Energy Efficient Data Collection and Dissemination Protocols in Self-Organised Wireless Sensor Networks

Chibuzor Jerry Edordu

A dissertation submitted in partial fulfilment
of the requirements for the degree of
Doctor of Philosophy
of the
University College London.

Department of Electrical & Electronic Engineering
University College London

2010

I, Chibuzor Jerry Edordu, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Abstract

Wireless sensor networks (WSNs) are used for event detection and data collection in a plethora of environmental monitoring applications. However a critical factor limits the extension of WSNs into new application areas: energy constraints. This thesis develops self-organising energy efficient data collection and dissemination protocols in order to support WSNs in event detection and data collection and thus extend the use of sensor-based networks to many new application areas.

Firstly, a Dual Prediction and Probabilistic Scheduler (DPPS) is developed. DPPS uses a Dual Prediction Scheme combining compression and load balancing techniques in order to manage sensor usage more efficiently. DPPS was tested and evaluated through computer simulations and empirical experiments. Results showed that DPPS reduces energy consumption in WSNs by up to 35% while simultaneously maintaining data quality and satisfying a user specified accuracy constraint.

Secondly, an Adaptive Detection-driven Ad hoc Medium Access Control (ADAMAC) protocol is developed. ADAMAC limits the Data Forwarding Interruption problem which causes increased end-to-end delay and energy consumption in multi-hop sensor networks. ADAMAC uses early warning alarms to dynamically adapt the sensing intervals and communication periods of a sensor according to the likelihood of any new events occurring. Results demonstrated that compared to previous protocols such as SMAC, ADAMAC dramatically reduces end-to-end delay while still limiting energy consumption during data collection and dissemination.

The protocols developed in this thesis, DPPS and ADAMAC, effectively alleviate the energy constraints associated with WSNs and will support the extension of sensor-based networks to many more application areas than had hitherto been readily possible.

Acknowledgements

I would like to thank my primary supervisor, Dr. Yang Yang, 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. I am deeply grateful to the Communications and Information Systems Group of UCL, in particular to Dr. John Mitchell, Dr. Richard Clegg and Mussie Woldeselassie for providing useful feedback and positive criticisms which deepened the results of the study. Special thanks also go to Senceive Ltd., in particular, Michael Gois and Dr. Matthew Britton for generously allowing me not only to use their demonstration toolkit, but also for providing me with their FlatMesh Firmware. Many thanks are due to both Dr. Lam Ling Shum and Peter Stacey for kindly introducing me to valuable materials for the study. I am indebted to Professor Nina Thornhill, originally of UCL but lately of Imperial College, for providing the initial guidance into the subject of this thesis. I am grateful to the Head of the Communication and Information Systems Group, Professor Izzat Darwazeh, for providing invaluable encouragement and financial support. I would also like to acknowledge the kind assistance offered by Dr. Tony Kenyon during the concluding stages of my PhD and my wife, Helen, for her continuous encouragement. I would like to thank my family, especially my parents, for their selfless love and support. Above all, I thank God for giving me the perseverance required to ensure that the good work I started was completed.

List of Abbreviations

ADAMAC Adaptive Detection-driven Ad hoc Medium Access Control

ARIMA Auto Regressive Integrated Moving Average

BS Base Station

CBR Critical Breakdown Rate

CM Continuous Monitoring

DC Duty Cycle

DPM Dynamic Power Management

DPPS Dual Prediction and Probabilistic Scheduler

DPS Dual Prediction Scheme

DVS Dynamic Voltage Scaling

EEDC Energy Efficient Data Collection

eSENSE Energy Efficient Stochastic Sensing

EWMA Exponentially Weighted Moving Average

FA Fully Active

FIFO First In First Out

IMA Integrated Moving Average

KF Kalman Filtering

LMS Least Mean Square

MAC Medium Access Control

MSE Mean Square Error

RDI Regulated Deficit Irrigation

RF Radio frequency

SMAC Sleep Medium Access Control

TP Toggling Period

WSN Wireless Sensor Network

Contents

List of Figures

List of Tables

Chapter 1

Introduction

For several centuries, devices have been used to observe and measure important aspects of the earth in order to facilitate vital decision making. As long ago as 800BC, ancient Egyptians used nilometers to observe and monitor the depth of the River Nile. This was vital for their survival as it enabled them not only to assess the likelihood of flooding or drought, but also to anticipate agricultural yields [BA52]. Since the invention of the telescope at the beginning of the 17^{th} Century, scientists have been able to observe and monitor the position and movement of celestial bodies and gain further understanding of the universe [Kin94]. Over 300 years after the invention of the telescope, the first artificial satellite was launched into space. Since then satellite monitoring technology has been used to understand and monitor different aspects of the earth's environment. In 1971 the first geo-stationary satellite was deployed to monitor the earth's vegetation and minerals from outside the stratosphere [Wei72]. In recent years, as computing became ubiquitous, scientists started using computer-related devices to observe and measure vital phenomena such as natural disasters including floods and landslides [BM08].

It is important to monitor events because they can have serious implications on the lives of many, either through loss of life or damages to infrastructure and the environment.

For example, according to the Association of British Insurers, the problems created when extreme weather caused flooding in 2007 led to estimated losses of £3 billion in the UK [Pit08]. Flooding also caused large losses in previous years, not only in the UK but also in other parts of the world. To deal with this type of problem, computer-related devices could be used to monitor weather systems as well as different parameters, such as soil moisture content, over a given period of time in vulnerable regions. The resulting data could then be analysed and used to determine a threshold of imminent flooding. This could lead to the implementation of an early warning system, allowing immediate remedial action to be taken when necessary, thereby reducing loss of life or damage to property. The data collected could also be used to identify trends and predict future events. As will be further discussed and explored in the next section, Wireless Sensor Networks (WSNs) are uniquely suited to this kind of monitoring. Furthermore, WSNs can be usefully implemented for data collection and monitoring in a wide range of other areas including security surveillance applications [ASYS02], habitat monitoring [MPS+02, DFB+07] and agricultural management [BTB04b, BTB04a, TGL05].

1.1 The Monitoring of Phenomena: A problem of resource use and efficiency

The monitoring of vital phenomena has been conducted for centuries using different tools, and has been useful to many peoples, as illustrated in Figure 1.1. However, each monitoring tool, including nilometers, telescopes, satellites and semiconductor chip-based devices, although operational and useful to a certain degree, also has engineering limitations that ultimately impede its effectiveness. Satellites, for example, are advantageous because they can be used to collect data covering large areas measuring hundreds of square kilometres, but this comes at the expense of spatial resolution. Nilometers, though accurate, were labour intensive; nilometers were manually operated and required going to the river Nile every day to monitor and record water levels.

