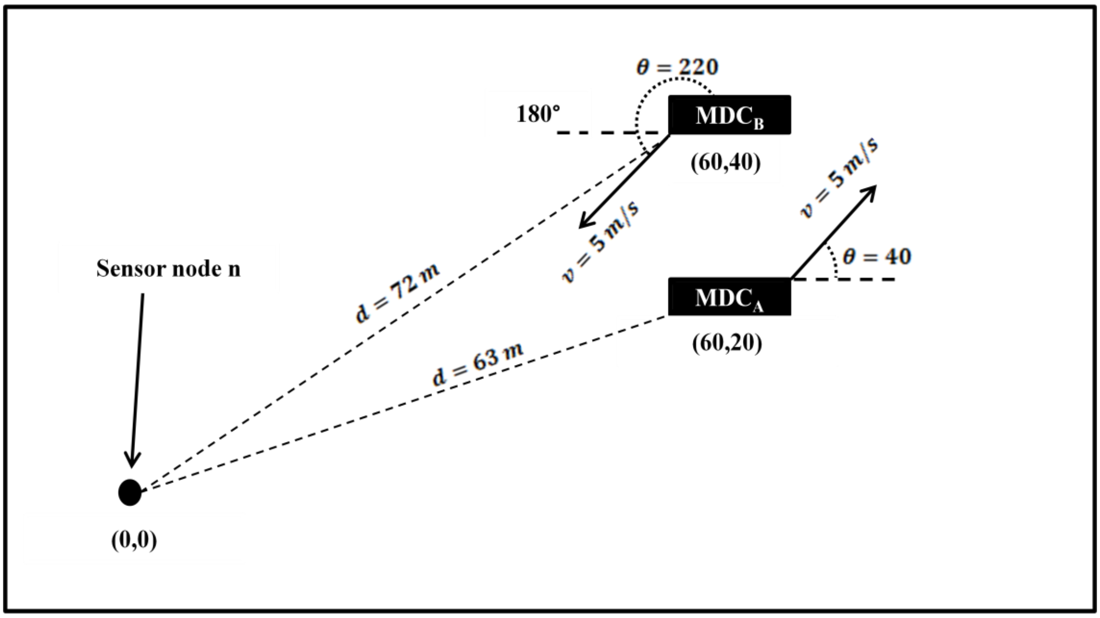


Figure 5.6: Connection Expiration Time example

Figure 5.6 illustrates the MDC selection problem using an example. A sensor node, n, located at coordinates $(0, 0)$ has received two advertisement from those two MDCs. MDC_A is $63\,m$ far from n, and moving away with a direction angle of $40°$. MDC_B is $72\,m$ far from n, and moving towards the node with a direction angle of $220°$. Both MDCs are moving with the same speed of $5\,m/s$. Although the distance between n and MDC_A is shorter than that to MDC_B, the CET of the MDC_B is longer. This is because MDC_B is moving towards the sensor node, and hence, it would stay connected to the node for $29\,seconds$. In contrast, MDC_A is moving away from n and will it will lose connectivity with n after $9\,seconds$. Accordingly, the optimal MDC for a sensor node is the closest MDC that would remain connected for longer time.

5.2.3 Collection Zones Creation Phase

Maintaining an up-to-date accurate neighbourhood knowledge is a challenging task. One solution to minimise neighbourhood updates is to predict when a node is expected to leave the neighbourhood. The basic and simple way for neighbourhood maintenance is by using periodic discovery messages. However, the most significant drawback for this method is choosing the rate at which the "$hello$" messages are sent. A high beacon rate results in increased bandwidth usage and communication cost. In the context of IE, this reduces both information affordability and system availability. As the system availability decreases, information timeliness will significantly retrograde. In contrast, a low beacon rate may possibly miss important topology changes or events where critical reconfigurations take place. This affects information availability, validity, and completeness.

In dynamic partitioning of the network to convex hulls, updating convex envelope using "$hello$" messages consumes high bandwidth and energy. To determine when a convex hull update is necessary, we propose and define a CZ. CZs allow the node to determine when to issue a neighbour discovery message and reconfigure its local connections.

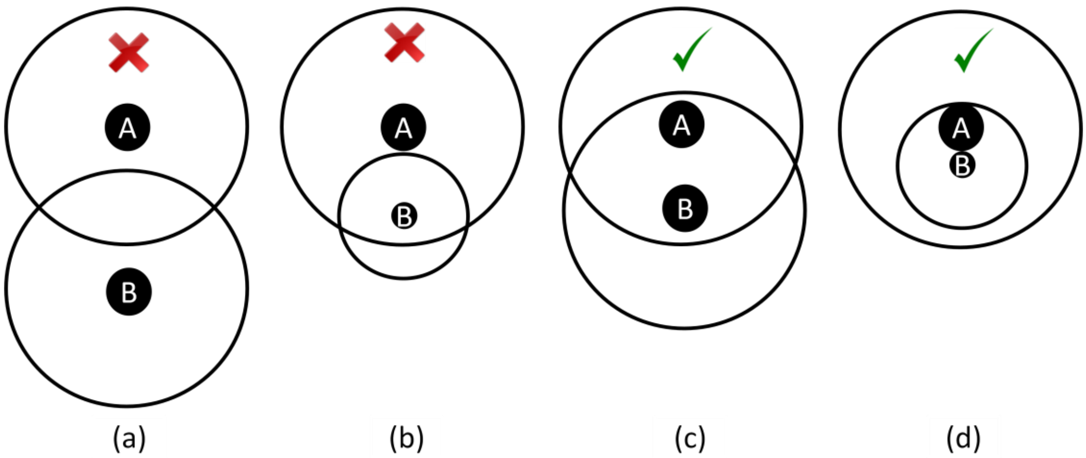


Figure 5.7: Symmetric communication between nodes

Before moving on to describe the creation of a CZ, it is critical to mention that a node can connect to any other node as long as both are within each other's range. Figure 5.7 shows some examples of the connectivity status of two nodes with same and different radio ranges. The overlapping of radio ranges in Figure 5.7-a is not sufficient for a connection because nodes are completely outside of each other's range. In Figure 5.7-b, node c is within node's a radio range, but not vice versa, and therefore, no connection can be made. In Figure 5.7-c, both nodes are barely within each other's range, but it is stuffiest to make a connection. Figure 5.7-d is similar to our case, since sensor nodes have less radio range than MDCs, the MDC needs to be within sensor nodes radio range. Therefore, the MDC active communication range in the view of sensor nodes is narrower than the actual radio transmission range. However, the MDC active communication range in the view of other MDCs is equal to their wide transmission range.

Let $N = \{n_1, n_2, \dots, n_n\}$ be the set of sensor nodes within the MDC active communication range. Then, the CZ is defined as

$$\{CZ \subset N \mid d\,(n_i, MDC) < R/2\}$$

where d is the distance between the sensor node and the MDC and R is the active communication range of the MDC.

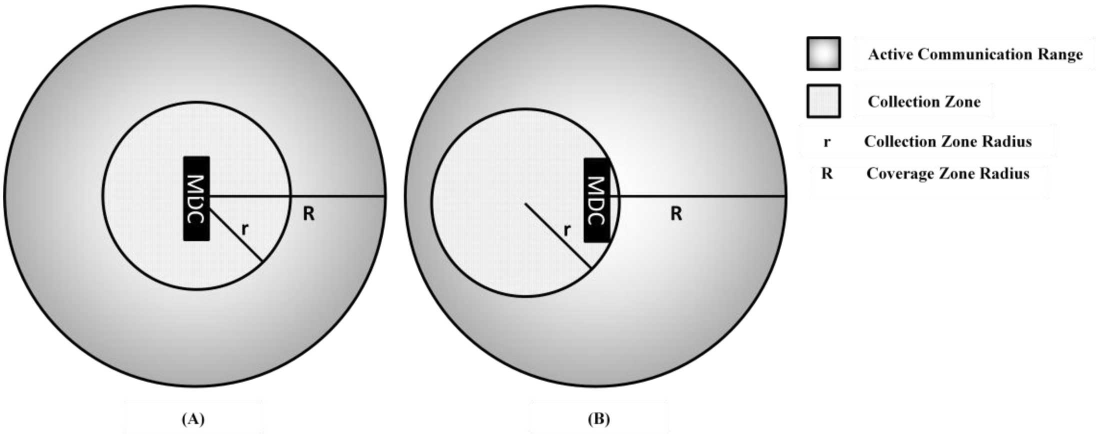


Figure 5.8: MDC coverage zone and collection zone

An MDC can move inside its CZ and stay directly connected to the same set of nodes. As long as the MDC is inside its CZ, it does not need to issue any neighbour update messages. In Figure 5.8, the MDC defines the CZ is a smaller inner circle that a radius r in its active communication range. Initially, when an MDC creates its CZ, it will be located in the centre of the created CZ, Figure 5.8-A. The MDC checks its binding table and determines which sensor nodes belong to its CZ (i.e., $d\,(n_i, MDC)$) using the following equation.

$$d = \{(\mathrm{X_n}, \mathrm{Y_n}) \in\ CZ : (\mathrm{X_n} - \mathrm{X_{MDC}})^2 + (\mathrm{Y_n} - \mathrm{Y_{MDC}})^2 < (\frac{R}{2})^2\}$$

In Figure 5.8-B, although the MDC has moved, no updates is required as long as the MDC is inside its CZ perimeter. When an MDC leaves its CZ, the collection zone will be updated and discovery messages will be exchanged to reconfigure the network changes.

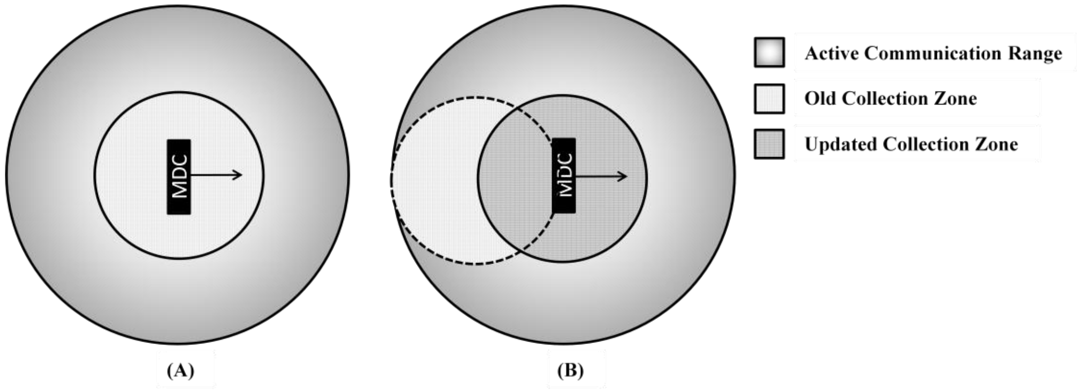


Figure 5.9: MDC defining a new collection zone

Figure 5.9 illustrates the situation where the MDC moves out of its current CZ. In Figure 5.9-A, the MDC started moving from the centre of its CZ towards the perimeter. The periodic membership updates continues as normal. If the MDC continues moving crossing the perimeter of the CZ, Figure 5.9-B, the MDC will update the collection zone and disconnect nodes that do not belong to the new collection zone. The update process includes adding some nodes located inside its active communication range to the updated CZ, and removing nodes that belong to the original CZ. Some of the nodes which are already in the original CZ remain inside the updated one, i.e., the intersection area between the two collection zones in (Figure 5.9-B). The MDC does not exchange configuration messages with these nodes.

5.2.4 Service Zones Formation Phase

The service zones (SZs) formation phase commences when MDCs have their CZs created. MDCs use local information stored in their binding tables to construct their SZs. The vertices of the SZs (convex hulls) will be the farthest connected nodes from the MDC. However, to maintain load balancing among various SZs, the SZs formation phase is comprised of two steps: CZs reconciliation step, and SZs construction step. The former step is only performed by MDCs that have overlapping in their CZs. The later step is performed by all MDCs in the network.

5.2.4.1 Collection Zones Reconciliation

At the end of the CZs creation phase, a situation may arise where two or more nearby MDCs has overlapping CZs. This situation can also occur after SZs construction step if an MDC updates its SZ after moving to a vicinity of other MDC, this scenario is described in Section 5.3. These situations can result in creating small SZs that contain MDCs close to the perimeter of their corresponding SZ.

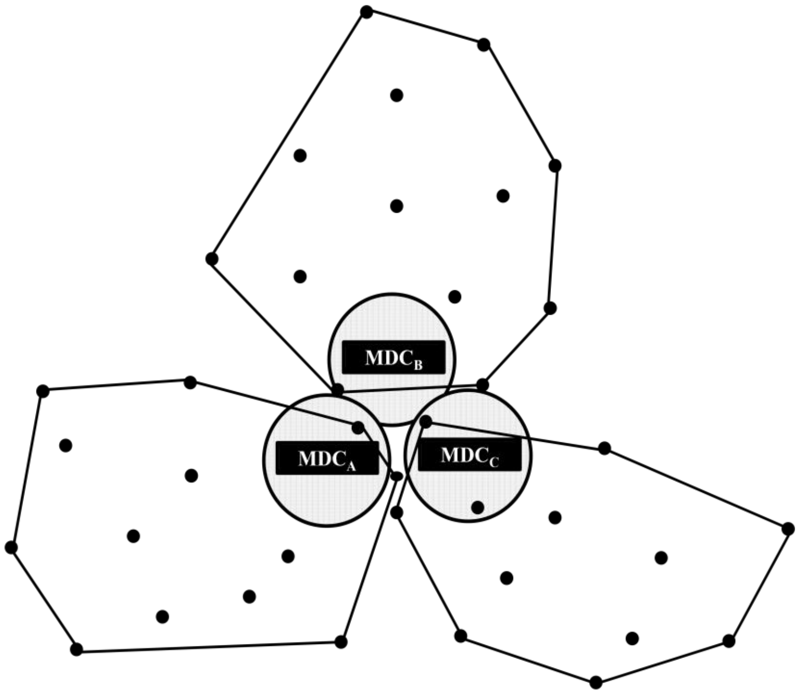


Figure 5.10: MDC merging situation

Consider the situation in Figure 5.10 where there are three MDCs that physically close to each other and have overlapped CZs. In this case, each MDC constructs a relatively small SZs. The MDCs will be located close to the perimeter of their SZs. Such situation is far from ideal due to: (1) It is possible that the MDC will very soon move outside its SZ. This results in a major SZs re-configuration at small time intervals, during which information delivery is interrupted. (2) Spatial events become more difficult to capture in smaller SZ without high-level coordination. (3) In the mobility management, because the update procedure will run within each SZ independently. To overcome the problem of over partitioning the network, a CZ reconciliation algorithm is designed for choosing the appropriate MDC to serve sensor nodes connected to the other MDCs.

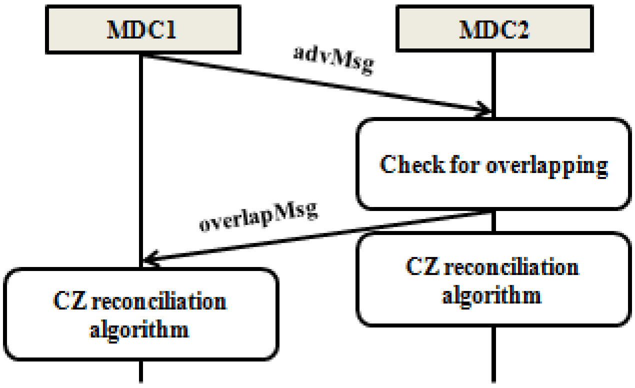


Figure 5.11: MDCs overlapping messages diagram

Figure 5.11 shows the steps MDCs follows to discover an overlap. MDCs checks for CZs overlapping when it directly receives an advertisement message from another MDC. If the distance separating two MDCs is less than the length of their CZ diameter, then the overlap is detected. Upon CZ overlap discovery, the discovering MDC sends an $overlapMsg$ to the advertising MDC. Then, both MDCs execute the CZ reconciliation algorithm described below.