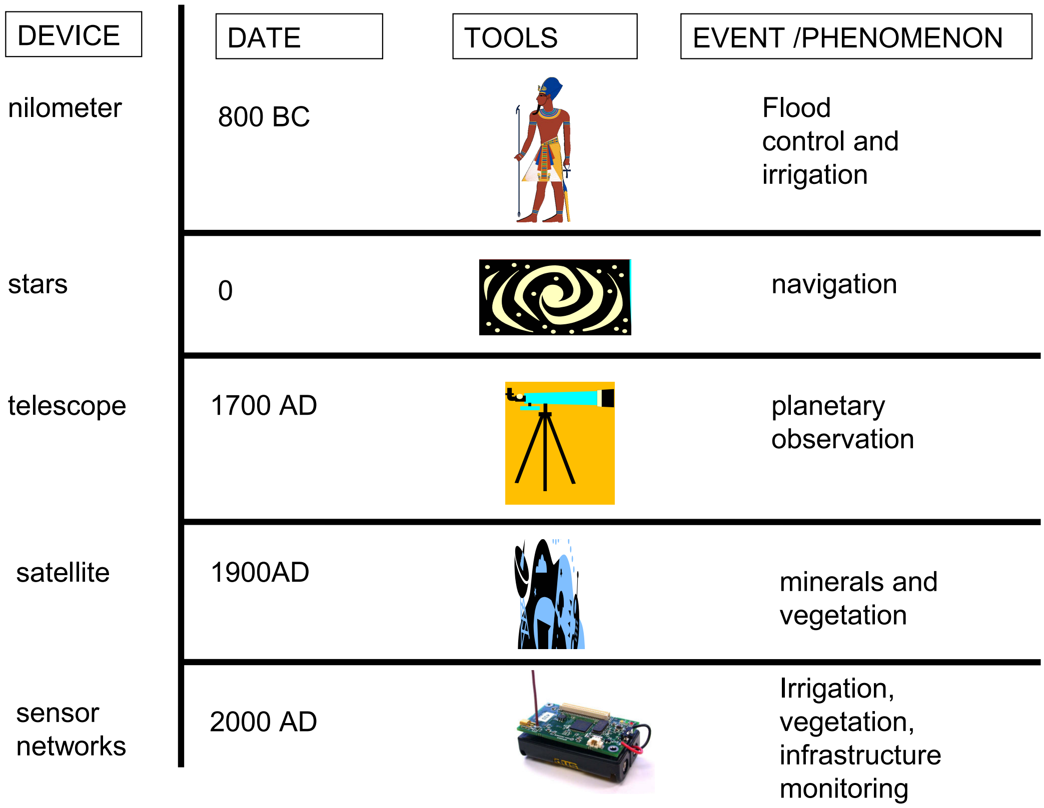


Figure 1.1: Event monitoring tools through the ages

Sensor networks, the most modern of the devices shown in Figure 1.1, are useful for monitoring a variety of environments. Sensor nodes in a sensor network comprise of three units: processing, communication and sensing units. Wirelessly interconnected systems of such nodes form a wireless sensor network. Sensor nodes within a WSN can exist in orders ranging from tens to thousands of devices to a user and are therefore uniquely different from other large scale systems such as the Internet.

As Figure 1.2 illustrates, when data on an event or a phenomenon is observed in a sensor field, sensor nodes send reports to a base station (BS) via multi-hopping. The BS is equipped with the capability for long-range communications and can therefore facilitate data access by a remote user.

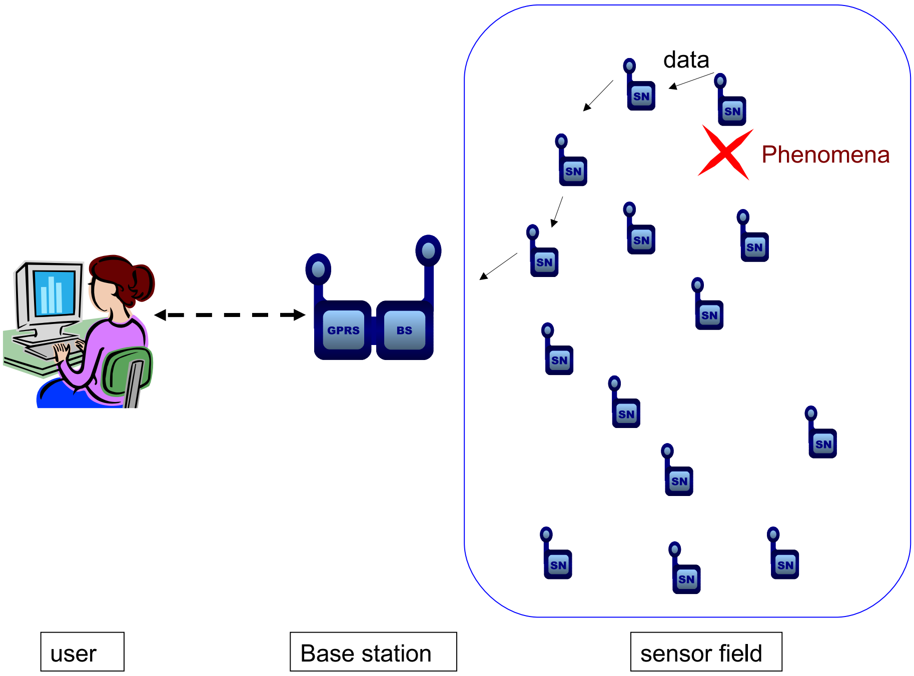


Figure 1.2: A wireless sensor network

As shown in Table 1.1, WSNs are currently the most favoured environmental monitoring tool, in comparison with other non WSN-based tools, for a variety of reasons. Non WSN-based tools are often not suitable for certain types of terrain, may be ineffective or impractical due to the physical characteristics of the device itself and are sometimes unusable due to the monetary costs involved in their installation. These non WSN-based tools can also be impractical and less reliable due to the logistics of deployment and can be difficult to install and complicated to use [TUML07, FW02]. For example, data loggers used in seismic exploration for oil consist of large geophone sensors which are powered from cables linked to a power supply [Hei00]. These sensors are extremely expensive to deploy [WZW06]. Other data loggers are impractical because they require lengthy calibration processes for set-up, or specialised knowledge during operation. Still others require large, heavy battery units which make them uneconomical as well as impractical [Wai07].

WSNs, on the other hand, are extremely small in size and require no wiring for data transport which means that they are easy to install in most locations and applications.

Table 1.1: The advantages and disadvantages of given monitoring instruments

Instrument	**Advantage**	**Disadvantage**
• Nilometer	• in-situ collection	• poor failure tolerance • no automation • calibration required • no remote monitoring • small spatial coverage
• Telescope	• useable for remote monitoring • large spatial coverage possible	• poor failure tolerance • no automation • calibration required • poor spatial granularity • relatively bulky
• Satellite	• useable for remote monitoring • large spatial coverage possible • autonomy possible	• poor failure tolerance • poor spatial granularity • relatively bulky
• WSN	• useable for remote monitoring • large spatial coverage possible • autonomy possible • in-situ collection	• relatively short lifetime

They are relatively inexpensive and yet reliable, with a high fault tolerance, because of the nature of distribution: with WSNs, multiple interconnected nodes are used, for example scattered over a field or distributed throughout a building, and each device has the capacity to collect data independently. The fact that the nodes are deployed in large numbers and that not all nodes need to be operational at the same time leads to increased tolerance of faults [ASYS02]. WSNs are also simple to use and extremely versatile because of their physical nature, thus rendering them usable for monitoring environments as contrasting as outbreaks of fire [Lad07] as well as glaciation [MOH04]. The characteristics of WSNs and non-WSNs devices are further contrasted in Table 1.2.

Table 1.2: Comparison of WSN and non-WSN based devices

Requirements	**WSNs**	**Non-WSN e.g. data loggers**
Application	One-to-many	One-to-one
Size	Small	Often bulky
Price	Relatively inexpensive	Can be expensive
Communication	Short-range	Long range
Architecture	Distributed	Centralised
Battery supply	Portable	Wired
Fault tolerance	High	Low
Energy restriction	Relatively high	Low

Despite all the obvious and aforementioned advantages which make WSNs potentially extremely effective for data collection and dissemination and superior to other alternative electronic monitoring devices, the widespread usability of WSNs is restricted by one engineering limitation: short lifetime. The critical resources required for operating a system are inherently limited. WSNs require an energy supply, usually in the form of batteries, the lives of which are often too short to fulfil application requirements [CES04]. While some efforts have been made to extend the lives of such batteries, the progress achieved has not been remarkable and would not make a notable difference to the functioning of such applications. (To date there has only been a modest improvement in the nominal capacity contained in Nickel-Cadium batteries, the most popular battery unit used over the last forty years [SCB96, Lim06].)

These energy constraints restrict the implementation of WSN monitoring operations in existing application areas such as environmental monitoring. Though expensive cabling is not required for WSNs, the battery replacement they require is highly impractical, costly and sometimes impossible because of the nature of the environment [BS06]. The short lifetime of sensors also prohibits innovative extensions of such programmes to new areas. The broad objective of this thesis, therefore, is to maximise the usability of WSNs by developing protocols that enhance energy efficiency. Minimising energy consumption will improve energy efficiency in data collection and dissemination, thereby enabling WSNs to reach their full potential in monitoring vital environmental phenomena, as well as in other areas and applications.

1.2 Benefits of Energy Efficient Data Collection Protocols

Maximising energy efficiency in data collection applications would result in both practical and monetary benefits in a number of varied sectors. In order to demonstrate the value of the protocols developed in this thesis, benefits associated with five monitoring applications, including environmental monitoring, irrigation management, soil conservation, oil/gas pipeline management and reservoir management are outlined below:

1.2.1 Environmental Monitoring

Monitoring the environment with regard to certain phenomena such as flooding is extremely important as the devastation caused can be catastrophic [Pit08]. Early determination that flooding is imminent is critical in order to give emergency services sufficient time to set up protective measures and implement evacuation procedures.

Previous monitoring systems for flood detection used bulky units similar to that shown in Figure 1.3 which were expensive, had a low fault-tolerance and required manual data collection. Researchers have already demonstrated that WSNs are a cost effective and reliable alternative monitoring system for flood detection in developing countries [flo08]. The collection and wireless reporting of data to a remote base station extends the time period during which evacuation can take place. The algorithms developed in this thesis could further extend this evacuation period by increasing the rate at which events are reported.

Figure 1.3: Air monitoring unit

1.2.2 Irrigation Management

One of the consequences of climate change is a disproportionate distribution of water and global shortages of fresh water [Law08]. At the same time, world irrigation systems need to improve to increase food production rates in order to feed the growing global population [Ora91, PHP$^+$97]. This issue has highlighted the need for greater efficiency in irrigation management methods that minimise water wastage [CC82, Bag05]. One such method that has been developed is Regulated Deficit Irrigation, (RDI), which maintains a slight water deficit in order to improve the partitioning of carbohydrates and limit excessive vegetative growth [CC82]. This process requires accurate and real time soil sensing in order to irrigate little and often.

WSNs are natural candidates for RDI and, in fact, sensors have already been deployed in such systems to manage water usage and optimise production [HP05]. But irrigation systems could be further improved and expanded through the use of the algorithms developed in this thesis. These algorithms allow relevant events, such as drought, to be captured and logged without the need for continual and constant levels of monitoring. Thus they reduce the amount of redundant data being relayed, while increasing the lifetime of the network.

1.2.3 Soil Conservation

Every year, the continent of Africa loses billions of dollars worth of soil nutrients [HB06] due to poor soil conservation practises. A combination of deforestation, soil erosion and inadequate crop rotation policies have led to poor crop yields and food shortages. Consequently the vital importance of demonstrating to policy makers and development partners the positive contribution that can be made by increasing and sustaining agricultural productivity through soil conservation is increasingly apparent [TB09]. Soil conservation practises are facilitated through the monitoring of soil conditions and the algorithms developed in this study could facilitate the collection of such decision data. Through analysis, this data could then be used to test and measure the integrity of agricultural management practises, allowing corrective action to be implemented and providing long-term improvement in crop yields and food production [CLBA+07, PRP+06].

1.2.4 Oil/Gas Pipeline Management

Oil and gas pipeline operators lose millions of pounds every year due to leakage incidents in pipeline infrastructure [TB08]. Therefore, early detection and location of leakages is essential. Traditional pipeline monitoring techniques use centralised systems, connecting measurement devices to remote terminal units installed along the pipeline infrastructure. Collected data is then typically delivered to station operators at control centres via satellite systems. This method of data collection is less effective and reliable when compared with a system that uses wireless sensors. Wireless sensors offer a decentralised approach whereby a large number of small nodes can be deployed throughout the pipeline infrastructure without the need for a planned framework [SNMT07]. As a result, WSNs have a higher fault tolerance when compared with traditional monitoring methods. Furthermore, the algorithms developed in this study would lead to faster relaying of reports when leakages occur, allowing remedial action to be taken more quickly.

1.2.5 Reservoir Level Monitoring

Dams play a critical role in many developing countries where hydroelectric power is used to generate electricity [Cal07]. Hydroelectric power comes from the potential energy of dammed water driving a water turbine and generator. The effectiveness of a dam is heavily influenced by its water holding capacity and, in sites with high erosion rates, the build up of in-flowing sediment and mud can limit the efficiency of a dam. Sensors are therefore used to monitor the rate of mud build-up in the dam to indicate when the capacity of the reservoir has decreased to a low level and to highlight the need for dredging [Wai07]. Previous work on sensor monitoring of hydroelectric dams used a centralised system where deployment was planned and fixed and thus changes or improvements were either difficult or impossible to implement [FHAM95]. The protocols developed in this study could be deployed more simply and in a flexible manner; a large number of nodes could be scattered across the site of the dam. The resulting data would be more accurate and provide a more complete analysis of the health of a dam because these sensor nodes could collect data from various locations across the site of the dam, rather than from one centralised point.

1.3 Objectives of the thesis

In comparison with other alternative electronic monitoring devices, WSNs have the potential to be effective for data collection and dissemination in a wider variety of applications. However, they are restricted by their short life span. The improvement of energy efficiency in WSNs would not only enhance the effectiveness of data collection in current monitoring programmes, but would also facilitate innovative extensions of such programmes to new application areas.

The specific objectives of the thesis are as follows:

- To develop data collection techniques that efficiently manage energy usage of sensor nodes in order to elongate battery life in sensor networks
- To develop techniques that will improve data collection efficiency in multi-hop sensor networks by reducing delay in communication between a source node and a remote base station
- To demonstrate that, by combining the new techniques developed, WSNs could be a more effective tool for monitoring events in diverse application areas

This thesis addresses the problem of energy limitations affecting WSNs by developing techniques that minimise energy consumption. The most effective method of achieving such minimisation is through the use of energy efficient data collection algorithms. To date, many algorithms have already been developed in order to address energy inefficiency in the communication unit of a sensor node. It is generally believed that the communication unit consumes the highest proportion of energy while energy expenditure in the sensing unit is negligible and therefore not worth addressing [CES04]. However, over a given period of time many specialised sensors expend more energy in the sensing unit than the communication unit [HHM+09]. Therefore, it is particularly advantageous to develop methods which reduce energy consumption in the sensing unit in order to considerably increase the lifetime of a sensor node.

Dynamic sensing algorithms have already been developed which increase the lifetime of a sensor node by trading off energy consumption in the sensing unit with the quality of data collection required in an application. Essentially these algorithms achieve energy savings by adjusting the sampling rate of a sensor. However, adjusting the sampling rate of the sensor can cause a number of events to be missed because the sensing unit is switched off when an event occurs or lead to increased false alarms if the sensing frequency is too high. In order to address this trade-off relationship between energy efficiency and data comprehensiveness, a Dual Prediction and Probabilistic Scheduler (DPPS) is proposed in this thesis. DPPS can be used to improve energy efficiency in event detection and data collection in monitoring applications. More specifically DPPS improves the energy efficiency of the sensing unit in a sensor node. This is achieved by combining Compression and Load Balancing techniques through a Dual Prediction Scheme. Thus DPPS adjusts the sensing frequency more effectively while also allowing fewer missed events and false alarms during the monitoring process when compared with other scheduling protocols.