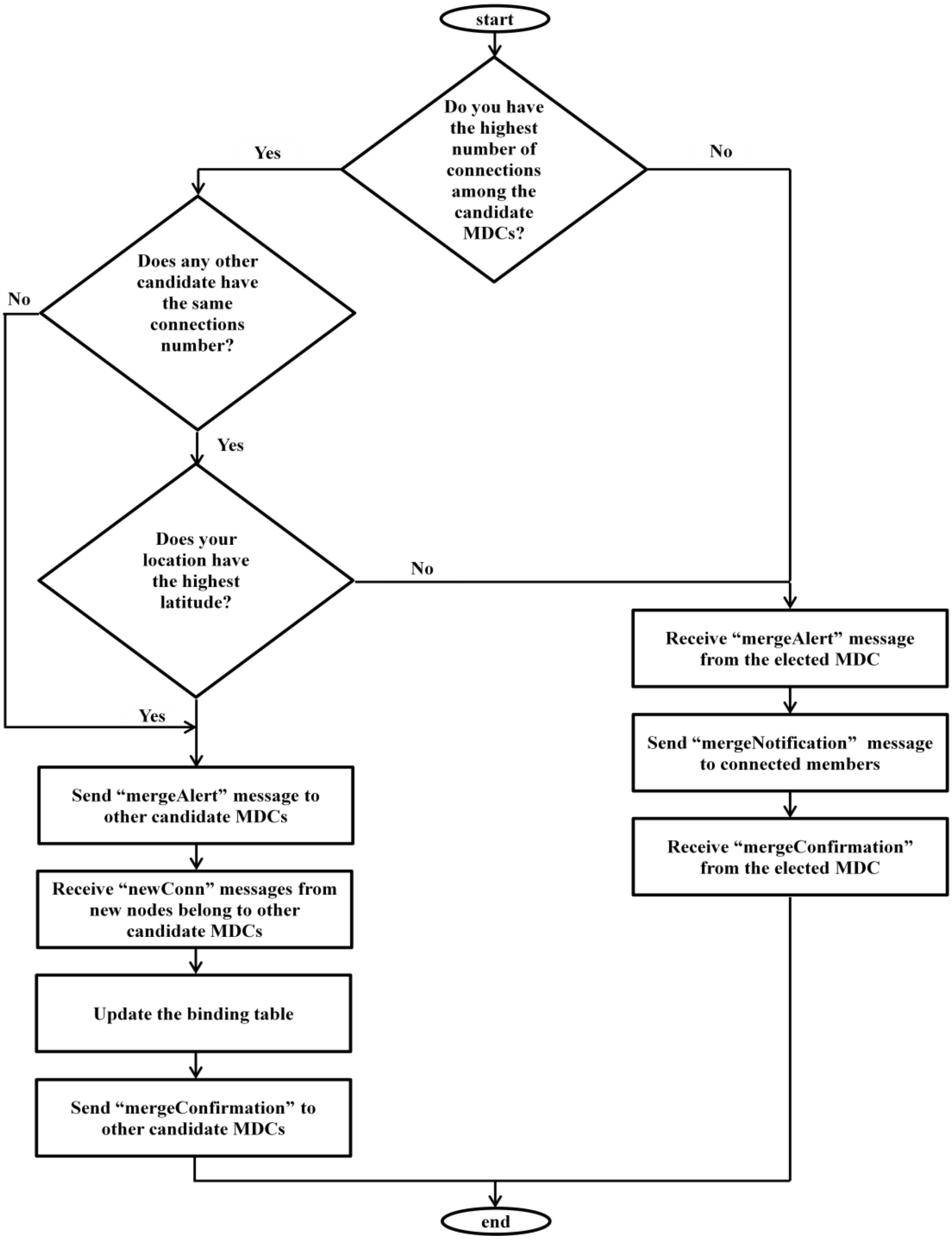


Figure 5.12: Collection zones reconciliation

Figure 5.12 illustrates the CZs reconciliation algorithm. Candidate MDCs start by finding the MDC that have more members in its binding table. This MDC, called primary MDC, is chosen to form the new SZ. The members of other SZs will be transferred to the new MDC. The MDC with more members is retained as it has more members in its SZ, which incurs higher cost to transfer them to a different SZ. When two candidate MDCs have equal number of members, the MDC that has higher latitude is retained.

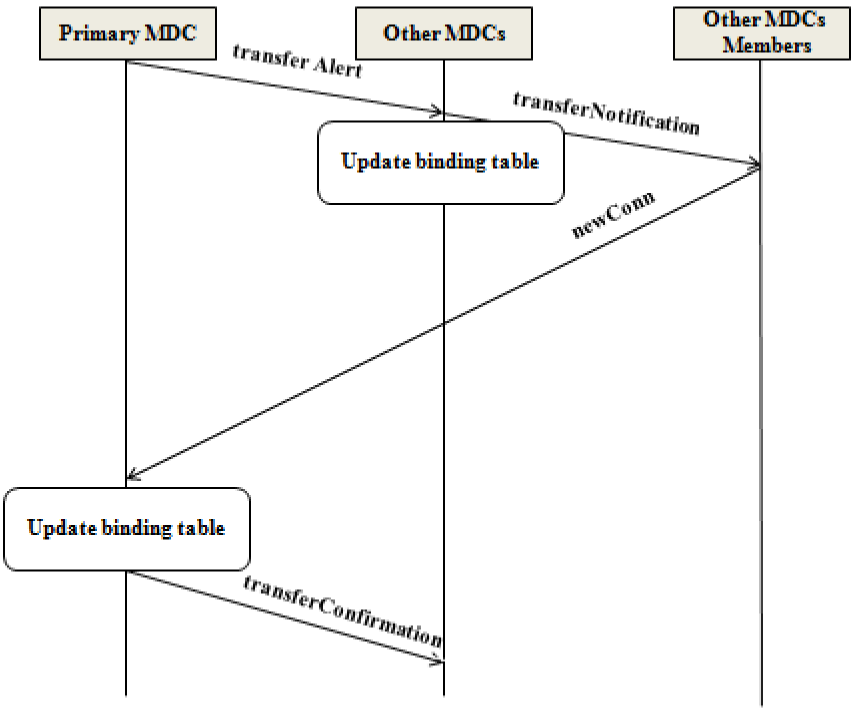


Figure 5.13: CZs reconciliation algorithm messages diagram

Figure 5.13 is a step-by-step illustration of the messages exchanged during this process. The primary MDC sends a $mergeAlert$ message to other involved MDCs. The receiving MDCs send $mergeNotification$ messages to their members. Each member sends $newConn$ message to join the primary MDC. Upon receiving the message by the primary MDC, it creates an entry for the new members to its binding table. Finally, the primary MDC sends a $mergeConfirmation$ messages to the other MDCs, then they update their binding tables.

5.2.4.2 Constructing the Convex Hull

This subsection presents the details of algorithm for the construction of the convex hulls. Convex hulls are constructed to determine the sensor nodes on the boundary of SZs and form groups. The convex hull construction is based on Graham scan algorithm [113]. The algorithm first explicitly sorts the nodes in $O(n\, log\, n)$ and then applies a linear-time scanning algorithm to finish building the hull. To compute the convex hull H, the function CH() performs the following three phases.

Phase I. select an anchor point (base node) $p0$ in Q, normally this is the node with the minimum y-coordinate. In case of a tie, the leftmost node (minimum x-coordinate) in the set is selected.

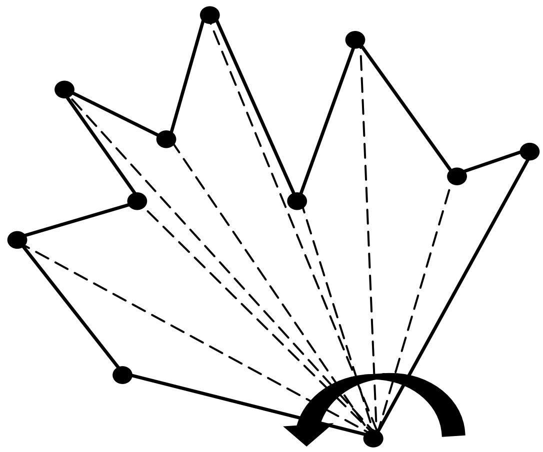


Figure 5.14: A simple polygon formed in the sorting phase of Graham's scan

Phase II. Sort the remaining nodes of Q (that is, $Q - \{p0\}$) lexicographically by polar angle, measured in radians. Interior nodes on the ray can not be convex hull points and are excluded during sorting. Once the nodes are sorted, they are connected in counter clockwise order with respect to the anchor node $p0$. The result is a simple polygon as shown in Figure 5.14. Note that no explicit computation of angles is performed by the algorithm.

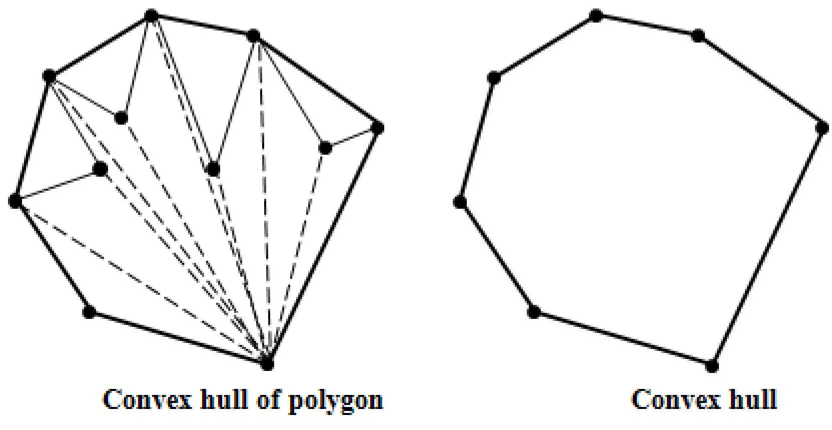


Figure 5.15: A simple convex hull formed in phase III of Graham's scan

Phase III. After pushing the anchor node $p0$ onto the stack SK, nodes are scanned in counter clockwise order, maintaining at each step a stack SK containing a convex chain surrounding the nodes scanned so far. At each node the following test is performed:

a. If pi forms a left turn with the last two points in the stack SK, or if SK contains fewer than two points, then push pi onto the stack SK.

b. Otherwise, pop the last point from the stack SK and repeat the test for pi.

The process halts when we return to the anchor point $p0$, at which point stack SK stores the vertices of the convex hull of Q in counter clockwise order. Figure 5.15 presents a convex hull after performing Phase III.

Let the vector $\overrightarrow{q_n q_{n+1}}$ represent the line segment between the last two nodes in the stack SK. To demine that a new node p is on the left of the line segment $\overrightarrow{q_n q_{n+1}}$, the MDC applies the right hand-rule, by checking the orientation of the cross product $\overrightarrow{q_n q_{n+1}} \times \overrightarrow{q_n p}$ which is equivalent to:

$$d = \left(q_n^{(y)} - q_{n+1}^{(y)}\right) p^{(x)} + \left(q_{n+1}^{(x)} - q_n^{(x)}\right) p^{(y)} + q_n^{(x)} q_{n+1}^{(y)} - q_n^{(y)} q_{n+1}^{(x)}$$

Then, the node p is left of the line segment $\overrightarrow{q_n q_{n+1}}$ if the result of equation is positive $d > 0$.

Algorithm 4: Graham Scan Algorithm

```
Input: a  set of points S = {P = (P.x,P.y)}

Select the rightmost lowest point P0 in S
Sort S radially (ccw) about P0 as a center {
    Use isLeft() comparisons
    For ties, discard the closer points
}
Let P[N] be the sorted array of points with P[0]=P0
Push P[0] and P[1] onto a stack Ω
while i < N
{
    Let PT1 = the top point on Ω
    If (PT1 == P[0]) {
        Push P[i] onto Ω
        i++     // increment i
    }
    Let PT2 = the second top point on Ω
    If (P[i] is strictly left of the line  PT2 to PT1)
{
        Push P[i] onto Ω
        i++     // increment i
    }
    else
        Pop the top point PT1 off the stack
}

Output: Ω = the convex hull of S
```

5.3 Network Partitions Maintenance

Logical zones (CZs and SZs) membership necessitates regular updates. Updating CZs is described in Subsection 5.2.3. This section provides the complete picture of how network partitioning is maintained. For handling changes in network topology due to frequent MDCs mobility, the proposed updating mechanism is triggered periodically by MDCs. The updating mechanism provides a continues process to keep track of changes in the network. The update mechanism is characterised by low communication overhead. To reduce the delay in implementing performance-critical logical zones updates, the update mechanism provide local checks and calculations performed by MDCs, sensor nodes only participate in the process when the MDC detect a change. This mechanism is energy efficient since updates are limited in scope; only the transferring MDCs and interconnected neighbouring nodes are to be aware of the handover.

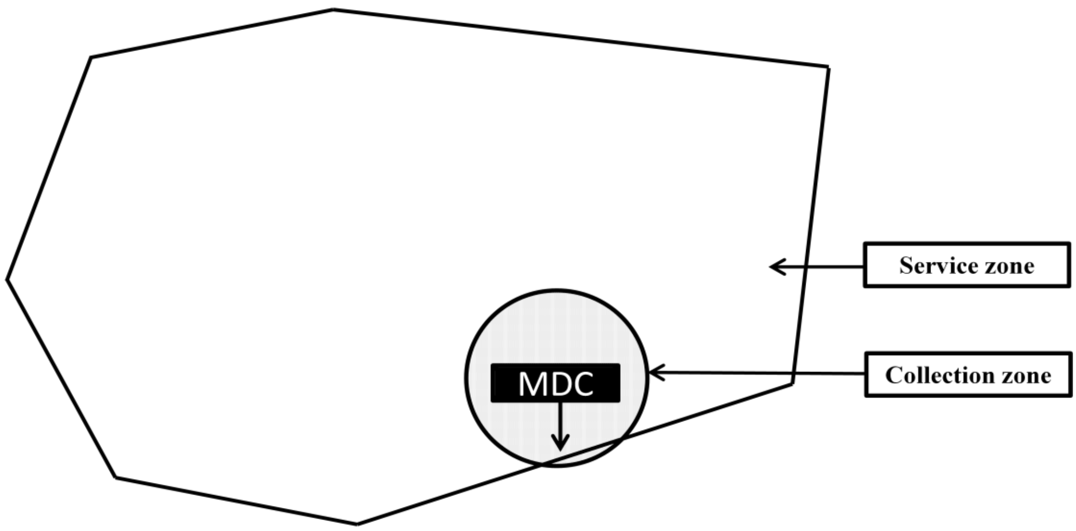


Figure 5.16: Service zone update

When an MDC is moving out of its CZ, a new CZ is created and nodes belonging to the corresponding SZ are reconfigured. When an updated CZ crosses its defined SZ boundary, the previously constructed SZ is destroyed and a new SZ will be constructed. SZ update may remove nodes that are no longer in an MDC vicinity, or add nodes disconnected from another SZ. Figure 5.16 shows an MDC moving to the south direction and out of its SZ. The MDC starts by scanning its binding table to determine all nodes that are more than d far from it. d is the maximum distance of a node to the MDC. These nodes are disconnected by a $disConn$ message. Next, the MDC sends advertisement messages to the new sensor nodes that are within the d distance. Upon receiving the advertisement, unconnected sensor nodes respond by a $newConn$ message. The previous procedure excludes nodes that already connected to the original service area, these nodes only forward the advertisements. Figure 5.17 shows the details of the partitions local update mechanism.

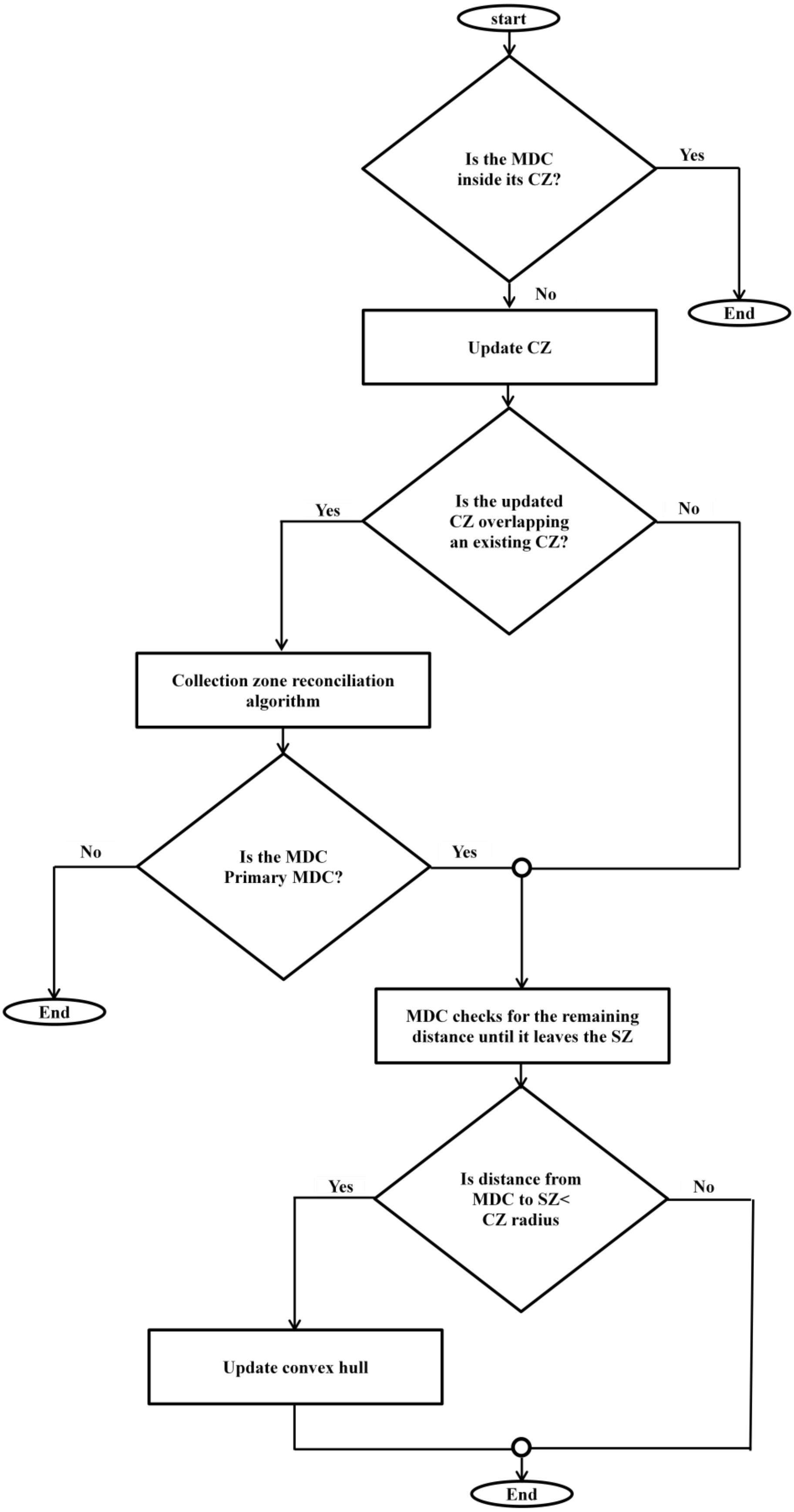


Figure 5.17: SZ updating mechanism

Partitions updating mechanism is periodically performed by the MDCs. An MDC checks whether it is inside its CZ by comparing the distance between its location and the centre of its CZ with the CZ radius. If that distance more than the radius of its CZ, then the MDC is not inside its CZ and the CZ will be updated. It is important to point out that the new updated CZ could overlap an existing CZ. In this case, CZs reconciliation algorithm is executed, and thus, one SZ would be constructed for both MDCs.

After a CZ update, the MDC calculate the estimated remaining distance and time in their current SZs. These information are used by the MDC to determine when to update its SZ. Intuitively, the MDC will intersect one of the SZ edges after some certain time. To calculate an estimation for this time and remaining distance for the MDC inside its SZ, the intersection point of the MDC and the SZ edge must be predicted.

Let S be the line segment between endpoints, P_0 and P_1, the MDC current position and its new location after it crosses the SZ, respectively. The extended line through P_0 and P_1 is given by the parametric equation:

$$P(t) = P_0 + t(P_1 - P_0) = P_0 + tv$$

with $v = P_1 - P_0$ the line direction vector. Then the segment S contains those points $P(t)$ with $0 \leq t \leq 1$.

Let a convex hull CH be given by n vertices $V_0, V_1, \dots, V_{n-1}$ going counterclockwise around the hull, and let $V_n = V_0$. Also let e_i be the i^{th} edge (line segment) V_iV_{i+1} for $i = 0, n-1$; and $ev_i = \left(e_{i1,} e_{i2}\right) = V_{i+1} - V_i$ be the edge vector. Then, an outward-pointing normal vector for e_i is given by $n_i = -ev_i{}^{\perp} = (e_{i2}, -e_{i1})$, where "$\perp$" is the 2D perpendicular operator.

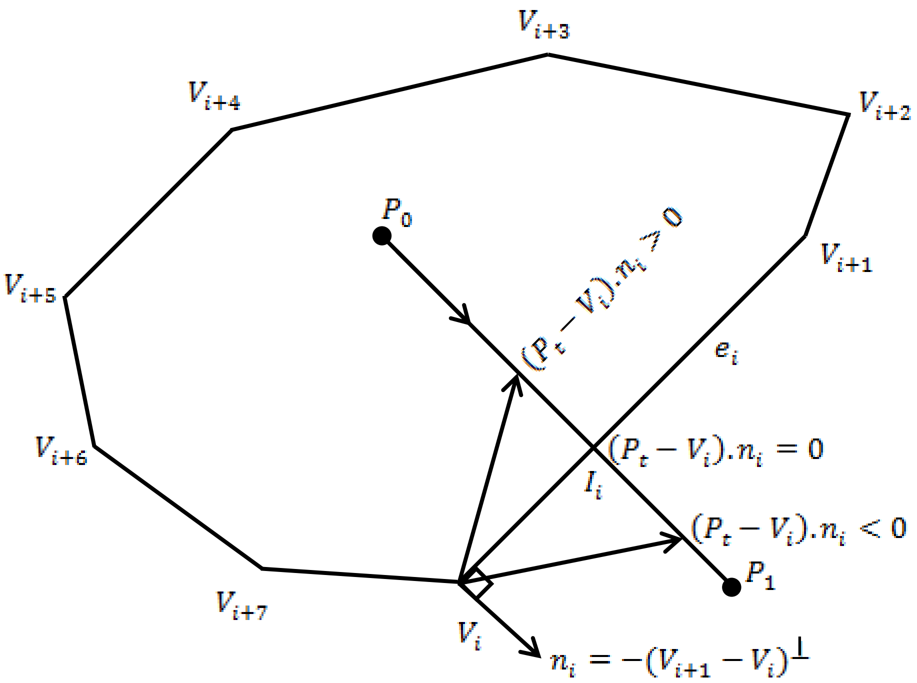


Figure 5.18: MDC and SZ intersection

To determine the hull edge that will intersect with the line segment P_0P_1, we scan the hull edges checking if the vector from V_i to P_0 points to the outside of the edge. When $(P_0 - V_i).n_i < 0$, there is no intersection with the edge, so ignore this edge, and continue processing the other edges.

As indicated in Figure 5.18, intersection occurs when $(P(t) - V_i).n_i = 0$, since any vector parallel to the edge e_i is perpendicular to the edge normal vector. substituting for $P(t)$ and solving for t, we get:

$$t_i = \frac{(V_i - P_0).n_i}{(P_1 - P_0).n_i}$$

at the intersection point $I_i = P(t_i)$, t is plugged back into the first equation

$$P(t) = P_0 + t(P_1 - P_0)$$

The pseudo-code in Algorithm 5 provides the details of this algorithm.

Algorithm 5: MDC and SZ intersection

```
    Input: a 2D segment S from point P0 to point P1
       a 2D convex polygon CH with n vertices
V0,...,Vn-1,Vn=V0

    if (P0 == P1) then S is a single point, so {
        test for point inclusion of P0 in CH; and
        return the test result (TRUE or FALSE);
    }
    Initialize:
        t = 1 for the minimum intersecting segment
parameter;
        dS = P1 - P0 is the segment direction vector;

    for each (edge ei = ViVi+1 of CH; i=0,n-1)
    {
        Let ni = an outward normal of the edge ei;
        N = - dot product of (P0-Vi) and ni;
        D = dot product of dS and ni;
        if (D == 0) then S is parallel to the
edge ei {
            if (N < 0) then P0 is outside the edge ei
                return FALSE since S cannot
intersect CH;
            else S cannot leave CH across edge ei {
                ignore edge ei and
                continue to process the next edge;
            }
        }

        Put t = N / D
    }

    Output:  P(t) = P0 + t * dS
```

5.3.1 Orphaned Nodes

Disconnecting nodes during updating service areas can result in 'orphaned nodes'. An orphaned node is a node that is not connected to any MDC. Such a node loses its connectivity neighbouring nodes, or is unable to obtain an advertisement message from any of the MDCs as it is located in a far position from the MDCs. An orphaned node may keep attempting to connect to its previous parent. This lead to segmentation problems where the network is divided into many unconnected segments. This situation could also occur when MDCs are located distant from each other, and there is an unconnected nodes between the SZs. Such situation may lead to disconnections and loss of data from orphaned nodes and other parts of the network. To resolve the orphaned nodes problem, we opted to extend the SZs to the whole monitored area by using the following steps:

Step 1: If a node does not receive an advertisement from an MDC or a gets disconnected, it waits for a back-off interval.

Step 2: If the node still did not receive an advertisement, it uses reactive discovery approach by sending out an MDC solicitation message to its neighbours to obtain MDCs information.

Step 3: Neighbours forward the message to the MDC and wait for reply. In case of the neighbour is also orphaned, the node enters another back off interval to allow their neighbour to obtain the MDC information.

Step 4: The MDC sends its information to the forwarding node.

Step 5: The forwarding node receives the MDC information message and forwards it to the orphaned node.

Step 6: Upon receiving information about the surrounding MDCs, the orphaned node executes the optimal MDC selection algorithm. Choosing the optimal MDC is based on the connection expiration time (CET). Subsection 5.2.2.1 presents our MDC selection algorithm.

Step 7: Orphaned node chooses the optimal MDC and sends to it a newConn message.

Step 8: The chosen MDC waits for a backoff interval waiting for other newConn messages from other orphaned nodes.

Step 9: The chosen MDC updates its convex hull to join the orphaned nodes.

Unlike the exhausted (or dead) nodes, the orphaned nodes can still receive and transmit messages; thus it is possible to restore them to the network. Handling and minimising the number of orphaned nodes preserve their energy and reduce signalling overhead, which assists in balancing energy consumption. Connecting orphaned nodes alleviates network segmentation and energy depletion. Yet, an orphaned node joins the optimal MDC that keeps it connected for a longer period of time.

5.4 Summary

This chapter presents a low-cost solution to mobility management at the middle layer of our defined network model. The aim of this solution is to enable the application of static-oriented IE approaches to mobile or hybrid networks. To extract high quality information, we argue for partitioning the network dynamically into many SZs, where each SZ is managed by an MDC. SZs partitioning is based on convex hulls, which envelopes nodes with their serving MDC. An MDC algorithm was presented to allow nodes choose the most appropriate MDC. This algorithm incorporates several important factors in calculating the CET between a node and its MDC. Connecting a node to an MDC with long CET results in fewer network reconfigurations, and therefore, conserving bandwidth and energy. The SZs are dynamically updated and maintained by the resource-rich CZs. The CZs have the potential to be utilised by IE approaches that are based on logical grouping of nodes to deliver their intended functionality, e.g. for query scoping or dissemination.

Chapter 6

Implementation and Evaluation

6.1 Introduction

This chapter presents the results from simulation studies to evaluate the performance of our proposal: (DNP/SS) under diverse conditions of network density and mobility. The performance DNP/SS is compared against the MDC/PEQ protocol [95] (as summarised in Subsection 2.8.1). MDC/PEQ is the closest rival to DNP/SS. It is widely cited in the literature and published in a reputable journal.

In their paper, the authors of MDC/PEQ provide most of the algorithm and its implementation details, which makes it possible to reproduce their results. Importantly, MDC/PEQ follows similar design principles to the ones we use; to achieve common objectives, i.e., to conserve energy and improve QoI by managing MDCs mobility without delaying the conveyed data. Therefore, direct comparison in terms of various performance metrics can be made between DNP/SS and MDC/PEQ.

MDC/PEQ measures the performance of its data dissemination approach in terms of packet delivery ratio, packet delay, energy consumption, and the number of hops between a node to its MDC. However, we use additional performance metrics and a set of carefully designed experimental setups to thoroughly evaluate DNP/SS under various run time conditions.

For the first set of experiments, we study the performance of the self-stabilisation algorithm. We investigate the impact of SS on the overall performance of the DNP/SS. The performance of DNP/SS is compared to the performance of DNP on top of the Optimized Link State Routing (OLSR) [114]. OLSR is used to manage nodes mobility.

The second set of experiments aims to evaluate the effectiveness of DNP/SS compared to MDC/PEQ. In these experiments, the exact simulation setup and parameters of MDC/PEQ are used. However, their simulation setup only suits small-sized networks. They assume that the communication range for sensor nodes is $25\,m$. The area of their simulated network deployment area is $150m\ x\ 150m$, which is relatively small compared to real-world applications such as battlefield surveillance. For this reason, the third group of experiments use a more realistic simulation setup based on iMote2 [115] hardware platform specifications, and uses mobility patterns that exist in real-world military scenarios.

In the fourth group of experiments, a synthetic mobility plan is designed to simulate the real-world movements of armoured vehicles. The primary goal of these experiments is to evaluate the performance of DNP/SS under various mobility conditions. The fifth group of experiments evaluates the success of deploying static-oriented IE approaches over hybrid or mWSN environments. To achieve this, a recent energy efficient, hybrid IE approach [112] (as summarised in Section 2.8.2) is chosen to be deployed on top of DNP/SS. The approach is applied to extract information regarding a pre-defined event. The effect of DNP/SS on the IE process is investigated in terms of the information's accuracy, timeliness, and energy efficiency.

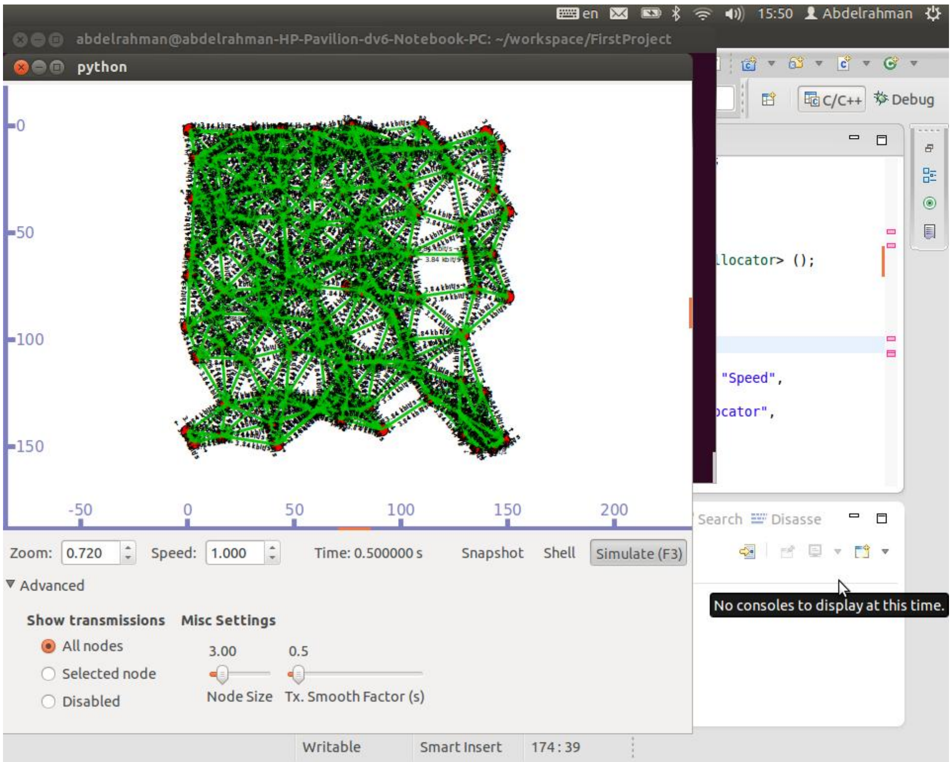
The rest of the chapter is organised as follows. Section 6.2 the simulation tool used to perform the experiments is presented. In section 6.3, defines the evaluation metrics used to evaluate the work in this thesis. Section 6.4 presents the simulation setup and the result of the analysis respectively.

6.2 Simulators

Evaluating protocols and algorithms in WSNs is often done by using one of the following techniques: analytical methods, simulation tools, or testbeds. The constraints and complexity of WSNs often cause analytical methods to be unsuitable or inaccurate [116]. Furthermore, the amount of algorithms evaluated through testbeds is relatively low. This is due to deployment cost and application dependence of WSNs. Consequently, simulation is the most adopted technique of evaluating WSNs, allowing low cost, rapid evaluation of new WSNs algorithms. As there are many simulation tools have been developed, we have published a comprehensive survey of simulation tools and methods for WSNs [117].

Among the various simulation tools, we have chosen to use NS-3 simulator to evaluate the work proposed in this thesis. This is due to many reasons: (1) It is a well-known open-source discrete-event network simulator; (2) Results provided by NS-3 are widely accepted in the research community; (3) NS-3 supports nodes mobility; (4) NS-3 supports multiple radio interfaces. NS-3 is an improvement upon the core architecture of the well-known NS-2 simulator.

NS-2 [118] is an object-oriented discrete event simulator targeted at networking research. It is an open source network simulator originally designed for wired, IP networks. The NS-2 simulation environment offered great flexibility in studying the characteristics of WSNs because it includes flexible extensions for WSNs. NS-2 has a number of limitations: (1) It puts some restrictions on the customisation of packet formats, energy models, MAC protocols, and the sensing hardware models, which limits its flexibility; (2) The lack of an application model makes it ineffective in environments that require interaction between applications and the network protocols; (3) It does not run real hardware code; (4) It has been built by many developers and contains several inherent known and unknown bugs; (5) It does not scale well for WSNs due to its object-oriented design; (6) Using C++ code and oTcl scripts makes it difficult to use.



6.1: NS3 simulator in action

To overcome the above drawbacks the improved NS-3 simulator [119] was developed. NS-3 supports simulation and emulation. It is totally written in C++, while users can use python scripts to define simulations. Hence, transferring NS-2 implementation to NS-3 require manual intervention. Besides the scalability and performance improvements, simulation nodes have the ability to support multiple radio interfaces and multiple channels. Furthermore, NS-3 supports a real-time schedule that makes it possible to interact with a real systems [119]. For example, a real network device can emit and receive NS-3 generated packets. NS3 supports a live simulation visualiser (PyVis), to figure out whether mobility models are what the developer expects, where packets are being dropped, etc. Figure 6.1 depicts NS3 simulator in action.

6.3 Evaluation Metrics

In this chapter, the performance of DNP/SS is to be evaluated via simulation with respect to the following metrics:

- **End to End delay (E2Ed)**:

  Recent studies in WSN focused on E2Ed as an important QoS metric [120]. It is measured as the time taken by a packet to be transmitted across the network from the source to its destination [121]. This includes all possible delays such as the one caused by the queuing, transmission, propagation, and processing delay. The average delays of all n nodes are given by the following equation:

$$E2Ed = \frac{\sum(arrive\ time - send\ time)}{\sum sent\ packets}$$

Where $send\ time$ is the time a packet is generated, $arrive\ time$ is the time a packet arrives at its final destination, and $sent\ packets$ is the number of data packets generated at sensor nodes and received by the sink.