Another issue that needs to be addressed in order for WSNs to reach their full potential as monitoring devices is the Data Forwarding Interruption problem which occurs during the process of dissemination [LKR07]. Sensor nodes disseminate data to a remote base by relaying through several intermediate nodes in a hop-by-hop manner. Delay is incurred when the forwarding of data is interrupted by sleeping nodes. Either data cannot be passed on because of an adjacent sleeping node, or data is lost when a node falls asleep before it has forwarded the data. Further delay is incurred due to the Broadcast Storm problem [NTCS99]; when an event is detected by several nodes at the same time in the same geographical region, their radio signals overlap resulting in high contention for a share of the wireless medium. Such congestion leads to increased collisions necessitating the rebroadcasting of data, thus considerably increasing delay. In the second part of this thesis, the effects of the Data Forwarding Interruption problem in periodic scheduling algorithms are addressed using an Adaptive Detection-driven Ad hoc Medium Access Control (ADAMAC) algorithm. ADAMAC minimises the delay incurred during the dissemination of data whilst at the same time reducing energy consumption.

In light of the energy constraints involved when using WSNs in the monitoring of critical events, the thesis will demonstrate that self-organising scheduling techniques can improve the efficiency of data collection and dissemination and facilitate the extension of WSNs to more application areas than had hitherto been readily possible.

These objectives have resulted in the following publications thus far:

C. Edordu, L. Sacks, "Self Organising Wireless Sensor Networks as a Land Management Tool in Developing Countries: A Preliminary Survey", LCS '06: Proceedings of the 12^{th} London Communication Symposium,IET, IEEE UK and RI Communication Chapter, 2006

C. Edordu, V. Shum, N. Thornhill and Y. Yang, "Environment Aware Sampling For Sensor Networks", LCS '07: Proceedings of the 13^{th} London Communication Symposium,IET, IEEE UK and RI Communication Chapter, 2007

C. Edordu and Y. Yang, "Towards ARIMA Models for Resource Management in Sensor Networks", PGNET '08: Proceedings of the 9^{th} annual postgraduate symposium on the Convergence of telecommunications, networking and broadcasting,The School of Computing and Mathematical Sciences LJMU, 2008

C. Edordu, Y. Yang, "Dual Prediction and Probabilistic Scheduling for Efficient Event Detection", Wireless ViTAE '09: IEEE International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology, 2009

C. Edordu, Y. Yang, "Lessons from a Pilot Deployment of Energy Efficient Data Collection Protocols in Wireless Sensor Networks", Sensors & their Applications (S & A XV), Journal of Physics, Volume 178, Issue 1, Institute of Physics Publishing, 2009

1.4 Organisation of the thesis

This thesis comprises of six chapters. This introductory chapter provides the primary motivation for the thesis which is to demonstrate that self-organising scheduling techniques can be used to improve the efficiency of data collection and dissemination in a Wireless Sensor Network, thereby facilitating the extension of WSNs into more application areas.

In **Chapter 2**, literature covering model-based techniques for optimising energy efficiency in wireless sensor networks are reviewed. The techniques of data compression, load balancing and scheduling are discussed in detail and relevant methodologies are examined.

Chapter 3 of this thesis outlines principles of self-organisation in communication systems. The advantages and disadvantages of both centralised and self-organised systems are discussed and contrasted. In particular, comparisons are made with regard to robustness and scalability.

A Dual Prediction Scheme and an energy efficient data collection framework are presented for data collection in self-organised systems in order to manage the instability and unpredictability inherent in such systems. By addressing these problems, the framework allows increased energy savings to be made in WSNs.

In **Chapter 4**, a Dual Prediction and Probabilistic Scheduler (DPPS) is developed as a means of improving energy efficiency in the use of WSNs. DPPS is then compared and contrasted with eSENSE, an alternative data collection protocol, with regard to energy consumption and data collection accuracy.

In **Chapter 5** an Adaptive Detection-driven Ad hoc Medium Access Control (ADAMAC) algorithm is developed in order to address the Data Forwarding Interruption problem and the Broadcast Storm problem. ADAMAC is then compared and contrasted with SMAC (Sleep Medium Access Control), a periodic scheduling protocol, with regard to minimising end-to-end delay and limiting energy consumption.

The thesis culminates in **Chapter 6** where conclusions are drawn and possible future extensions to this study are considered.

Chapter 2

Data Collection Systems and Energy Efficiency

2.1 Introduction

Monitoring of phenomena is an age-old practice widely conducted in many areas of society including business [DAL+10], energy related industries such as oil and gas [TCP09, CPD08], agricultural [Bag05] and environmental entities [SHX+09]; defence and security administration [ASYS02]; and for policy making and governance [PRP+06]. Devices used for monitoring are numerous and have experienced improvements over the years to enhance their effectiveness. Several phenomena, for example solar eclipses or the movement of migratory birds, attract considerable public interest and the monitoring of such events is often widespread and voluminous. However, this thesis focuses on the collection of data which can be used for vital decision making.

Environmental monitoring requires that sufficient data is collected in order to record certain phenomena. The effectiveness of this process however, can be constrained by limited resources. While older monitoring devices were mechanical, the most recent versions are electrical and electronic in form. Indeed, since the development of semiconductor technology in the 60's, a host of electronic devices have emerged which collect data on given phenomena in alternative forms including sound, light, heat and other properties [Jeo09]. Among the most modern of such electronic monitoring devices are wireless sensor networks which possess numerous advantages (such as robustness to failure) over other monitoring technologies. Unfortunately, in spite of these advantages, their effectiveness is constrained by the limited energy supply available to them during data collection [DGM05].

Improving the effectiveness of WSNs as tools for data collection and dissemination has attracted the attention of researchers and scholars who seek to bring energy efficiency to monitoring applications. The purpose of this chapter is to demonstrate that although the problem of energy efficient data collection has been explored to alleviate constraints in a sensor network, further improvements are needed. In multi-hop WSNs for example, bringing efficiency gains in environmental monitoring necessarily involves reducing end-to-end delay; this is the time elapsed between the occurrence of an event at a source and its detection by a base station (BS) several hops away.

This chapter is organised into four sections. Section 2.2 examines the anatomy and functioning of the typical network hardware of a sensor node in order to provide a basis for illuminating efficiency issues of the system. Section 2.3 reviews writings and techniques designed to achieve efficiency through the use of models embedded in the WSN which are aimed at controlling the functioning of the system. Section 2.4 evaluates the merits and limitations of the major techniques examined and provides a basis for developing new techniques. Finally, section 2.5 summarises the discussions.

2.2 The Wireless Sensor Network and Data Collection System

The typical wireless sensor network in a monitoring and data collection environment has a number of standard components as depicted in Figure 2.1. Because efficiency issues immediately arise as a result of the design of these components, it is appropriate to formally outline their role and functions. The major components of the WSN include the sensing, communication, processing and power units. Their functions and resource needs are outlined below.

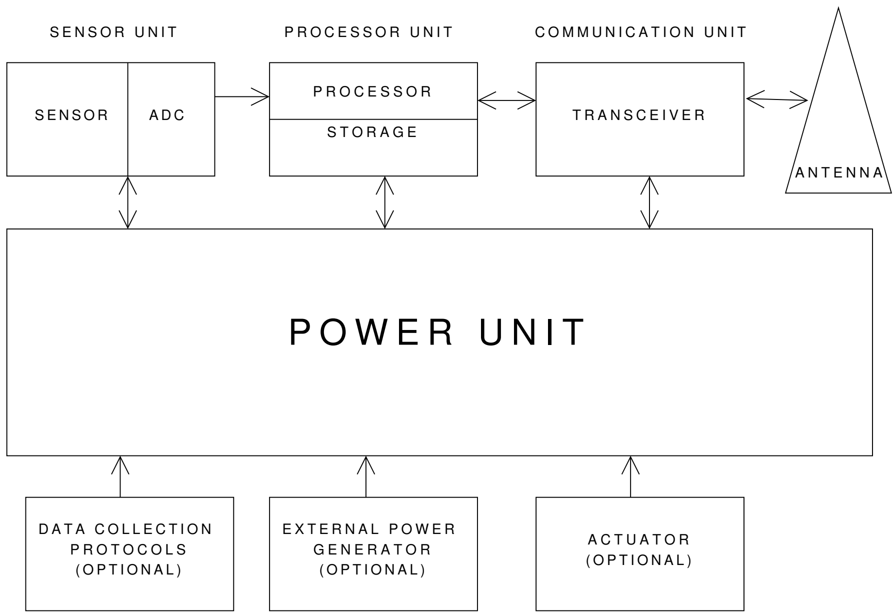


Figure 2.1: Typical hardware components in a wireless sensor node

Sensing unit - This is the part of the sensor node which physically reads and collects data from an environment. According to *Raghunathan et al.* [RSPS02], such units fall into one of two categories: passive and active. Passive sensors take measurements at a point in space without directly interacting with the point through active probing. Some passive sensors can be self-powered and hence can operate without a dedicated power supply. Examples of passive sensors include light sensors, thermometers and pressure sensors. Active sensors on the other hand must probe the environment to take measurements as exemplified by sonar and radar sensors. A large majority of sensors used in WSNs are passive because they are less expensive and consume less energy. Traditionally it has been argued that the energy used in the sensing unit is negligible compared to the other components in a sensor node. However, as will be discussed, this assumption is not always valid. The development of techniques in Chapter 4 is motivated by those applications in which the energy usage in the sensing unit is considerably higher than that of the other units.

Communication unit - Considerable research has been done on the communication unit because it is believed to be the most energy demanding component of a sensor node [Hae03, MC02]. Indeed, it is thought that the power required to send just one bit of information could power 1000 processor operations [YG03]. The transceiver units form a major part of the communication unit and are tasked with the transmission and reception of data. Some examples of popular radio transceiver units include the CC2420 family and the EMBER RF transceiver range, both of which consume about 20 mA per cycle for either data transmission or reception [Emb04, cc207]. This leads to very significant energy consumption when the communication unit is used over extended durations [YG03]. Because this energy increases by a factor proportional to the squared transmission distance, short range transceiver units that utilise multi-hopping scheduling protocols have become the standard. Typically, these protocols assume that applications are insensitive to end-to-end delay and therefore trade-off this delay in order to increase energy savings. Chapter 5 is motivated by the need to reduce energy consumption and limit end-to-end delay in monitoring applications.

Processing unit - The processing unit is a microunit responsible for computation and provision of intelligence. In conjunction with the operating system, it delivers and receives instructions from the sensing and communication units through microdevice drivers as shown in Figure 2.2. Energy consumed in the processing unit can be subdivided into switching and leakage energy. Energy is consumed during switching when software instructions are executed. Leakage energy concerns the nominal consumption of energy even when no computation is occurring and can be as high as 50% of the total computing energy [WH06].

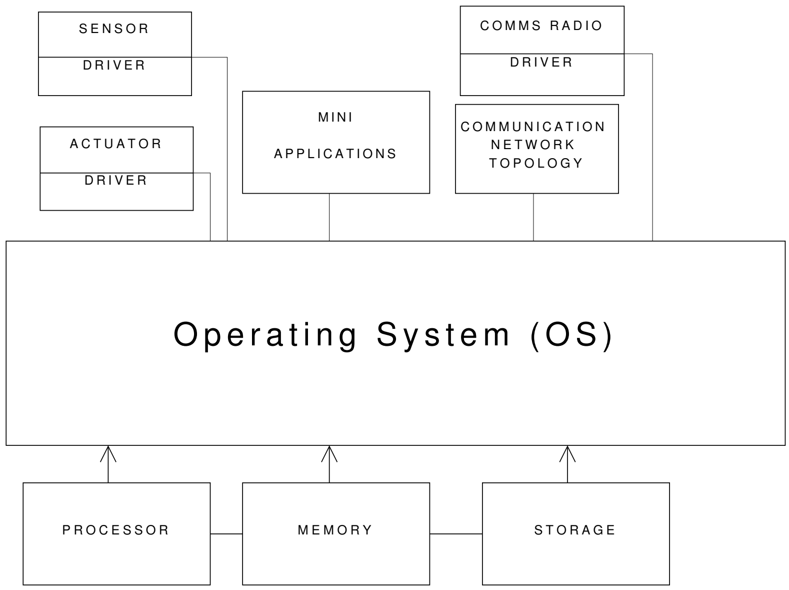


Figure 2.2: Typical software components in a wireless sensor node

Power unit - A suitable power unit is required to maintain all computational, sensing and communication operations from a few hours to several years depending on the application. Battery sources within these units may be categorised into two groups namely non-rechargeable (primary) and rechargeable (secondary) batteries. In untethered wireless sensor nodes, the finite supply of energy, typically 2 AA batteries with $2.2 - 2.5Ah$ at $1.5V$ [RSF$^+$04], presents unique problems because replacements can be costly and recharging impossible. For example, renewable methods like photovoltaic panels can only generate about $15mW/cm^2$ which, for a standard $10 \times 10cm$ sized panel, would amount to only $1.5W$ assuming consistent replenishment [RAdS$^+$00]. Another method of replenishing depleted power supplies has formed a body of work in itself called *energy scavenging* [RAdS$^+$00, Rou03] which seeks to utilise sources of energy around a node's immediate environment.

The four components of the sensor node as outlined above: communication, sensing, power and processing units, represent the hardware environment in which specific efficiency issues can be addressed using model-based data collection techniques.

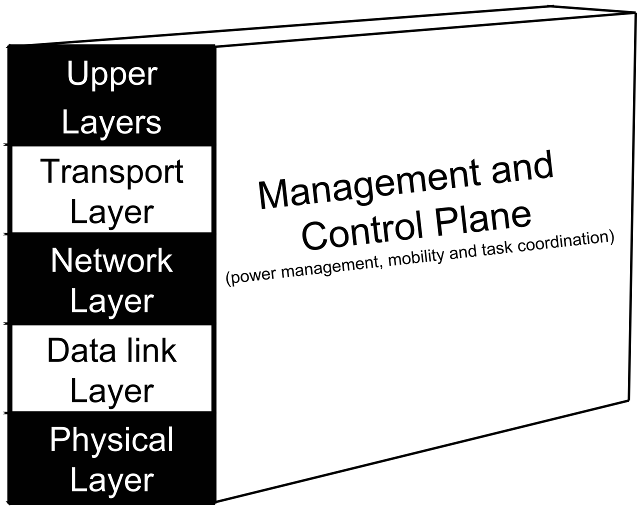


Figure 2.3: Wireless sensor network protocol stack

Data Collection System:

Reducing energy consumption in data collection protocols may be discussed in terms of the protocol stack design, which illustrates the different levels of communication required within a network. At its conception in the early 80's, Zimmermann's protocol stack design was acclaimed to be an accurate representation of the state of networking in communication systems [Zim80]. The advantage was that it had clearly defined layers which facilitated the partitioning of the network into smaller parts and thus allowed protocols to be configured independently in each layer. Since the invention of WSNs, a *management plane* has been superimposed on Zimmermann's original architecture, as shown in Figure 2.3.

WSNs require a more flexible protocol stack which includes *management planes* for power, mobility and task execution because of the highly dynamic environments in which sensors are deployed. The power, mobility and task management planes monitor the power, movement and task distribution respectively and help WSNs designers develop protocols that co-ordinate sensing tasks between nodes in order to reduce overall energy consumption and allow sensor nodes to perform a broader range of functionality [KFV11]. Table 2.1 summarises some of these functions performed using WSNs.