- **Packet Delivery Ratio (PDR)**:
  In several studies, PDR is considered as one of the prime measures of protocol or algorithm effectiveness. PDR is the ratio of packets that are successfully delivered to a destination compared to the number of packets that have been sent by the sender(s) [122]. The packets which arrived late at the destination are considered ineffective. PDR is defined as:

$$PDR = \frac{\sum Number\ of\ packets\ received}{\sum Number\ of\ packets\ sent} * 100\%$$

- **Average Energy Consumption per Node**:
  Since radio communication is the most power-hungry operation [123], the energy consumption of DNP/SS is measured as the cost of mobility management added to the cost of IE. The average energy consumption of a sensor node is directly related to the operational lifetime of the network.

$$Average\ Energy\ Consumption = \frac{\sum(I_{energy} - R_{energy})}{n}$$

  Where I_{energy} is the initial energy of a sensor node, R_{energy} is the residual energy of the sensor node when the simulation ends, and n is the number of sensor nodes in the network.

- **Mobility Management Cost**: Is the total energy consumed by mobility management techniques to react to changes triggered by the movement of nodes. It does not include the cost to transmit user data. The recording of these measurements start after the setup phase is completed. Mobility management cost is calculated as the energy difference between the available energy at the last stable state ($Energy1$), and the available energy when an network becomes stable again after a certain mobility event ($Energy2$) - it is indicated by:

$$Mobility\ management\ cost = \sum(Energy1 - Energy2)$$

- **IE Cost**: The accumulated energy consumed in order to extract information. IE cost is calculated as the energy difference between the total energy available just before a request for information is made ($Energy3$), and the total energy available after information is delivered to the end user ($Energy4$). It is given by:

$$Mobility\ management\ cost = \sum(Energy3 - Energy4)$$

- **Network Lifetime**:

  Network lifetime is measured as the time duration before a portion of the network is exhausted [124]. A network is considered exhausted when the energy level becomes zero for 50% of total node population. Network lifetime can be calculated as the time difference between when the network becomes exhausted ($Tfinish$), and the time when it started operating ($Tstart$).

$$Network\ lifetime = Tfinish - Tstart$$

- **Dissemination Ratio (R_{nodes}):**

  The ratio of sensor nodes that become engaged in data dissemination in response to IE request [48]. This ratio is given by:

$$Rnodes = \frac{T_{diss}}{T_n}$$

  Where T_{diss} is the total number of data disseminating nodes, and T_n is the total number of nodes in the network.

- **Number of Transmissions (N_{comm}):**

  N_{comm} can be measured as the total number of data transmissions in the network [48]. This metric directly influences the energy consumption of a protocol, so one should try to minimise the Ncomm to enhance energy conservation at sensor nodes [125, 126]. The number of data transmissions is given by:

$$\text{Ncomm} = \sum Data\ transmissions$$

6.4 Self-stabilisation Performance Evaluation

In this experiment, we study the performance of the self-stabilisation algorithm within a large-scale network. The effectiveness of SS is studied by running DNP without and on top of SS. In the absence of SS, a standard routing protocol (OLSR) is used to manage nodes' mobility at the network layer of the WSN protocol stack. The key concept used in OLSR is that of multipoint relays (MPRs). Only nodes selected as MPRs can relay broadcast packets received from their selectors. This technique substantially reduces the message overhead as compared to a classical flooding mechanism. The protocol is particularly suitable for large-scale and mobile ad hoc networks as the technique of MPRs works well in this context.

6.4.1 Simulation Setup

1000 sensor nodes were deployed at random in a square region of $1000m\ x\ 1000m$. Sensor nodes have wireless radio range of 75 meters. The transmission and reception power of a sensor node is set to $0.033\ J/s$. 500 data sources are chosen randomly to generate $2\ pkts/s$ during the entire simulation. The packet size is $32\ bytes$ for all types of messages except data packets, which are set to $64\ bytes$. The initial energy of sensor nodes is enough to complete the simulation. 50% of sensor nodes are mobile, and their speed can reach up to $3\ m/s$. They move according to the random waypoint mobility model [127, 128].

The number of MDCs is set to 50. They are also deployed randomly and their mobility speed reaches $5\ m/s$, they move according to the random waypoint mobility model. Their wireless radio range can reach $300\ m$. The sink node is located in the centre of the simulation area, and has wireless radio range of $300\ m$. A summary of the simulation parameters and their respective values are shown in Table 6-1. The simulation parameters' values are applicable to the iMote2 hardware platform.

Table 6-1: Self-stabilisation simulation parameters

Parameter	Value
Number of nodes	1000
Simulation area	1000 m x 1000 m
Wireless radio range (SN)	75 m
Wireless radio range (MDC)	300 m
Number of source node	500
Source nodes data rate	2 pkts/s
Number of MDCs	50
MDC velocity	20 m/s
Bandwidth	250 Kbps
Data packet size	64 Byte
TX power dissipation	0.033 W
RX power dissipation	0.033 W
Mobility Model	Random waypoint

6.4.2 Experiment 1: End-to-End Delay

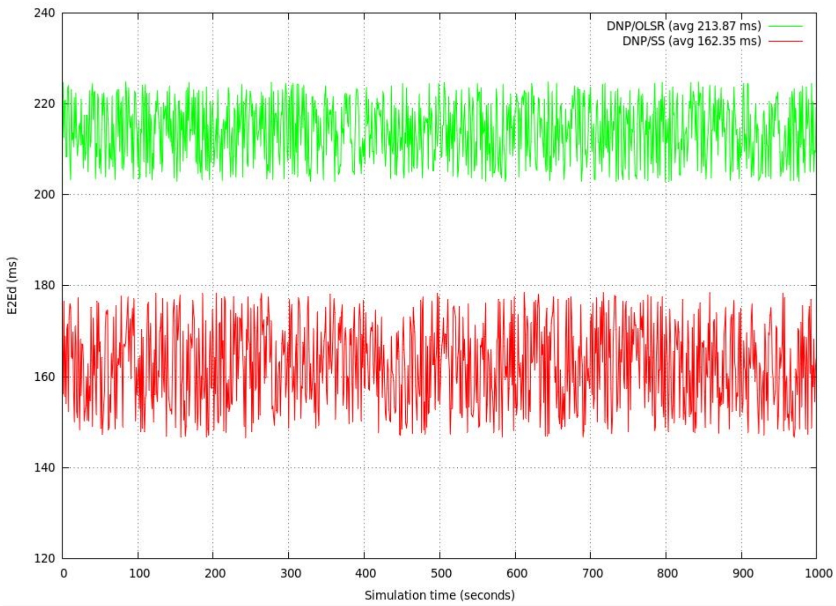


Figure 6.2: Comparison of timeliness of DNP/SS and DNP/OLSR

Aim:

This experiment aims to measure the time taken to send a packet from a sensor node to the sink in both DNP/SS and DNP/OLSR. The experiment extracts variations of average end-to-end delay with respect to simulation time.

Results and discussion

In this experiment, 50% of sensor nodes are selected randomly to produce data at a rate of $2\,packets$ per second. Figure 6.2 shows the end-to-end delay in both DNP/SS and DNP/OLSR. A comparison of the two results reveals that SS decreased the end-to-end delay by around $24\,\%,$ from $214\,ms$ to $162\,ms$. There are several factors accounting for this outcome. Firstly, the SS algorithm minimises data transmission interruption times and maintains high network connectivity by rapidly responding to topological changes. Secondly, the SS reduces the amount of communication and computation to converge to a stable state; therefore, it frees more bandwidth for data transmissions. On the other hand, OLSR uses larger signalling traffic as it periodically floods the status of its links. Moreover, each node re-broadcasts link state information received from its neighbour.

6.4.3 Experiment 2: Packet Delivery Ratio

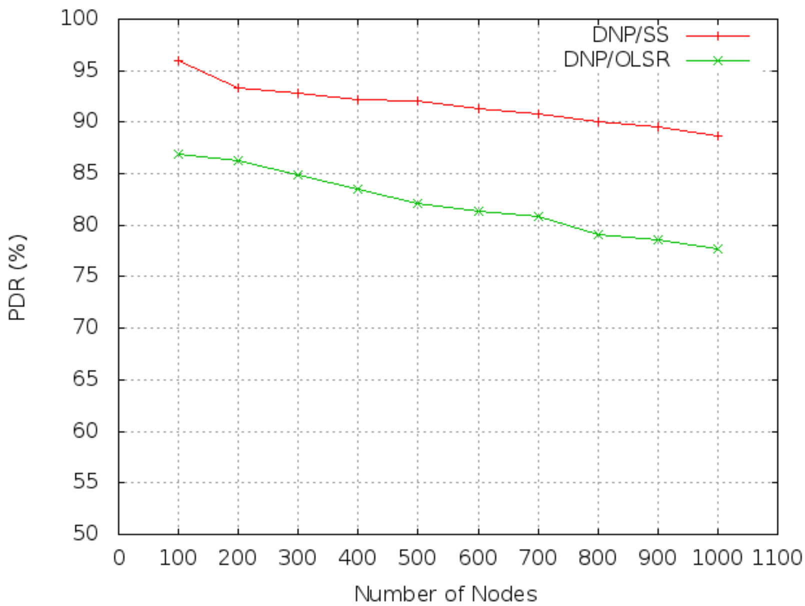


Figure 6.3: Comparison of PDR against number of nodes, DNP/SS and DNP/OLSR

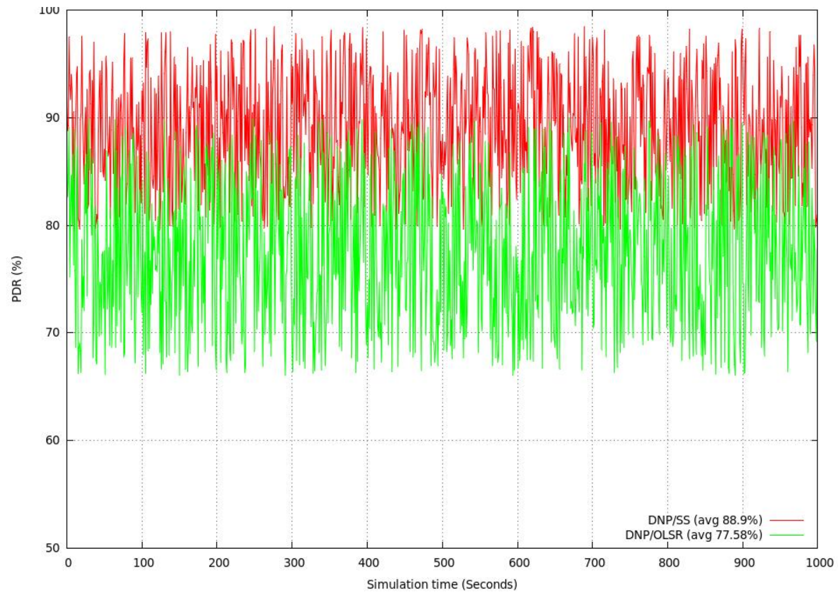


Figure 6.4: Comparison of PDR against simulation time, DNP/SS and DNP/OLSR

Aim:

This experiment aims to measure the SS algorithm's ability to maintain high PDR with given timeliness constraint and in the presence of communication failures. The experiment was repeated 10 times, varying the deployment area size and the number of nodes from 100 to 1000.

Results and discussion

Figure 6.3 plots the average PDR for DNP/SS and DNP/OLSR at different network sizes. It can be observed that in all network densities, the SS algorithm has performed better than its rival by up to 13%. This is due to the ability of SS to reduce the mobility management overhead, which avoids congestion and bottlenecks. Additionally, the SS algorithm converts the network to a stable state rapidly, which reduces the number of packets lost due to broken communication links. This also allows data packets to arrive at their destination within the given time constraints. In contrast, every node in OLSR periodically broadcasts a list of its MPR Selectors. This leads to the consuming of extra bandwidth, and thus higher packet loss. It is noticeable that SS performs well in large-scale networks. Figure 6.4 shows the PDR in a 1000 nodes network. This is expected as SS delivers its functionality in a distributed manner, i.e. all mobility updates are managed locally.

6.4.4 Experiment 3: Network Lifetime

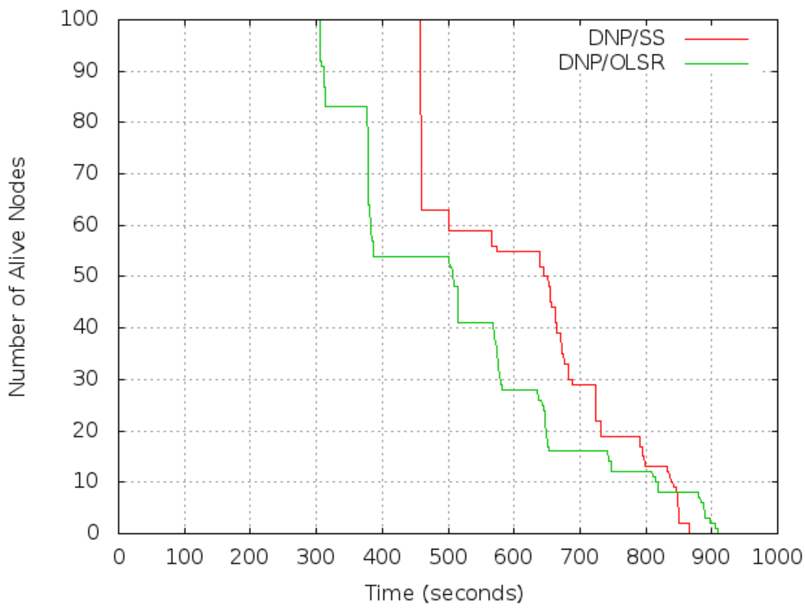


Figure 6.5: The number of alive nodes using DNP/SS and DNP/OLSR

Aim:

This experiment aims to measure the network lifetime when using DNP/SS and DNP/OLSR algorithms. The experiment uses two metrics, the time take for the first node to die (FND), and time taken for the last node to die (LND). From these metrics, we can evaluate the energy balancing among nodes.

Results and discussion:

In this experiment, the initial nodes energy was set to $8\,joules$. Figure 6.5 shows that SS outperformed OLSR by 33% when comparing FND readings. This energy saving is mainly due to the reduction of the amount of communication and computation to recover from interruptions and disconnections caused by nodes mobility. At the time of $830\,seconds$, only 8 nodes were left alive. After $80\,seconds$, those nodes died dramatically. Those 8 nodes were segmented and could not relay data to the sink. In effect, no data was successfully delivered to the sink during this period of time.

When the interval between FND and LND time is minimised, this indicates a more balanced energy consumption among all sensor nodes in the network. The interval between FND and LND in DNP/OLSR is $604\,seconds$, while the interval in DNP/SS algorithm is $404\,seconds$. Thus, energy balancing is improved by $33\,\%$. This is mainly due to reducing the amount of communication and computation to convert the network into a stable state in SS algorithm. Unlike in OLSR where the MPR nodes consume its energy faster than other nodes due to its important activity in the network, forwarding the packet from its selectors to the rest of the networks also relays data packets intended to its selector.

6.4.5 Experiment 4: Energy Consumption

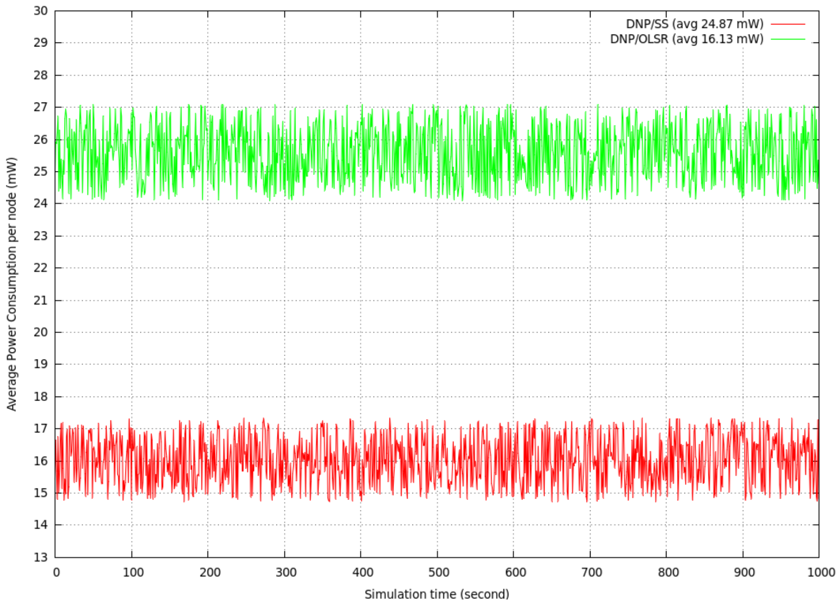


Figure 6.6: Power consumption per node in both DNP/SS and DNP/OLSR

Aim:

This experiment aims to evaluate energy consumption per node in both algorithms over the simulation time.

Results and discussion:

Figure 6.6 shows the energy consumption in both studied approaches. The graph shows that SS decreases energy consumption drastically when compared to OLSR. The overall average energy consumption is $16.1\ joules$, $24.8\ joules$ for DNP/SS and DNP/OLSR respectively. Hence, the SS algorithm significantly prolongs the network lifetime by approximately35%. The main factor accounting for this outcome is that the SS algorithm considerably reduces the amount of communication and computation to converge to a legitimate state. In SS algorithm, mobility maintenance is managed locally. Moreover, SS allows planning for mobility management in advance; considering that upcoming mobility updates also contribute to reducing the cost of mobility management. However, OLSR uses more signalling traffic and power consumption. This is due to the high amount of the periodic broadcasts of the link state information.