Table 2.1: Functions of layers in the protocol stack

Layers	**Communication Protocol**
Upper 5-7	In-network applications for data aggregation, query processing and data collection
Transport 4	Transport layer for assuring data reliability and integrity
Networking 3	Networking including routing, topology control
Data link 2	Managing medium access, timing etc.
Physical 1	Communication channel, sensing, actuation

2.3 Model-based Data Collection Techniques in Wireless Sensor Networks

Given the critical roles of the various components which make up a sensor node, efficiency is essentially an issue of minimising energy consumption within a given unit while still maintaining the ability to collect and disseminate data effectively. It is advocated by some writers [Zha03, RV06] that as communication energy is a major factor influencing the total energy dissipated in a sensor node, only vital parts from a stream of data should be transmitted, thereby reducing the overall amount of data transmitted leading to a decrease in energy consumption. Other writers have advocated in-network processing techniques which achieve energy reductions by processing data locally and using short range multi-hop communications to disseminate data from a source to a destination [PK00]. Another method of minimising energy consumption is through scheduling sensor nodes in multi-hop systems to sleep periodically. However, it is apparent that efficiency gains achieved in one area typically compromise performance gains in another area. For example scheduling sensor nodes conserves energy by putting nodes to sleep but, as a result, end-to-end delay is increased.

The next section reviews research that has been done to generate alternative techniques for addressing the energy efficiency issue in sensor networks. This will include a discussion of the more traditional techniques for minimising energy consumption in the communication unit, as well as more recent approaches which include minimising energy in the sensing unit. Figure 2.4 summarises the model-based data collection techniques discussed in this section.

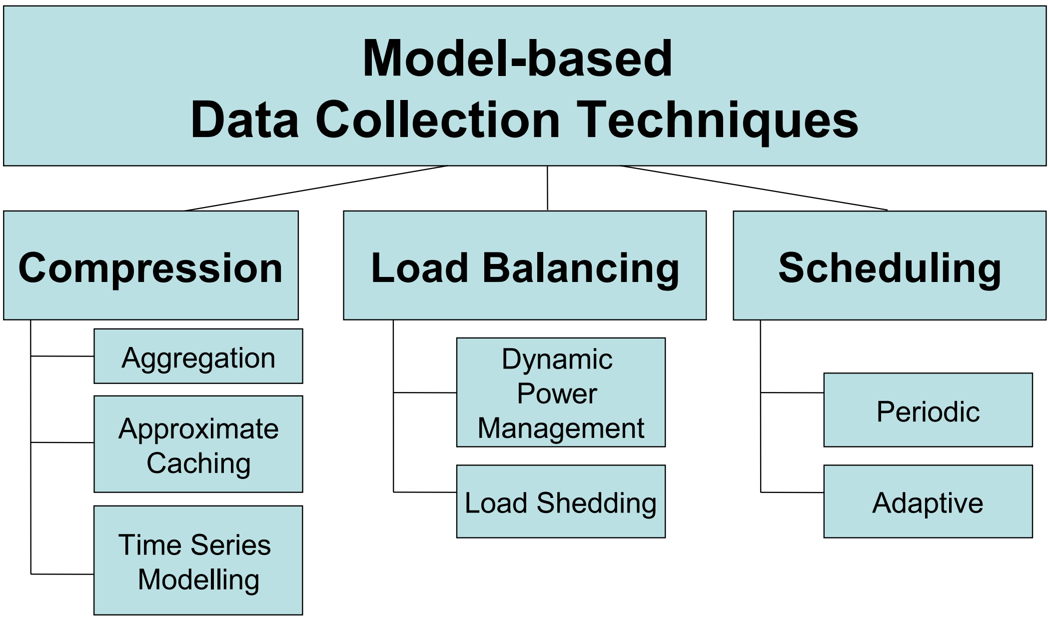


Figure 2.4: Model-based data collection protocols

2.3.1 Compression

Compression techniques use models, estimates or other forms of approximation to reduce the amount of data required to represent a given time series. These include but are not limited to Aggregation, Time Series Modelling and Approximate Caching. By compressing data, energy consumed during data transmission and reception is reduced.

2.3.1.1 Aggregation

Aggregation is a type of compression technique which seeks to gather and fuse critical data before delivering them to a base station for potential access by a user. In order to maximise efficiency, several factors including the architecture of a network and the aggregation protocol have to be considered. Several aggregation protocols have been specifically designed for implementation in specific network architectures [HHW97, HP05, MFH02, ABDH08]. In energy constrained networks, it is unnecessary for all sensors to forward collected data directly to the base station. Instead data communication is reduced by selecting local aggregator nodes, also called *clusterhead* nodes, which are given responsibility for long-ranged communication with remote base stations. Data communication is reduced by aggregating data from *non-clusterhead* nodes before transmission by *clusterhead* nodes. One of the earliest and arguably most popular cluster-based data aggregation protocol is LEACH (Low Energy Adaptive Clustering Architecture Hierarchy) [HCB00]. LEACH focuses on reducing the energy dissipated for communication by using a randomised *clusterhead* selection technique to evenly distribute the communication load among sensors and therefore increase the network lifetime as compared to a direct transmission method. However, LEACH, like other cluster-based aggregation protocols, is only aimed at reducing data communication. It is not appropriate for use in environmental monitoring because it does not address the main emphasis of this thesis which is limiting energy consumption while also enhancing the quality of the collection process.

One problem encountered in cluster-based protocols occurs when *non-clusterhead* nodes are located a long distance away from the *clusterhead* node. This makes long transmission distances necessary thus increasing energy consumption considerably. This is resolved using a chain-based data aggregation scheme where sensors transmit data only to neighbours close by. *Lindsey et al.* developed PEGASIS (Power-Efficient data GAthering Protocol for Sensor Information Systems) [LR02] using such an idea. PEGASIS organises all nodes into a linear chain from a source to a base station. The node farthest away from the base station sends data to its closest neighbour. This neighbour in turn fuses this received data with its own and forwards the combination to the next node along the chain. This process continues until a *clusterhead* forwards the aggregated data to the base station. Although PEGASIS uses aggregation techniques to reduce energy consumption in the communication unit, it is less suited for some monitoring applications because it does not seek to improve the quality of the collection process.

Researchers such as *Yoon et al.* [YS05] and *Sharaf et al.* [SBLC03] proposed the earliest protocols in a class of aggregation protocols which offer some guarantees on the quality of the collection process. TiNA (Temporal in-Network Aggregation) developed by *Sharaf et al.*, allows users to specify a tolerance requirement during the data collection process. This means that data can be compressed while ensuring a certain level of quality in the data collection process.

The efficiency of aggregation protocols is highly influenced by the topology and density of a network [IEGH02]. The effectiveness of aggregation is hindered if there are not enough nodes to aggregate data across a particular path in the network. In this instance spatial and temporal correlations in the data itself can be used to directly limit the volume of data transmitted by each node. This limitation of data can be achieved using Time Series Modelling techniques as discussed in the following section.

2.3.1.2 Time Series Modelling

Time Series Modelling techniques are used to reduce the amount of information required to represent raw time series data and thereby limit the amount of communication required for data collection [Mid00].

An example of such an approach is PCA (Piece-wise Constant Approximation) built by *Iosif Lazaridis and Sharad Mehrotra* [LM03]. PCA is effective because data are approximated at a base station using a prediction model. Using temporal correlation among readings, the amount of data transmitted is significantly reduced and therefore energy is conserved.

Further reductions of energy usage in the communication unit can be achieved using the spatial correlation exhibited among data. For instance in SBR (Self Based Regression), a data compression technique by *Deligiannakis et al.*, spatial correlation between multiple data streams is modelled using a regression type technique [DKR04]. Although results indicated that SBR increased the quality of measurement while saving energy, it incurs significant latency in order to populate the sensor network with measurements before any compression can be carried out. Furthermore the degree of spatial correlation between nodes is greatly reduced as the distance between these nodes increases. These factors render SBR unsuitable for use in environmental monitoring applications.

A more robust approach to compression involves both spatial and temporal correlation. One of the earliest adopters of spatio-temporal correlation techniques was DIMENSIONS, a general purpose data collection tool [GEH03]. DIMENSIONS sought to support long term data collection and communication in the context of resource constrained networking while satisfying a user specified quality constraint on collected data. Spatio-temporal compression in DIMENSIONS was achieved using a *Wavelet* type technique. Because this technique is computationally intensive and requires that nodes take measurements at regular intervals, it is inadequate for many monitoring applications.

More energy can be saved in communication units if sensor nodes are able to adjust their spatial and temporal sampling rates in accordance with the occurrence rate of an event being monitored [BTC05]. For example, it is desirable to decrease the amount of data transmitted by reducing the sampling rate when the dynamics of an environment are consistent and increasing the sampling rate as the dynamics change more rapidly. Adjusting the spatio-temporal sampling rate requires co-ordination of all sensors but can yield useful results. The potential for efficiency gains using this method was demonstrated by *Mehmet C. Vuran et al.* whose CC-MAC algorithm modelled spatial and temporal correlations among WSNs and produced an energy efficient medium access and data transportation protocol [VAA04, VA06]. To account for spatial correlation, the authors proposed a spatial correlation function which summarised the reliability of data received from spatially separated nodes around an event. This spatial function was supported by a temporal function that modelled the relationship between sensor measurements. CC-MAC reduced energy consumption and ensured reliable results by collecting data using a small subset of powered nodes (as opposed to all nodes) around an event.

Further work seeking to take advantage of spatio-temporal correlations using control theory was done by *Emekci et al.* who developed a sensor monitoring framework called BINOCULAR [ETAA04]. A spatio-temporal model was used to estimate the value of readings of all sleeping sensor nodes using a few working nodes thus saving energy. BINOCULAR however assumes that a subgroup of working sensor nodes always exists around sleeping nodes. In some monitoring applications, this scenario cannot always be guaranteed and therefore BINOCULAR is inadequate.

A more flexible method of taking advantage of the spatio-temporal correlations was introduced by *Deshpande et al.* who built a data collection framework, called BBQ, using probabilistic models [DGM^{+}04]. BBQ based these probabilistic models on time varying multivariate Gaussians in order to allow users to specify tolerances and confidence intervals during the collection process. Analogously DSC, by *Lidan Wang et al.*, used similar probabilistic models to construct a distributed algorithm for data collection which exploited correlations while also taking advantage of the broadcast nature of wireless sensor networks [WD08]. Experiments done using synthetic and real datasets suggested that both BBQ and DSC reduced the communication overhead and limited the amount of error to within the application requirements. However, these algorithms have been designed to report all data to a base station in their entirety while satisfying an error requirement. In such a system no distinction is given to capturing event data and therefore would be inefficient for environmental monitoring. The authors also assume that spatial data will always have high correlation thus making compression advantageous. In fact results indicate that for correlation ratios lower than 0.75, DSC offers no advantage over traditional data collection techniques such as aggregation [WD08].

Time Series Modelling techniques have been shown to be effective at reducing the communication energy even with low correlation ratios. *Liu et al.* produced the first Auto Regressive Integrated Moving Average (ARIMA) model for energy efficient data collection in a WSN [LWT05]. The ARIMA model was constructed using a combination of real and predicted data. This allowed energy savings to be made because a sensor node transmitted real data to a base station less frequently. *Liu et al.* also expanded their data collection framework into an ARIMA-based spatio-temporal correlation algorithm [LWP07]. More energy was saved by adjusting the ARIMA model to use a subset of sensor nodes for data collection while other sensor nodes could be switched off. The ARIMA model however requires a long and computationally intensive training phase which is impractical in some sensor-based monitoring applications. Using a lighter weight time series model may help to avoid this intensive training phase. *Tulone et al.*, for instance, developed PAQ (Probabilistic Adaptable Query) [TM06]. PAQ used a light weight AR (Auto Regressive) model to predict sensor measurements in order to efficiently answer queries in a sensor network database. Like Liu's ARIMA framework, energy is saved in the communication unit of sensor nodes because sensors only communicate when significant changes are detected. PAQ is constructed for a database application and is unsuitable for an environmental monitoring application. Moreover algorithms like PAQ focus on minimising energy in the communication unit at the expense of energy consumption in the sensing unit.

2.3.1.3 Approximate Caching

Prediction models used in Time Series Modelling techniques that aim for precision may require frequent updates when the measurements being taken are highly variable and thus consume high amounts of energy. Therefore the trade-off relationship between data comprehensiveness and energy efficiency must be considered. An effective approach for managing the trade-off between data quality and energy usage without a prediction model is to maintain a stored/cached value of sampled data at a base station. This technique is called Approximate Caching.

In Approximate Caching techniques, a sensor must automatically report data to a base station. This reporting, done by any sensor in the network, transmits measurements that exceed a specified threshold. This threshold is called the caching width. This transmitted data is cached at a base station for user access and analysis. Energy is saved in the communication unit because only data outside the caching width is transmitted to the base station.

Arguably the most popular approximate caching algorithm was proposed by *Olston et al.*; Olston's APS (Adaptive Precision Setting) algorithm adjusted the caching width so that the amount of communication energy expended by sensor nodes was minimised while satisfying precision requirements of the collecting application [OLW01]. As Olston's precision setting algorithm does not provide any latency guarantees during data collection, a more recent algorithm called QUASAR by *Qi Han et al.* was developed. QUASAR expanded APS by incorporating a latency constraint that allowed application users to explicitly guarantee response times of any requests made to a base station [HMV07, HMV04]. The authors proposed three models for guaranteeing this response time; an active-active (AA) model, an active-listening (AL) model and an active-sleeping (AS) model. In an AA model, the communication unit is always fully active; the communication unit alternates between being either fully active or listening in the AL model and the communication unit is either fully active or asleep in an AS model. Experimental simulations indicated that the AS model was the most energy efficient sensor state for data collection because it maximised the network's lifetime. Although these results demonstrate that AS models maximise lifetime, approximate caching is unsuitable for environmental monitoring because of a heavy reliance on prior knowledge about the characteristics of the data.

An alternative to Approximate Caching techniques which is useful in applications where less is known about the characteristics of the data being collected is Load Balancing.

2.3.2 Load Balancing

Load Balancing techniques can be described as a class of heuristic methods that use a set of rules or standards to sustain the performance level of a component in a sensor node. These rules reduce the amount of work done by a processing unit during data collection. Load Balancing techniques fall into two categories: Dynamic Power Management (DPM) and Load Shedding.

2.3.2.1 Dynamic Power Management

In Dynamic Power Management (DPM) techniques, energy consumption is minimised by scheduling the flow of data in order to allow a system to be idle when there is no workload to be processed. This is exemplified in MSUS (Multiple Sensing Unit Scheduling) proposed by *Poornachandran R. et al.* [PAC05]. MSUS was developed for managing energy consumption in a sensor node by scheduling the flow of data from the sensing to the communication unit. MSUS organises this scheduling by assigning the resources of a sensor node's processing unit according to a user's priorities. MSUS saves approximately 50% of energy in comparison with a greedy algorithm and limits missed events [PAC05]. MSUS, however, is unsuitable for the types of sensor nodes used in many monitoring applications because it requires sensor units with eight sensor states; most sensor units used in sensor-based monitoring applications have only two sensor states.

Sinha et al. demonstrated that by embedding DPM techniques into a sensor node's operating system, power consumption in the processing unit can also be reduced without compromising system performance [SC01]. This power reduction is demonstrated using various filters that calculate the expected processor's workload in the future: MA (Moving Average), EWMA (Exponential Weighted Moving Average) and LMS (Least Mean Square). In certain applications, where high performance is not always a requirement, energy can be conserved in the processing unit, by scaling down a processor's operating frequency and voltage to match the workload expected in the processor. This type of voltage scaling is also called DVS (Dynamic Voltage Scaling).