6.4.6 Conclusion

The results from the previous set of experiments prove that the SS algorithm outperforms OLSR. SS demonstrated the best performance in terms of end-to-end delay, packet delivery ratio, network scalability, energy balancing, and energy consumption. Therefore, the SS algorithm effectively adapts to the mobility-driven topological changes. There are several factors accounting for this outcome. In first place, minimising service interruption times by asynchronously changing the topology and the state of nodes; secondly, by reducing the amount of communication and computation to converge the network to a stable state; thirdly, via performing mobility maintenance in a distributed manner.

6.5 DNP/SS Performance Evaluation

6.5.1 Simulation Setup

In this section, we evaluate the performance of DNP/SS against MDC/PEQ. To increase the reliability of measured results, this set of experiments used the exact simulation parameters applied in MDC/PEQ. A 100 sensor nodes were deployed at random in a square region $150m \, x \, 150m$. Sensor nodes have wireless radio range of $25 \, meters$. The transmission and reception power of a sensor node is set to $0.01488 \, J/s$ and $0.01250 \, J/s$ respectively. Regarding data sources, 50 sensor nodes are chosen randomly to generate $2 \, pkts/s$ during the entire simulation. The packet size was $32 \, bytes$ for all types of messages except data packets, which were set to $64 \, bytes$.

Initial energy of sensor nodes were enough to complete the simulation. The number of MDCs varied from 1 to 10; when not specified, the default number of MDCs is 5. They were deployed uniformly and have a mobility speed of $5 \, m/s$. They move according to the bounded random mobility model (BRMM) [129]. BRMM is a group-based mobility model that presents smooth movements patterns. MDCs have wireless radio range of $100 \, m$. The sink node is located on the border of the simulation area, and have a wireless radio range of $100\,m$. A summary of the simulation parameters and their respective values are shown in Table 6-2.

Table 6-2: MDC/PEQ simulation parameters

Parameter	Value
Number of nodes	100
Simulation area	150 m x 150 m
Wireless radio range (SN)	25 m
Wireless radio range (MDC)	100 m
Number of source node	50
Source nodes data rate	2 pkts/s
Number of MDCs	5
MDC velocity	5 m/s
Bandwidth	38.4 Kbps
Data packet size	64 Byte
TX power dissipation	0.01488 W
RX power dissipation	0.01250 W
Mobility Model	BRMM

6.5.2 Experiment 1: End-to-End delay

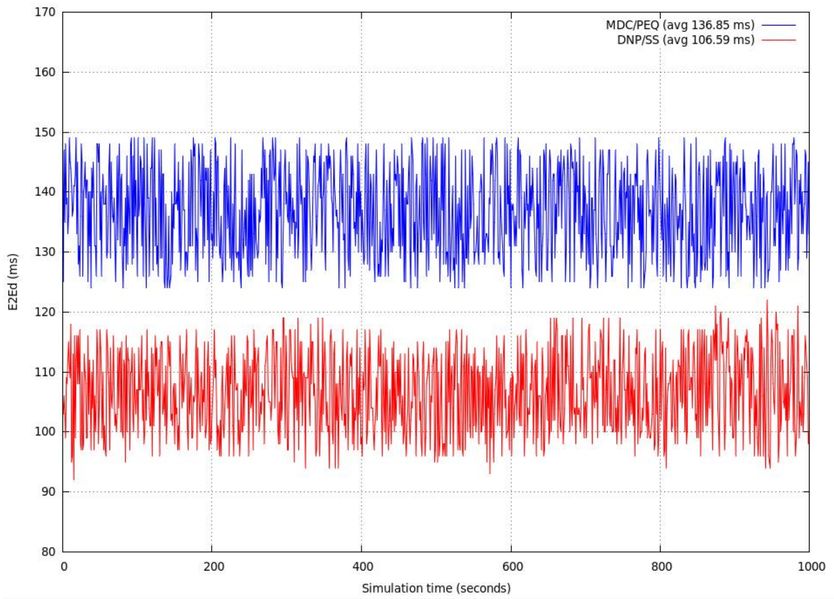


Figure 6.7: End-to-end delay comparison between DNP/SS and MDC/PEQ

Aim:

This experiment aims to measure the timeliness of both DNP/SS and MDC/PEQ over the simulation time, whereby 50 randomly selected nodes generate traffic on events of interest at a rate of 2 packets per second, with a packet size of $64\ Bytes$.

Results and discussion:

Figure 6.7 demonstrates that end-to-end delay is substantially reduced when using DNP/SS. The average end-to-end delay is reduced from $136\ ms$ in MDC/PEQ to $106\ ms$ in DNP/SS, i.e., information timeliness has improved by 22% using DNP/SS. There are several reasons for this outcome. MDC/PEQ only considers signal strength in selecting the next serving MDC, while DNP/SS considers direction, distance and speed. Hence, DNP/SS has well delimited dynamic groups of nodes that use less routes, updates and topology reconfigurations. The partitioning technique used in DNP/SS reduces signalling overhead by localising mobility management traffic, consequently with a higher utilisation for the network's bandwidth for data transmission. Moreover, our optimal MDC selection predicts the future disconnection time, and hence, nodes use short paths to MDCs that last for longer time. This significantly shortens the data transmission and queuing delay.

6.5.3 Experiment 2: Packet Delivery Ratio

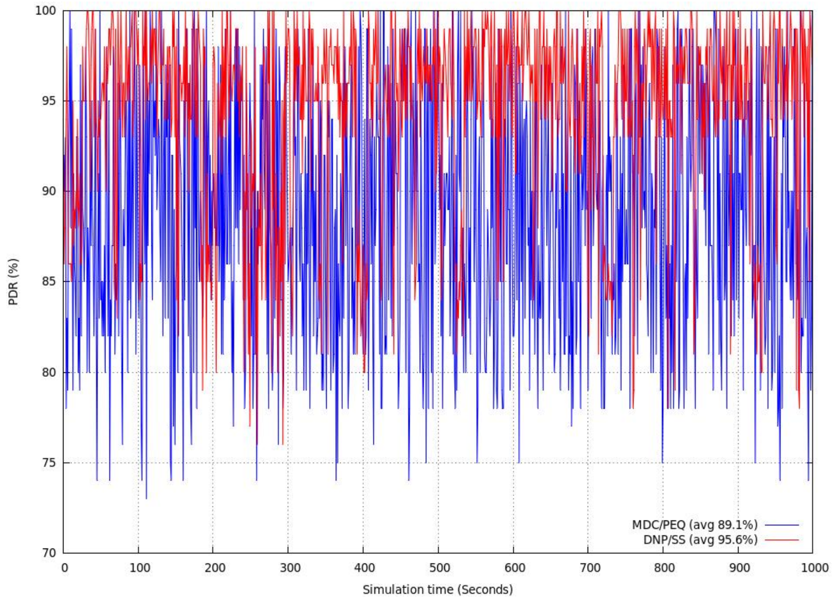


Figure 6.8: Packet delivery ratio comparison between DNP/SS and MDC/PEQ

Aim:

This experiment aims to measure the studied algorithms ability to maintain high packet delivery ratio within a given timeliness constraint in spite of some communication failures.

Results and discussion:

Figure 6.8 shows that PDR of DNP/SS outperforms the MDC/PEQ protocol. The average PDR has improved from 89.1% to reach 95.6% for the tested scenario. This improvement is due to grouping nodes into smaller SZs and localising mobility management traffic. It can be observed that the performance in term of packet delivery has a frequent fluctuation: it reaches 100 % when the network is stable (i.e., during time intervals that have no SZs reconfigurations). This sometimes drops suddenly below 80%, and may occur when sensor nodes execute the CZs reconciliation procedure, whereby the bandwidth utilisation increases dramatically due to the heavy exchange of reconfiguration messages. Another reason for this improvement would be our optimal MDC selection algorithm, which helps nodes to be connected for a longer period of time before its next reconfiguration; therefore, reducing IE service interruption times.

6.5.4 Experiment 3: Network Lifetime and Energy Balancing

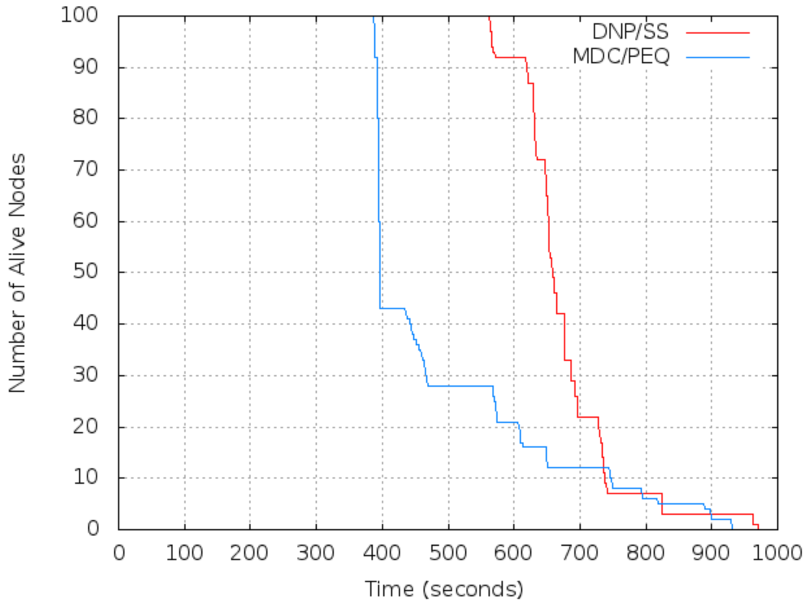


Figure 6.9: The number of Alive nodes comparison between DNP/SS and MDC/PEQ

Aim:

This experiment aims to measure the network lifetime in both studied algorithms by comparing the number of alive nodes over time. The experiment uses two important metrics: the time take for the First Node to Die (FND), and time taken for the Last Node to Die (LND). From these metrics, we can also assess the energy balancing among nodes.

Results and discussion:

In this experiment, the initial energy of sensor nodes is set to $4\,Joules$. Figure 6.9 shows the number of nodes alive with respect to simulation time. MDC/PEQ consumes more energy but at the expense of some nodes having high energy consumption. These nodes correspond to MDC/PEQ clusters that have more route changes, and this has resulted in early FND. This explains why after a simulation time of $392\ seconds$, more nodes are alive in DNP/SS than in MDC/PEQ. DNP/SS performs better because its sensor nodes connect to the MDCs with the longest CET, thus avoiding frequent handoffs. As a result, if 50% node death is considered as a network lifetime measure, then DNP/SS substantially prolongs the network lifetime by 40.6% over MDC/PEQ protocol.

Another finding that was revealed by this graph is the improvement on energy balancing. Energy balancing can be measured by comparing the time interval between FND and LND. The time interval between FND and LND is $546\ seconds$ in MDC/PEQ and $407\ seconds$ in DNP/SS. Therefore, using energy balancing as a measure of performance, DNP/SS outperforms MDC/PEQ by 25%. This is primarily because DNP/SS updates CZs locally and constructs near equivalent size SZs. Furthermore, permitting orphaned nodes to join an existing SZ and participate in the communication helps in balancing the energy in the network. On the other hand, some MDC/PEQ sensor nodes consume more energy in receiving, processing and forwarding beacons. After $800\ seconds$ in simulation time, the rest of the alive nodes can not exchange packets among them, so there is no further packets flows. That is the reason why they live for a longer time.

6.5.5 Experiment 4: Energy Consumption

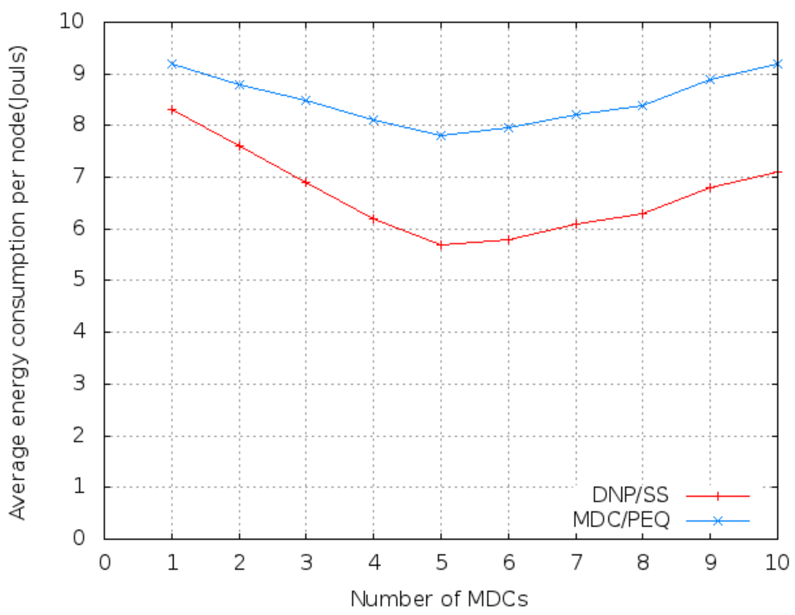


Figure 6.10: Energy consumption over number of MDCs, DNP/SS VS MDC/PEQ

Aim:

This experiment compares the energy consumption of the two studied algorithms by measuring the average energy consumption per node. The experiment was repeated ten times, varying the number of MDCs from 1 to 10.

Results and discussion:

As shown in Figure 6.10, a single MDC network consumes high energy as it causes high traffic load and resulting bottlenecks in the MDC vicinity. The energy consumption decreases gradually when increasing the number of MDCs from 1 to 5. This is due to multiple MDCs benefitting the network by balancing the load among MDCs. Furthermore, multiple MDCs can reduce the number of hops that data packets have to traverse. When increasing the number of MDCs to more than 5, the average energy consumption per node starts to increase moderately. This is due to the fact that when having more MDCs traversing the network, the number and frequency of SZs updates increase.

This results in a gradual rise in the amount of signalling overhead. It is important to note here that in our experiment, 5 MDCs is the optimal number of MDC as empirically proved by the authors of [130] (the optimal number of partitions in the network is estimated at about 5% of the total number of nodes in the network). DNP/SS performs better than MDC/PEQ in all cases. With this experiment, it is shown that DNP/SS is able to achieve 39.5% less energy consumption than MDC/PEQ. The self-stabilisation algorithm contributes to this result as it also helps to reduce the amount of network communications. Additionally, our MDC selection algorithm reduces reconfiguration messages in the network.

6.5.6 Experiment 5: Scalability

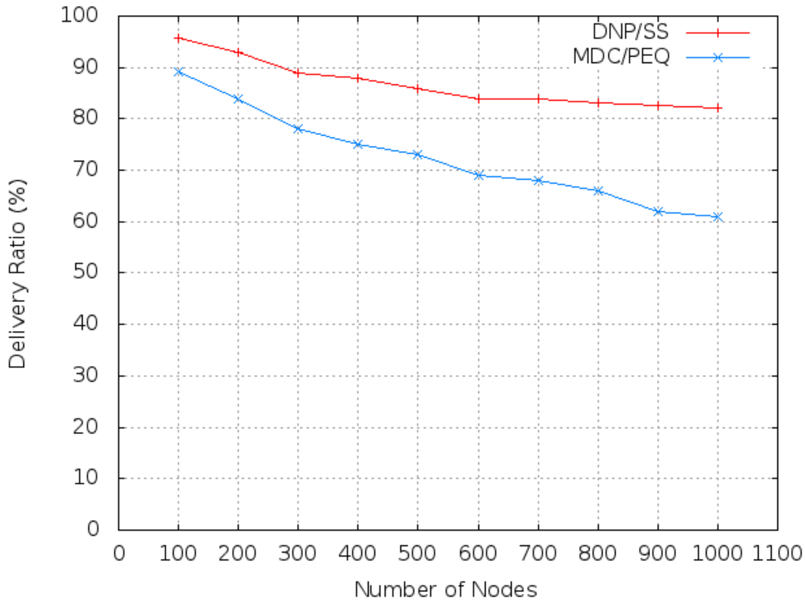


Figure 6.11: Delivery ratio over number of nodes, DNP/SS vs MDC/PEQ

Aim:

The aim of this experiment is to investigate the scalability of the studied algorithms. This is done by measuring the data delivery ratio at different network sizes (from 100 to 1000).

Results and discussion:

Figure 6.11 plots the results from ten experiments for each of the algorithms. It can be observed from the graph that the DNP/SS outperforms the MDC/PEQ, as it improves the delivery ratio by 4% in 100 nodes network, and 25% in 1000 nodes network. In DNP/SS, the slope falls slightly and gradually, while in MDC/PEQ the slope falls sharply. This is due to the self-stabilisation algorithm's effective way to locally and rapidly adapt to the changes in the network topology. The low topology management overhead and the logical grouping of nodes increases the scalability of DNP/SS. At a higher level of the network, also using SZ and CZ can help DNP/SS to scale to thousands of nodes.

6.5.7 Conclusion

In the previous set of experiments, direct comparisons between DNP/SS and MDC/PEQ illustrated that the former outperforms the latter in various metrics including: End-to-End delay; packet delivery ratio; network lifetime; energy balancing; and network scalability. The experimental results prove that DNP/SS may have some features that raise its performance over MDC/PEQ. Firstly, the self-stabilisation algorithm considerably reduces the amount of communication and computation to converge to a legitimate state. Secondly, SZs localise topology updates within their perimeter. Thirdly, the optimal MDC selection algorithm reduces topology updates by selecting the optimal MDC that have the longest connection time.

6.6 DNP/SS in Real Mobility Scenarios

6.6.1 Military scenario implementation

This section implements and evaluates the military application scenario described in Section 3.2. In this set of experiments, a synthetic MDC mobility plan is designed and used. This mobility plan simulates the real-world movement of armoured vehicles and allows the testing of DNP/SS under various mobility conditions. DNP/SS will be evaluated with various mobility velocities including: static, low speed, and high speed. At the beginning of the simulation, MDCs are programmed to remain static for $5\ \mathrm{minutes}$. In the following $5\ \mathrm{minutes}$, the MDCs start moving with the velocity: $\mathrm{v} = 5\ \mathrm{m/s}$. Then, for the following $5\ \mathrm{minutes}$, the MDCs velocity is increased to reach $20\ \mathrm{m/s}$. The number of the MDCs is set to 10% of the network population based on the real capacity of an armoured vehicle. They are also deployed randomly and have $300\mathrm{m}$ wireless radio range. They move according to the bounded random mobility model (BRMM). To increase the reliability of the obtained results, we test the performance using today's hardware capabilities, namely, iMote2 specifications [115]. Overall, these parameters are suitable for today's battlefield surveillance applications.

6.6.2 Simulation Setup

Sensor nodes were deployed at random in a square region $1000m\ x\ 1000m$. The wireless radio range of sensor nodes is set to $75\ meters$. The transmission and reception power of a sensor node is set to $0.033\ J/s$. The number of data sources is set to 500, while they are chosen randomly from the sensory field. Each data source generates $2\ pkts/s$ during the entire simulation time period. The initial energy of sensor nodes is enough to complete the simulation. The sink node is positioned in the centre of the simulation area and has a wireless radio range of $300\,m$. Table 6-3 shows a summary of the simulation parameters and their respective values.

Table 6-3: DNP/SS simulation parameters

Parameter	Value
Number of nodes	1000
Simulation area	1000 m x 1000 m
Wireless radio range (SN)	75 m
Wireless radio range (MDC)	300 m
Number of source node	500
Source nodes data rate	2 pkts/s
Number of MDCs	50
MDC velocity	0, 5, 20 m/s
Bandwidth	250 Kbps
Data packet size	64 Byte
TX power dissipation	0.033 W
RX power dissipation	0.033 W

6.6.3 Experiment 1: End-to-End Delay

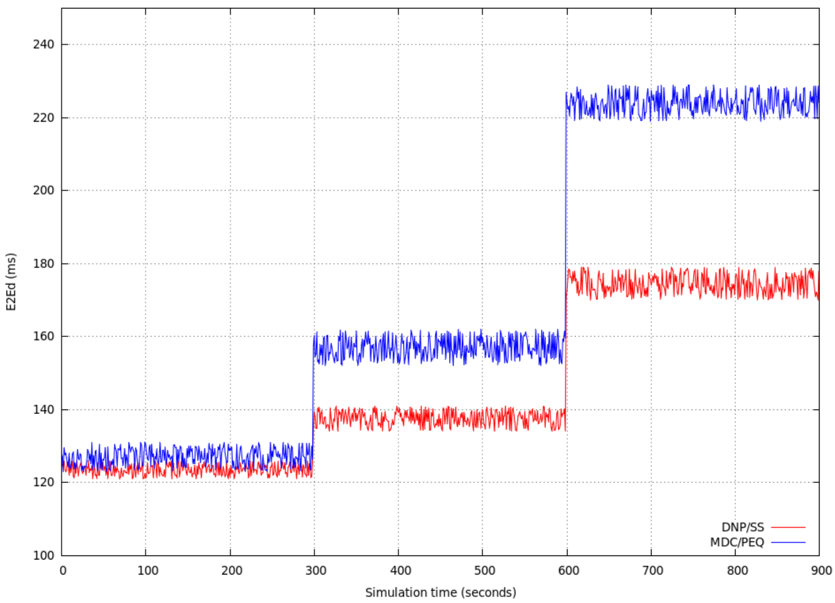


Figure 6.12: End-to-End delay at different mobility velocities

Aim:

The aim of this experiment is to study the end-to-end delay of MDC/PEQ and DNP/SS under different mobility speeds.

Results and discussion:

In this experiment, the mobility speed is increased twice, from $0\ \mathrm{m}/s$ to $5\ \mathrm{m}/s$ and from $5\ m/s$ to $20\ m/s$, at simulation times of $300\ \mathrm{seconds}, 600\ \mathrm{seconds}$, respectively. Figure 6.12 shows that increasing the mobility speed leads to a higher end-to-end delay in both algorithms. When the network topology is static, the End-to-End delay in MDC/PEQ is almost consistent with that for DNP/SS. In the absence of congestion due to mobility management traffic, data messages can be transmitted more quickly and with fewer errors. It can also be noticed that the end-to-end delay of both algorithms proportionally increases with higher mobility speed. As MDC mobility increases, topology reconfigurations become more frequent. However, increased mobility management overhead has a greater impact on MDC/PEQ's end-to-end delay than DNP/SS. DNP/SS is more immune to frequent network changes, since it incurs less topology management overhead and it converges quickly. DNP/SS generates less control packets for a number of reasons. Firstly, in DNP/SS, the topology update is localised due to the effect of SZs. Secondly, the optimal MDC selection algorithm allows nodes to connect to MDCs for longer periods of time; consequently, there are less interruptions to data communications. Thirdly, CZs make MDCs appear static for certain periods of time that are determined by MDC speed.

6.6.4 Experiment 2: Packet Delivery Ratio

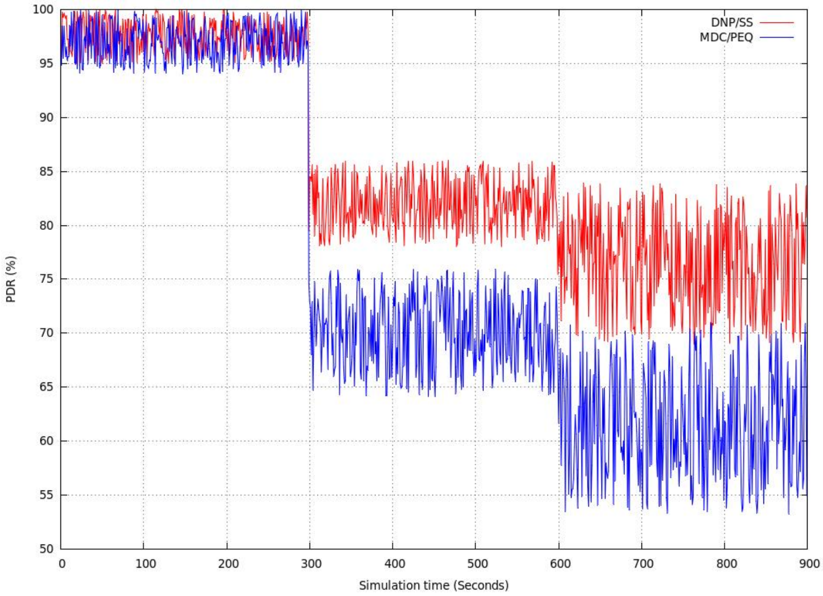


Figure 6.13: PDR at different mobility speeds

Aim:

This experiment studies the effects of varying the mobility speed on the performance of both algorithms in terms of PDR.

Results and discussion:

Figure 6.13 depicts the performance of both algorithms when increasing mobility speed. The results show that DNP/SS outperforms MDC/PEQ. As the velocity of mobile nodes increases, the PDR decreases. Increasing MDCs movement increased overall the frequency and amount of mobility management traffic; consequently, a higher number of packets will be dropped or will arrive late due to buffer overflows and congestion. Yet, DNP/SS performed better than MDC/PEQ as it isolates the topological updates and limits them to the SZ boundary. Moreover, in DNP/SS nodes are always connected to the MDC with the highest CET.

6.6.5 Experiment 3: Power Consumption

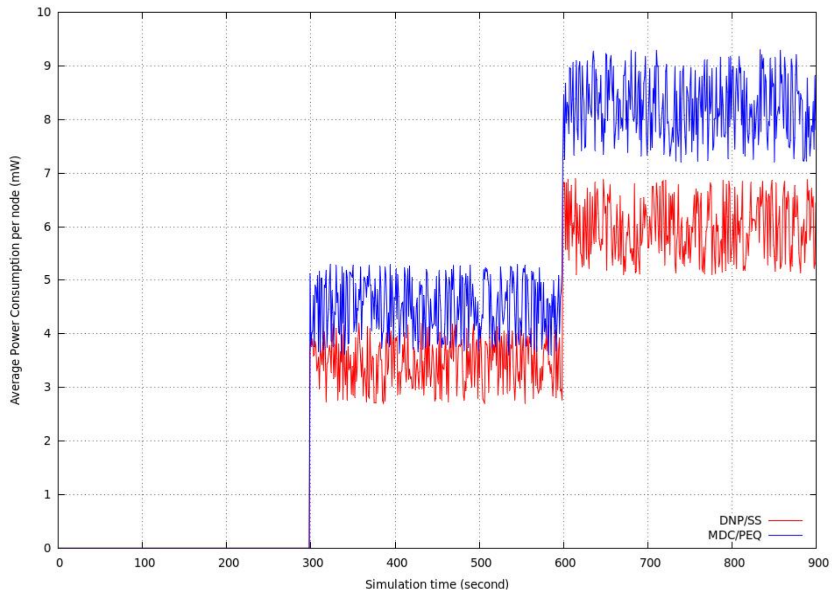


Figure 6.14: Power consumption at different mobility velocities

Aim:

The aim of this experiment is to measure the energy consumed by mobility management in both algorithms, and reveal the effect of MDC movement speed on the cost of managing mobility.

Results and discussion:

This experiment is concerned with measuring the topology maintenance cost. As depicted in Figure 6.14, When all nodes are static, the mobility management cost is zero. As mobility speed is increased, the cost to maintain the topology becomes higher due to the increasing frequency of link updates. It can be observed that DNP/SS consumes less energy than MDC/PEQ at all times. The main factor accounting for this result is that DNP/SS predicts when a SZ update is necessary; until that moment all routes update are performed locally within the SZ. Additionally, DNP/SS uses less control messages to reconfigure the network, since sensor nodes consider the MDC static until it moves out of its CZ, whereas in MDC/PEQ, sensor nodes receive and process beacons continuously.

6.6.6 Conclusion

This set of simulation experiments demonstrates the behavioural adaptability of DNP/SS and MDC/PEQ over the mobility speed. Increasing the mobility speed leads to more frequent topology changes, which in turn produces link reconfigurations. In order to reconfigure the network and maintain its mobility, more mobility management messages will be broadcasted, and thus the number of collisions and retransmission increases. This results in less utilisation of network's resources such as bandwidth and battery power. The experiments show that DNP/SS outperforms MDC/PEQ in terms of end-to-end delay, packet delivery ratio, and energy efficiency. This proves that DNP/SS reduces network reconfiguration and generates less control packets to update the topology.

6.7 Static-oriented IE over DNP/SS

This set of experiments evaluates the adaptivity of heterogeneous environments maintained by DNP/SS to the static-oriented IE approaches. The IE approach presented in [48] is chosen and applied to get an accurate picture about the performance of static-oriented IE approaches on top of DNP/SS. We apply [48] over a hybrid environment using DNP/SS and compare the results obtained from this set of experiments to the ones published by the authors. Subsection 2.8.2 in Chapter 2 presents a summary of [48].

6.7.1 Simulation Setup

In this experiment, the performance of [48] is evaluated in two settings. First, it is applied over a static network, as published by its authors. Second, it is applied over a heterogeneous network using DNP/SS. For the static environment, the simulation is performed on a network of 1000 sensor nodes that are deployed randomly in a square region of $150\mathrm{m} \times 150\mathrm{m}$. A fixed data sink is located in the centre of the square region. It is assumed that all sensor nodes will not change their position. For the heterogeneous environment, we use the same simulation parameters presented in Table 6-3. The hybrid IE approach is used to gather location information about a moving attack (e.g. moving tanks). The detection of such an attack is based on the magnitude of seismic vibration. The attack starts from the border of the monitored region with a randomly selected direction of movement. The duration of the attack is asserted to be 50 seconds. Each simulation run lasts for $100\ seconds$. Regarding the configurable parameters, the threshold value is set to $3\ Hz$, and TTL is set to 4.

6.7.2 Experiment 1: The Ratio of Data Dissemination Nodes-R_{nodes}

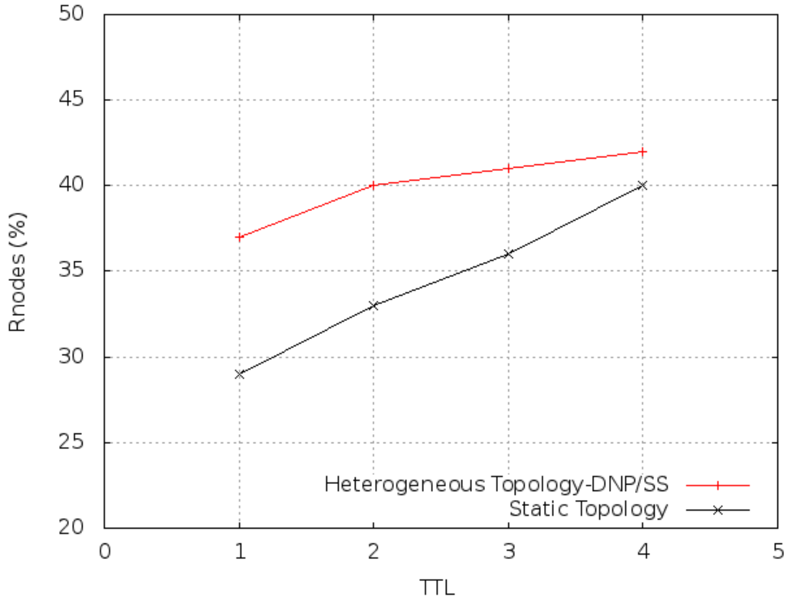


Figure 6.15: The ratio of data dissemination nodes

Aim:

This experiment aims to measure the ratio of nodes that are involved in detecting a certain event during a specified time interval. The experiment was repeated four times, varying the value of TTL from 1 to 4.

Results and discussion:

Figure 6.15 shows that the number of nodes engaged in data dissemination is higher when using DNP/SS. It can be observed that the number of nodes has a dramatic increase at$\mathrm{TTL} = 1$, and a slight increase at $\mathrm{TTL} = 4$. This is primarily due to the fact that nodes continuously approaching to or moving away from the event location. However, since the MDC and sensor nodes follow the event of interest as a group, this means that the self-stabilisation will work effectively, resulting in a virtually static network. At the same time, sensor nodes are focused in the vicinity of the event of interest, which results in accurate analysis of the event but more nodes within the defined hop level. The increase in the number of nodes when using NDP/SS may be due to the unlikely case of detecting the moving object by nodes located in separate SZs.

6.7.3 Experiment 2: Amount of Data Transmissions-N_{comm}

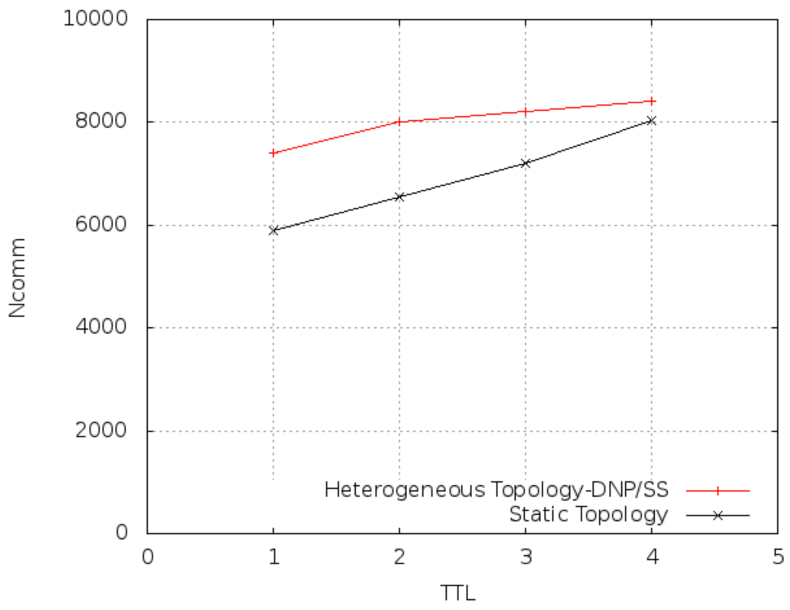


Figure 6.16: Amount of data transmissions

Aim:

The aim of this experiment is to evaluate the amount of data transmissions in order to extract information about the moving object. The experiment was repeated four times varying the value of TTL from 1 to 4.

Results and discussion:

The results shown in Figure 6.16 are consistent with the ones from the previous experiment. The total number of transmitted data messages increases proportionally to the number of nodes participating in the event detection. Since the density of senor nodes is high in the areas closer to the MDC, the increase of data messages is higher with a lower number of hops. Moreover, the presence of the MDC is in very close proximity to the sensor nodes and the event helps to reduce the number of communicated messages.

6.7.4 Experiment 3: End-to-End Delay

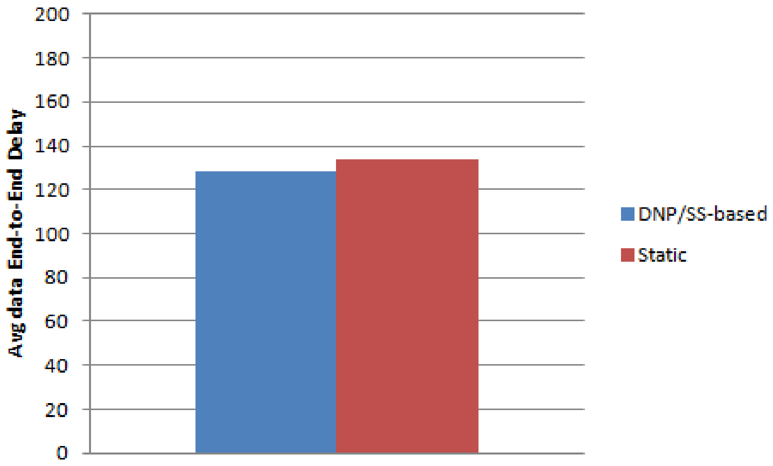


Figure 6.17: Information timeliness

Aim:

This experiment compares the average time needed for a single data message to be conveyed from its source to the sink node.

Results and discussion:

Figure 6.17 depicts the performance of [48] in terms of the average end-to-end delay. It can be observed from the figure that both topologies have comparable end-to-end data delivery delay. The NDP/SS deployment offers shorter latency time as the MDC moves closer to the event of interest. This reduces the bridging distance between the data sources and the data sink, which in turn reduces the buffering, processing and queuing delays. This compensates the time elapsed in network maintaining due to the nodes mobility. Furthermore, the SZs partitioning reduces the data search zone for nodes carrying significant data. This offers a faster nodes selection via the IE approach.

6.7.5 Experiment 3: Energy Consumption

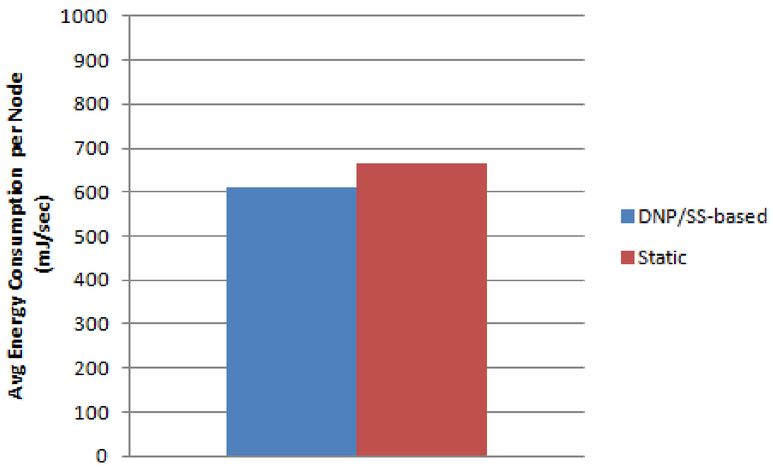


Figure 6.18: Information extraction cost

Aim:

The aim of this experiment is to evaluate the average energy consumed by a sensor node to extract information about the moving object. In this experiment we only consider the energy cost involved in extracting the required information; this excludes any costs related to mobility management or other network tasks.

Results and discussion:

To calculate the cost of IE about the moving object, we measure the average energy consumed only by sensor nodes engaged in the data gathering or processing in relation to this event. Figure 6.18 shows that the energy consumption in both settings remains comparable. This is mainly due to the following factors: First, in DNP/SS, MDCs move towards the event of interest, which reduces the bridging distance between source nodes and the sink. Second, the grouping of nodes in logical structures (SZ's) aids in limiting the scope IE towards areas of data significance.

6.7.6 Conclusion

The experimental results demonstrate that the DNP/SS can successfully allow the deployment of static-oriented IE approaches in hybrid networks. In some aspects, DNP/SS has introduced little extra cost to the system. However, DNP/SS accepted the static-oriented approach without any modifications or adaptations. Moreover, DNP/SS has passed some of the benefits of the underlying mobile network to the IE layer; thus, improving the QoI extracted using the static-oriented method.

Chapter 7

Conclusion

The main purpose of a WSN is to provide users with access to the information from data collected by spatially distributed sensors. IE is a practical multistage process at which end user's applications operate using a structured methodology to discover and extract the information enfolded in data. A literature review summarised research related to IE from mWSNs in Chapter 2. The following topics were investigated: the process of IE, static-oriented IE and its classifications, mobile-oriented IE and its classifications, mobility benefits, mobility challenges, and the attributes of mWSNs information. The main findings from the literature review were that the most important design challenges for IE from mWSNs are energy efficiency and the QoI set by the system users.

Nodes mobility introduces new challenges to IE including intermittent connectivity, data latency, and high power consumption imposed by frequent topology reconfiguration. Existing mobile-oriented IE approaches dealt with individual mobility related problems as the need arises. These approaches have many inherent problems that makes them inefficient outside the application domain they are designed for. On the other hand, existing static-oriented IE approaches have been reviewed and proved to be effective and efficient.

This thesis addressed the issues of efficient IE from mWSNs. Our research efforts focused on improving the efficiency of mWSNs, by improving the quality of the returned information while keeping the energy consumption to the minimum. We demonstrated that static-oriented IE approaches can be applied successfully to extract information from hybrid environments. This is achieved by the development of mobility maintenance techniques to adapt static-oriented IE approaches to mobile and hybrid networks, by making mobility transparent to the IE layer of the WSN protocol stack.

To enable seamless application of static-oriented IE approaches to work with mobile nodes, a self stabilising algorithm, called SS, has been presented. The proposed algorithm stabilises nodes mobility in the bottom layer of the network model. Using self-stabilisation, some approximation of fixed topology is established to recover from an arbitrary transient failures rapidly and at low cost. Thus, the proposed self-stabilisation algorithm provides autonomous adaptability against topology changes in a transparent manner to both upper network layers and to user applications. Simulation results show that the presented approach is an efficient and effective method for stabilising nodes movements in mWSNs.

Motivated by increasing the extracted QoI in mWSN, a dynamic network partitioning protocol, called DNP, is proposed in Chapter 5. Network partitioning is a well-tested solution to achieve scalability and load balancing. Grouping nodes into smaller logical sets makes buffer overflows and energy depletion less of a problem. DNP manages MDCs mobility at the middle layer of the network model, and results in a set of well delimited network partitions of sensor nodes that are updated dynamically. Experimental evaluation showed that DNP reduces mobility management cost and increases the QoI.

To improve the stability of the network and reduce topology reconfiguration frequency, an optimal MDC selection algorithm was developed. The choice of an MDC has direct impact on the frequency of network reconfiguration, which significantly impact the energy conservation, load balancing, and QoI. Most of previous solutions calculate the best MDC based on factors such as: distance between the MDC and the selecting node, the signal strength of the MDC antenna, or transmission latency. However, the proposed MDC selection algorithm is based on the principle that the sensor node will be able to predict the future disconnection time. Accordingly, the optimal MDC for a sensor node is the closest MDC that would remain connected for longer time.

We have integrated the SS algorithm and DNP protocol into DNP/SS protocol. Simulation results presented in Chapter 5 demonstrated that the DNP/SS protocol outperforms its rivals in the literature in terms of timeliness, packet delivery ratio, network scalability, network lifetime, and energy consumption. Furthermore, it proved that DNP/SS successfully allows the deployment of static-oriented IE approaches in hybrid networks without any modifications or adaptations.

The mobility maintenance techniques proposed in this thesis give strong basis for a number of interesting directions for future work, which will lead to improved QoI. This work could be extended to other, related areas including:

1. The CZs have the potential to be utilised by IE approaches that are based on logical grouping of nodes to deliver their intended functionality, e.g. for query scoping or dissemination.
2. To improve the handover interruption time (i.e., the time between disconnecting from the current SZ and connecting to a new one). This can be done by integrating a precise prediction algorithm to predict when the SZ needs update. Informing nodes that will be affected with the update process before the time of the update is due. This gives nodes time to proactively execute the optimal MDC selection algorithm. Consequently, handoffs would be performed more rapidly.
3. We feel it is essential to explore the benefits this solution can bring to other areas of WSNs, e.g. fault tolerance.
4. Finally, it will be interesting to test more static-oriented IE approaches over DNP/SS and study their behaviour and effectiveness.

Bibliography

[1] M. Hammoudeh*, et al.*, "An Approach to Data Extraction and Visualisation for Wireless Sensor Networks," in *Networks, 2009. ICN '09. Eighth International Conference on*, 2009, pp. 156-161.

[2] C. Curtis*, et al.*, "The layered sensing operations center: a modeling and simulation approach to developing complex ISR networks," 2010, pp. 769415-769415-7.

[3] J. H. McQuiddy, "A roadmap for future UGS," 2010, pp. 76940N-76940N-12.

[4] B. Coltin and M. Veloso, "Mobile robot task allocation in hybrid wireless sensor networks," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2010, pp. 2932-2937.

[5] B. Ren*, et al.*, "The hybrid mobile wireless sensor networks for data gathering," presented at the Proceedings of the 2006 international conference on Wireless communications and mobile computing, Vancouver, British Columbia, Canada, 2006.

[6] X. Yang and Z. Yanping, "Surveillance and Tracking System with Collaboration of Robots, Sensor Nodes, and RFID Tags," in *Computer Communications and Networks, 2009. ICCCN 2009. Proceedings of 18th Internatonal Conference on*, 2009, pp. 1-6.

[7] A. T. Erman*, et al.*, "Enabling mobility in heterogeneous wireless sensor networks cooperating with UAVs for mission-critical management," *Wireless Communications, IEEE,* vol. 15, pp. 38-46, 2008.

[8] S. A. Munir*, et al.*, "Mobile Wireless Sensor Network: Architecture and Enabling Technologies for Ubiquitous Computing," in *Advanced Information Networking and Applications Workshops, 2007, AINAW '07. 21st International Conference on*, 2007, pp. 113-120.

[9] S. Sarawagi, "Information Extraction," *Found. Trends databases,* vol. 1, pp. 261-377, March 2008.

[10] L. Tzevelekas and I. Stavrakakis, "Sink mobility schemes for data extraction in large scale WSNs under single or zero hop data forwarding," *Wireless Conference (EW), 2010 European,* pp. 896 -902, april 2010.

[11] G. Pantziou*, et al.*, "Mobile Sinks for Information Retrieval from Cluster-Based WSN Islands," *Proceedings of the 8th International Conference on Ad-Hoc, Mobile and Wireless Networks,* pp. 213-226, 2009.

[12] R. Sugihara and R. K. Gupta, "Optimal Speed Control of Mobile Node for Data Collection in Sensor Networks," *Mobile Computing, IEEE Transactions on,* vol. 9, pp. 127 -139, jan. 2010.

[13] Y.-C. Tseng*, et al.*, "Opportunistic data collection for disconnected wireless sensor networks by mobile mules," *Ad Hoc Networks,* vol. 11, pp. 1150-1164, 2013.

[14] S. Chowdhury and C. Giri, "Data collection point based mobile data gathering scheme with relay hop constraint," in *Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on*, 2013, pp. 282-287.

[15] A. Sangeetha*, et al.*, "Energy efficient data gathering via mobile element using adaptive clustering," in *Automation, Computing, Communication, Control and Compressed Sensing (iMac4s), 2013 International Multi-Conference on*, 2013, pp. 596-601.

[16] M. Hammoudeh*, et al.*, "An Approach to Data Extraction and Visualisation for Wireless Sensor Networks," *Proceedings of the 2009 Eighth International Conference on Networks,* pp. 156-161, 2009.

[17] H. Mohammad and A. A. A. Tariq, "Building programming abstractions for wireless sensor networks using watershed segmentation," presented at the Proceedings of the 11th international conference and 4th international conference on Smart spaces and next generation wired/wireless networking, St. Petersburg, Russia.

[18] M. Luca and P. Gian Pietro, "Programming wireless sensor networks with logical neighborhoods," presented at the Proceedings of the first international conference on Integrated internet ad hoc and sensor networks, Nice, France, 2006.

[19] V. Gupta and R. Pandey, "Data fusion and topology control in wireless sensor networks," *Proceedings of the 5th conference on Applied electromagnetics, wireless and optical communications,* pp. 135-140, 2007.

[20] J. Llinas and D. L. Hall, "An introduction to multi-sensor data fusion," *Circuits and Systems, 1998. ISCAS '98. Proceedings of the 1998 IEEE International Symposium on,* vol. 6, pp. 537 -540 vol.6, may-3 jun 1998.

[21] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," *Decision and Control, 2007 46th IEEE Conference on,* pp. 5492 -5498, dec. 2007.

[22] J. Yue*, et al.*, "A Novel Cluster-Based Data Fusion Algorithm for Wireless Sensor Networks," *Wireless Communications, Networking and Mobile Computing (WiCOM), 2011 7th International Conference on,* pp. 1 -5, sept. 2011.

[23] L. Krishnamachari*, et al.*, "The impact of data aggregation in wireless sensor networks," *Distributed Computing Systems Workshops, 2002. Proceedings. 22nd International Conference on,* pp. 575 - 578, 2002.

[24] E. Fasolo*, et al.*, "In-network aggregation techniques for wireless sensor networks: a survey," *Wireless Communications, IEEE,* vol. 14, pp. 70 -87, april 2007.

[25] K. Padmanabh and S. K. Vuppala, "An Adaptive Data Aggregation Algorithm in Wireless Sensor Network with Bursty Source.," *Wireless Sensor Network,* pp. 222-232, 2009.

[26] S. Munir*, et al.*, "Mobile Wireless Sensor Networks: Architects for Pervasive Computing," *InTech,* 2011.

[27] M. M. Gaber*, et al.*, "An analytical study of central and in-network data processing for wireless sensor networks," *Inf. Process. Lett.,* vol. 110, pp. 62-70, December 2009.

[28] C. Zhang*, et al.*, "Unspecific Event Detection in Wireless Sensor Networks," *Communication Software and Networks, International Conference on,* vol. 0, pp. 243-246, 2009.

[29] R. Bhargavi*, et al.*, "Complex Event Processing for Object Tracking in Wireless Sensor Networks," *Web Intelligence and Intelligent Agent Technology (WI-*

IAT), 2010 IEEE/WIC/ACM International Conference on, vol. 3, pp. 211 -214, 31 2010-sept. 3 2010.

[30] B. Gedik*, et al.*, "ASAP: An Adaptive Sampling Approach to Data Collection in Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems,* vol. 18, pp. 1766-1783, 2007.

[31] M. L. Segal*, et al.*, "Method and apparatus for automatic event detection in a wireless communication system," September 2000.

[32] G. Werner-Allen*, et al.*, "Deploying a Wireless Sensor Network on an Active Volcano," *IEEE Internet Computing,* vol. 10, pp. 18-25, 2006.

[33] M. Bahrepour*, et al.*, "Distributed Event Detection in Wireless Sensor Networks for Disaster Management " *Proceedings of the International Conference on Intelligent Networking and Collaborative Systems, INCoS 2010,* pp. 507-512, November 2010.

[34] B. Abid*, et al.*, "An Event-Driven Clustering Scheme for Data Aggregation in Real-Time Wireless Sensor Networks," in *Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference on*, 2013, pp. 48-55.

[35] M. Andelic*, et al.*, "Collaborative Event Driven Energy Efficient Protocol (CEDEEP)," *Wireless Communications Letters, IEEE,* vol. 2, pp. 231-234, 2013.

[36] D. Chu*, et al.*, "Approximate Data Collection in Sensor Networks using Probabilistic Models," *Proceedings of the 22nd International Conference on Data Engineering,* pp. 48-, 2006.

[37] Y. W. Guorui Li, "An Efficient Data Aggregation Scheme Leveraging Time Series Prediction in Wireless Sensor Networks," *International Journal of Machine Learning and Computing,* vol. 1, pp. 372-377, oct 2011.

[38] G.-S. Ahn, "Information-driven Tracking and Access Control in Wireless Ad hoc and Sensor Networks," Columbia University, 2009.

[39] S. Bhattacharjee, "Distributed Algorithm for Dynamic Data-Gathering in Sensor Network," *ArXiv e-prints,* October 2013.

[40] X. Kui*, et al.*, "Constructing a CDS-Based Network Backbone for Data Collection in Wireless Sensor Networks," *International Journal of Distributed Sensor Networks,* vol. 2013, p. 12, 2013.

[41] G. Trajcevski and P. Scheuermann, "(Reactive /+ Proactive Behavior) Situation Awareness in Sensor Networks," *Workshop on Research Directions in Situational-aware Self-managed Proactive Computing in Wireless Adhoc Networks,* 2009.

[42] Y. Yao and J. Gehrke, "The cougar approach to in-network query processing in sensor networks," *SIGMOD Rec.,* vol. 31, pp. 9-18, 2002.

[43] M. Bestehorn*, et al.*, "Energy-efficient processing of spatio-temporal queries in wireless sensor networks," *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems,* pp. 340-349, 2010.

[44] S. R. Madden*, et al.*, "TinyDB: an acquisitional query processing system for sensor networks," *ACM Trans. Database Syst.,* vol. 30, pp. 122-173, 2005.

[45] D.-W. Choi and C.-W. Chung, "REQUEST+: A framework for efficient processing of region-based queries in sensor networks," *Information Sciences,* vol. 248, pp. 151-167, 2013.

[46] B.-D. Lee, "Adaptive Data Dissemination Protocol for Wireless Sensor Networks," in *Security-Enriched Urban Computing and Smart Grid* vol. 78, ed, 2010, pp. 188-195.

[47] C.-H. Lee*, et al.*, "Effective processing of continuous group-by aggregate queries in sensor networks," *J. Syst. Softw.,* vol. 83, pp. 2627-2641, December 2010.

[48] L. Byoung-Dai and L. Kwang-Ho, "An Energy-Efficient Hybrid Data-Gathering Protocol Based on the Dynamic Switching of Reporting Schemes in Wireless Sensor Networks," *Systems Journal, IEEE,* vol. 6, pp. 378-387, 2012.

[49] V. Coskun, "Relocating Sensor Nodes to Maximize Cumulative Connected Coverage in Wireless Sensor Networks," *Sensors,* vol. 8, pp. 2792-2817, 2008.

[50] G. Wang*, et al.*, "Movement-assisted sensor deployment," *Mobile Computing, IEEE Transactions on,* vol. 5, pp. 640 -652, june 2006.

[51] P. Guo*, et al.*, "An Adaptive Coverage Algorithm for Large-Scale Mobile Sensor Networks," in *Ubiquitous Intelligence and Computing* vol. 4159, ed: Springer Berlin / Heidelberg, 2006, pp. 468-477.

[52] N. Gajaweera and D. Dias, "FAMA/TDMA Hybrid MAC for Wireless Sensor Networks," *Information and Automation for Sustainability, 2008. ICIAFS 2008. 4th International Conference on,* pp. 67 -72, dec. 2008.

[53] S. Sivaramakrishnan and A. Al-Anbuky, "Analysis of network connectivity: Wildlife and Sensor Network," *Telecommunication Networks and Applications Conference (ATNAC), 2009 Australasian,* pp. 1 -6, nov. 2009.

[54] T. Wark*, et al.*, "Transforming Agriculture through Pervasive Wireless Sensor Networks," *Pervasive Computing, IEEE,* vol. 6, pp. 50 -57, april-june 2007.

[55] S. Jain*, et al.*, "Exploiting mobility for energy efficient data collection in wireless sensor networks," *Mob. Netw. Appl.,* vol. 11, pp. 327-339, June 2006.

[56] R. C. Shah*, et al.*, "Data MULEs: modeling a three-tier architecture for sparse sensor networks," *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on,* pp. 30 -41, may 2003.

[57] D. N. Jayasingha*, et al.*, "Wireless Sensor Network Testbed for Mobile Data Communication," *Information and Automation for Sustainability, 2008. ICIAFS 2008. 4th International Conference on,* pp. 97 -103, dec. 2008.

[58] G. Anastasi*, et al.*, "Data collection in sensor networks with data mules: An integrated simulation analysis," *Computers and Communications, 2008. ISCC 2008. IEEE Symposium on,* pp. 1096 -1102, july 2008.

[59] A. Chakrabarti*, et al.*, "Communication power optimization in a sensor network with a path-constrained mobile observer," *ACM Trans. Sen. Netw.,* vol. 2, pp. 297-324, August 2006.

[60] W. Alsalih*, et al.*, "Routing to a Mobile Data Collector on a Predefined Trajectory," *Communications, 2009. ICC '09. IEEE International Conference on,* pp. 1 -5, june 2009.

[61] D. Jea*, et al.*, "Multiple controlled mobile elements (data mules) for data collection in sensor networks," *In DCOSS,* pp. 244-257, 2005.

[62] M. Xi*, et al.*, "Run to Potential: Sweep Coverage in Wireless Sensor Networks," *Parallel Processing, 2009. ICPP '09. International Conference on,* pp. 50 -57, sept. 2009.

[63] L. He, *et al.*, "Evaluating On-Demand Data Collection with Mobile Elements in Wireless Sensor Networks," *Vehicular Technology Conference Fall (VTC 2010-Fall), 2010 IEEE 72nd,* pp. 1 -5, sept. 2010.

[64] S. Gao, *et al.*, "Efficient data collection in wireless sensor networks with path-constrained mobile sinks," *World of Wireless, Mobile and Multimedia Networks Workshops, 2009. WoWMoM 2009. IEEE International Symposium on a,* pp. 1 -9, june 2009.

[65] L. Song and D. Hatzinakos, "Architecture of Wireless Sensor Networks With Mobile Sinks: Sparsely Deployed Sensors," *Vehicular Technology, IEEE Transactions on,* vol. 56, pp. 1826 -1836, july 2007.

[66] T.-S. Chen, *et al.*, "Geographic convergecast using mobile sink in wireless sensor networks," *Computer Communications,* vol. 36, pp. 445-458, 2013.

[67] H. Lee, *et al.*, "Data stashing: energy-efficient information delivery to mobile sinks through trajectory prediction," *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks,* pp. 291-302, 2010.

[68] G. Xing, *et al.*, "Rendezvous design algorithms for wireless sensor networks with a mobile base station," *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing,* pp. 231-240, 2008.

[69] H. Sabbineni and K. Chakrabarty, "Datacollection in Event-Driven Wireless Sensor Networks with Mobile Sinks," *International Journal of Distributed Sensor Networks,* vol. 2010.

[70] I. Chatzigiannakis, *et al.*, "Fast and energy efficient sensor data collection by multiple mobile sinks," *Proceedings of the 5th ACM international workshop on Mobility management and wireless access,* pp. 25-32, 2007.

[71] A. Viana, *et al.*, "Decoupling data dissemination from mobile sink's trajectory in wireless sensor networks," *Communications Letters, IEEE,* vol. 13, pp. 178 -180, march 2009.

[72] M. Grossglauser and D. N. C. Tse, "Mobility increases the capacity of ad hoc wireless networks," *Networking, IEEE/ACM Transactions on,* vol. 10, pp. 477 -486, aug 2002.

[73] C. Zhu*, et al.*, "Research issues on mobile sensor networks," *Communications and Networking in China (CHINACOM), 2010 5th International ICST Conference on,* pp. 1 -6, aug. 2010.

[74] J. Hightower and G. Borriello, "Location systems for ubiquitous computing," *Computer,* vol. 34, pp. 57 -66, aug 2001.

[75] A. Kansal*, et al.*, "Controlled mobility for sustainable wireless sensor networks," *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on,* pp. 1 - 6, oct. 2004.

[76] P. Guo*, et al.*, "An adaptive coverage algorithm for large-scale mobile sensor networks," *Proceedings of the Third international conference on Ubiquitous Intelligence and Computing,* pp. 468-477, 2006.

[77] G. Dini*, et al.*, "An algorithm for reconnecting wireless sensor network partitions," *Proceedings of the 5th European conference on Wireless sensor networks,* pp. 253-267, 2008.

[78] F. Ishmanov*, et al.*, "Energy consumption balancing in Wireless Sensor Networks," *Computer Applications and Industrial Electronics (ICCAIE), 2010 International Conference on,* pp. 322 -327, dec. 2010.

[79] Y. Bi*, et al.*, "Moving Schemes for Mobile Sinks in Wireless Sensor Networks," *Performance, Computing, and Communications Conference, 2007. IPCCC 2007. IEEE Internationa,* pp. 101 -108, april 2007.

[80] H. Lee*, et al.*, "Data stashing: energy-efficient information delivery to mobile sinks through trajectory prediction.," *IPSN'10,* pp. 291-302, 2010.

[81] A. Kinalis*, et al.*, "Biased Sink Mobility with Adaptive Stop Times for Low Latency Data Collection in Sensor Networks," *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE,* pp. 1 -6, 30 2009-dec. 4 2009.

[82] Y. Gu*, et al.*, "Partitioning based mobile element scheduling in wireless sensor networks," *Sensor and Ad Hoc Communications and Networks, 2005. IEEE SECON 2005. 2005 Second Annual IEEE Communications Society Conference on,* pp. 386 - 395, sept., 2005.

[83] L. Jun*, et al.*, "MobiRoute: routing towards a mobile sink for improving lifetime in sensor networks," presented at the Proceedings of the Second

IEEE international conference on Distributed Computing in Sensor Systems, San Francisco, CA, 2006.

[84] Y. Yang*, et al.*, "Improving network lifetime with mobile wireless sensor networks," *Comput. Commun.,* vol. 33, pp. 409-419, March 2010.

[85] J.Naskath*, et al.*, "Article: Coverage Maintenance using Mobile Nodes in Clustered Wireless Sensor Networks," *International Journal of Computer Applications,* vol. 21, pp. 6-12, May 2011.

[86] R. Sugihara and R. K. Gupta, "Speed control and scheduling of data mules in sensor networks," *ACM Trans. Sen. Netw.,* vol. 7, pp. 4:1-4:29, August 2010.

[87] A. A. Somasundara*, et al.*, "Controllably mobile infrastructure for low energy embedded networks," *Mobile Computing, IEEE Transactions on,* vol. 5, pp. 958 -973, aug. 2006.

[88] V. Sachidananda*, et al.*, "Quality of Information in Wireless Sensor Networks: A Survey," *ICIQ,* 2010.

[89] C. Bisdikian, "On Sensor Sampling and Quality of Information: A Starting Point," *Pervasive Computing and Communications Workshops, 2007. PerCom Workshops '07. Fifth Annual IEEE International Conference on,* pp. 279 -284, march 2007.

[90] M. Ismat*, et al.*, "Cautious rating for trust-enabled routing in wireless sensor networks," *EURASIP J. Wirel. Commun. Netw.,* vol. 2009, pp. 1-16, 2009.

[91] R. Hoes*, et al.*, "Analysing qos trade-offs in wireless sensor networks," *Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems,* pp. 60-69, 2007.

[92] J. Luo*, et al.*, "Efficient Data Delivery with Packet Cloning for Underwater Sensor Networks," *Underwater Technology and Workshop on Scientific Use of Submarine Cables and Related Technologies, 2007. Symposium on,* pp. 34 -41, april 2007.

[93] B. Mihajlovi´c, "Compression and Security Platform for the Testing of Wireless Sensor Network Nodes," PhD, Department of Electrical & Computer Engineering, McGill University, Montreal,, 2008.

[94] H. Green*, et al.*, "Calculating network availability," *Aerospace conference, 2009 IEEE,* pp. 1 -11, march 2009.

[95] R. W. N. Pazzi and A. Boukerche, "Mobile data collector strategy for delay-sensitive applications over wireless sensor networks," *Comput. Commun.,* vol. 31, pp. 1028-1039, 2008.

[96] T. Alsboui*, et al.*, "Information Extraction from Wireless Sensor Networks: System and Approaches," *Sensors & Transducers,,* vol. 14-2, pp. 1-17, 2012.

[97] Aggeliki Prayati*, et al.*, "A Simulation Framework for the Performance Evaluation of Localisation Techniques over WSNs," in *SENSORCOMM, The Fifth International Conference on Sensor Technologies and Applications*, French Riviera, Nice/Saint Laurent du Var, France, August 21-27, 2011, pp. 122-127.

[98] B. M. Blum*, et al.*, "IGF: A State-Free Robust Communication Protocol for Wireless Sensor Networks," 2003.

[99] T. Alsboui*, et al.*, "Information Extraction from Wireless Sensor Networks: System and Approaches," *Sensors & Transducers Journal,* vol. 14-2, pp. 1-17, March 2012.

[100] P. S. Sandhu*, et al.*, "A survey on Software Reusability," in *Mechanical and Electrical Technology (ICMET), 2010 2nd International Conference on*, 2010, pp. 769-773.

[101] M. Hammoudeh and T. A. Alsbou'i, "Building Programming Abstractions for Wireless Sensor Networks Using Watershed Segmentation," in *Smart Spaces and Next Generation Wired/Wireless Networking*. vol. 6869, S. Balandin*, et al.*, Eds., ed: Springer Berlin Heidelberg, 2011, pp. 587-597.

[102] N. Guellati and H. Kheddouci, "A survey on self-stabilizing algorithms for independence, domination, coloring, and matching in graphs," *J. Parallel Distrib. Comput.,* vol. 70, pp. 406-415, 2010.

[103] A. Larsson and P. Tsigas, "Self-stabilizing (k,r)-clustering in wireless ad-hoc networks with multiple paths," in *Proceedings of the 14th international conference on Principles of distributed systems*, ed. Tozeur, Tunisia: Springer-Verlag, 2010, pp. 79-82.

[104] D. S. J. De Couto*, et al.*, "A high-throughput path metric for multi-hop wireless routing," in *Proceedings of the 9th annual international conference on Mobile computing and networking*, ed, 2003, pp. 134-146.

[105] A. Munir and A. G. Ross, "An MDP-Based Dynamic Optimization Methodology for Wireless Sensor Networks," *Parallel and Distributed Systems, IEEE Transactions on,* vol. 23, pp. 616 -625, April 2012.

[106] Ha*, et al.*, "Quantifying Network Partitioning in Mobile Ad Hoc Networks," in *Mobile Data Management, 2007 International Conference on*, 2007, pp. 174-181.

[107] R. Jurdak*, et al.*, "Adaptive Low Power Listening for Wireless Sensor Networks," *Mobile Computing, IEEE Transactions on,* vol. 6, pp. 988-1004, 2007.

[108] M. Bahrepour*, et al.*, "Distributed Event Detection in Wireless Sensor Networks for Disaster Management," in *Intelligent Networking and Collaborative Systems (INCOS), 2010 2nd International Conference on*, 2010, pp. 507-512.

[109] O. Saukh*, et al.*, "Convex groups for self-organizing multi-sink wireless sensor networks," in *Industrial Electronics, 2009. IECON '09. 35th Annual Conference of IEEE*, 2009, pp. 2624-2629.

[110] A. Shahid*, et al.*, "Proactive multipath data dissemination for Multimedia Sensor Networks," in *Multitopic Conference (INMIC), 2012 15th International*, 2012, pp. 349-354.

[111] J. Niu*, et al.*, "R3E: Reliable Reactive Routing Enhancement for Wireless Sensor Networks," *Industrial Informatics, IEEE Transactions on,* vol. 10, pp. 784-794, 2014.

[112] C. Hsung-Pin and H. Shun-Chih, "A hybrid intelligent protocol in sink-oriented wireless sensor networks," in *Information Security and Intelligence Control (ISIC), 2012 International Conference on*, 2012, pp. 57-60.

[113] R. L. Graham, "An Efficient Algorithm for Determining the Convex Hull of a Finite Planar Set," *Information Processing Letters,* vol. 1, pp. 132-133, 1972.

[114] P. Jacquet*, et al.*, "Optimized link state routing protocol for ad hoc networks," in *Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International*, 2001, pp. 62-68.

[115] Intel, "Intel Mote 2," in *Engineering Platform Data Sheet* ed, 2006, pp. 1-9.

[116] G. Chen*, et al.*, "Sense: A sensor network simulator," in *Advances in Pervasive Computing & Networking*, ed, 2004, pp. 249-267.

[117] A. Abuarqoub*, et al.*, "Simulation Issues in Wireless Sensor Networks: A Survey," in *SENSORCOMM '12: Proceedings of the 2012 Sixth International Conference on Sensor Technologies and Applications*, Rome, Italy, 2012.

[118] NS-2. (2007, *The Network Simulator*.

[119] nsnam. (2011, Retrieved 18 April 2012). *NS-3*. Available: from http://www.nsnam.org/

[120] M. Caccamo*, et al.*, "An Implicit Prioritized Access Protocol for Wireless Sensor Networks," presented at the Proceedings of the 23rd IEEE Real-Time Systems Symposium, 2002.

[121] L. Jiajia*, et al.*, "End-to-end delay in mobile ad hoc networks with generalized transmission range and limited packet redundancy," in *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*, 2012, pp. 1731-1736.

[122] T. Ken*, et al.*, "Congestion controlled adaptive lightweight multicast in wireless mobile ad hoc networks," in *Computers and Communications, 2002. Proceedings. ISCC 2002. Seventh International Symposium on*, 2002, pp. 967-972.

[123] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Commun. ACM,* vol. 43, pp. 51-58, 2000.

[124] C. J. Fung and Y. E. Liu, "Lifetime Estimation of Large IEEE 802.15.4 Compliant Wireless Sensor Networks," in *Modeling, Analysis and Simulation of Computers and Telecommunication Systems, 2008. MASCOTS 2008. IEEE International Symposium on*, 2008, pp. 1-4.

[125] W. Qin*, et al.*, "A Realistic Power Consumption Model for Wireless Sensor Network Devices," in *Sensor and Ad Hoc Communications and Networks, 2006. SECON '06. 2006 3rd Annual IEEE Communications Society on*, 2006, pp. 286-295.

[126] H. Zhou*, et al.*, "Modeling of Node Energy Consumption for Wireless Sensor Networks," 2011.

[127] J. Broch*, et al.*, "A performance comparison of multi-hop wireless ad hoc network routing protocols," presented at the Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking, Dallas, Texas, USA, 1998.

[128] D. Johnson and D. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," in *Mobile Computing*. vol. 353, T. Imielinski and H. Korth, Eds., ed: Springer US, 1996, pp. 153-181.

[129] Jorge Nuevo and J.-C. Gre´goire, "Analysis of the effects of entity mobility models on ad hoc network communication," presented at the International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), Montreal Qc, Canada, 2003.

[130] W. R. Heinzelman*, et al.*, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," presented at the Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8 - Volume 8, 2000.