Because DVS is primarily focussed on minimising energy consumption in a processing unit, it has also been shown to be a useful tool for managing power usage in a wide variety of devices. For example *Xiaotao Liu*, *Prashant Shenoy* and *Weibo Gong* showed that by using statistical techniques to dynamically compute processor demands in multimedia devices, the rotational speed and voltage requirements on processors and disks can be reduced [LSG04]. This was done using a Time Series Based Power Management (TSPM) approach of which there are two forms: TSDVFS (Time Series Dynamic Voltage and Frequency Scaling) and TSDRPM (Time Series based Dynamic Rotations Per Minute). TSDVFS uses time series methods to optimise processor settings for various tasks and TSDRPM uses time series methods to adjust the disk rotational speeds to match the access patterns of the disk. Results indicated that TSPM saved 36% more energy in comparison with devices without any power saving features and 20% when compared with traditional power management based tools [LSG04].

Jacob R. Lorch and *Alan J. Smith* also used a DVS based algorithm called *RightSpeed* to minimise energy consumption by determining the optimum speed required for applications to complete tasks within deadlines [LS03]. This required that processors have the ability to alter their speed and voltage. Simulation results revealed that using *RightSpeed* can lead to an energy reduction in the processor unit of approximately 10% when compared with other commonly employed DVS algorithms. Despite these advantages, the high volume of data which a sensor node may have to transmit means that, although DPM techniques reduce energy consumption in the processing unit, these reductions are comparatively small when compared with the amount of energy consumed in the sensing and communication units.

2.3.2.2 Load Shedding

A more aggressive form of DVS where tasks are either dropped, or where components of a sensor are switched off in order to minimise energy consumption is called load shedding. Load shedding is a subset of DPM which involves decreasing the workload of a sensor component by shutting the system down when performance is below a predefined threshold. Recently, Load Shedding techniques and its variants have appeared in several publications because of their ability to conserve energy in multiple components of a sensor simultaneously [SR06, SBF^{+}07, SCB96]. As an illustration of the aggressive nature of DVS, *Srivastava et al.*, the authors of [SCB96] developed a prediction shut down algorithm. The prediction shut down algorithm minimised energy consumption by switching off portable devices using prediction based on various heuristics, such as the computational history of a device.

Jain et al. viewed DVS fundamentally as a filter where as much useless data as possible is filtered or suppressed from transmission in order to minimise energy consumption in the communication unit. This was done using an approach where a DKF model (Dual Kalman Filter) was used to predict readings at sensor nodes and a base station [JCW04]. Energy is conserved because DKF predictions, which satisfy an accuracy constraint, negate the need for some transmissions from a sensor node. This is because the base station-side DKF could be used to accurately approximate the actual reading. Similarly, *Santini et al.* used a Least Mean Square (LMS) based load shedding technique to minimise energy consumption in the communication unit by predicting readings at both sensor nodes and base stations [SR06]. Energy is saved because whenever predictions from the sensor node are within an accuracy constraint, the sensor node moves into a stand-alone mode (a lower energy consuming sensor state) which suppresses data from transmission. When contrasted with a Continuous Monitoring system, results showed that in the communication unit, a 92% saving in energy was realised using the LMS load shedding technique.

None of the algorithms listed so far exclusively focus on minimising energy consumption in the sensing unit. This is because the traditional view within the research community was that energy consumption within the sensing unit is negligible. This is not however the case when specialised sensing units are used. As an example, XBow's heading sensor, which measures azimuth angles, consumes $375mW$ of power for sensing compared with the $60mW$ used for transmitting in MICA2 nodes [LCS05].

To address this problem, eSENSE, an energy efficient sensing framework for wireless sensor networks was developed by *Haiyang Liu, Abhishek Chandra and Jaideep Srivastava* [LCS06].

eSENSE achieves energy savings by trading off energy consumption in the sensing unit with an application's data quality requirements. Essentially eSENSE adapts the sensing frequency in proportion to the chance that an event occurs. Hence the sensing frequency is relatively high when an event is likely to occur and relatively low when an event is unlikely. Between taking measurements, the sensing unit can be switched off thereby conserving energy.

eSENSE also includes a thresholding algorithm; this means that only certain events, which occur when the value of data exceeds an *event threshold*, are detected. eSENSE maximises the detection of event data by adjusting the sampling rate so that as the value of measurements increase towards the *event threshold*, the sampling rate is also increased. As the sampling rate increases, so does energy consumption in the sensing unit. Conversely, when the value of collected data is relatively far below the *event threshold* the sampling rate is low and therefore energy is conserved.

Although this method conserves energy, adjusting the sampling rate of the sensing unit inevitably causes some relevant events to be unseen or missed when the sensing unit is off. Therefore improvements in eSENSE are needed in order to facilitate a more efficient data collection process. This is the motivation behind the Dual Prediction and Probabilistic Scheduler (DPPS) proposed in Chapter 4. DPPS incorporates a Dual Prediction Scheme as discussed later in Chapter 3. This scheme ensures that energy is conserved while, at the same time, the number of unseen events is minimised by guaranteeing the precision of the collection process.

2.3.3 Scheduling

Efficient sensing is not the only essential requirement in data collection protocols. It is also important that sensor networks in monitoring applications reliably deal with delay-critical messages. In such applications an event message is triggered when the value of a reading exceeds an *event threshold*. This can occur in natural disasters such as flooding or before a fire outbreak. The ability to detect such events and report event messages quickly to a base station is critical so that early remedial action may be taken by emergency services. Such a rapid response to the occurrence of an event can be achieved using a Fully Active network.

In a Fully Active (FA) network, sensing and communication units of all sensor nodes are always active and therefore events that occur are detected without delay. Additionally, data between a source and a destination can be transmitted immediately because communication units are always on. Despite these advantages, a Fully Active network consumes the most amount of energy because all of a sensor node's components are switched on. A popular variant to a Fully Active network is Continuous Monitoring (CM). In CM, only the sensing unit of a sensor node are fully activated in order to detect any events that occur; the communication unit is switched off to conserve energy unless data needs transmission or reception. Although CM uses less energy than FA, a substantial amount of energy can still be consumed, especially when specialised sensing units are used, as previously mentioned.

A more energy efficient alternative is Scheduling techniques. Scheduling techniques elongate the lifetime of a network by putting a sensor node's components to sleep intermittently, thus reducing the energy consumed by those components. There are two main types of scheduling; Periodic Scheduling and Adaptive Scheduling.

2.3.3.1 Periodic Scheduling

Periodic Scheduling techniques offer an effective, yet simple means of minimising energy consumption in a node. This is done by causing a sensor node to sleep at regular intervals throughout the life of a network. Energy is conserved because sensor nodes only become active sporadically and can remain asleep for longer periods of time.

The most popular type of periodic scheduling protocol for a sensor network is SMAC (Sleep MAC) [YHE02]. The inventors of SMAC achieved energy savings by making nodes in a network sleep periodically. SMAC also produced further reductions in energy consumption by avoiding message *overhearing*. Message *overhearing* occurs when nodes wrongly receive data intended for neighbouring transmitting or receiving nodes. By ensuring that *overhearing* sensor nodes are asleep during communication, energy is saved. Periodic protocols however suffer from the inherent trade-off between energy consumption and delay. SMAC, as an example, increases energy savings by increasing the duration for which a sensor node is asleep. This increased sleep duration however also increases the end-to-end delay during the dissemination of data.

The end-to-end delay in periodic protocols is worsened by the effects of Data Forwarding Interruption which occurs during the dissemination of data from a source to a base station. Figure 2.5 shows an example which contrasts the effects of a system with and without Data Forwarding Interruption.

In both examples event data generated at a source is propagated to a destination via a relay sensor node. In the example without Data Forwarding Interruption, the event data is transmitted from the source to the destination within one second. Conversely, in the example with Data Forwarding Interruption, data arrives at the destination after a delay of two seconds. This occurs because the relay node is off at time $t = 1$, thus data cannot be passed by the source immediately. Instead, a source must wait until time $t = 3$ when the source and the relay nodes are awake simultaneously.

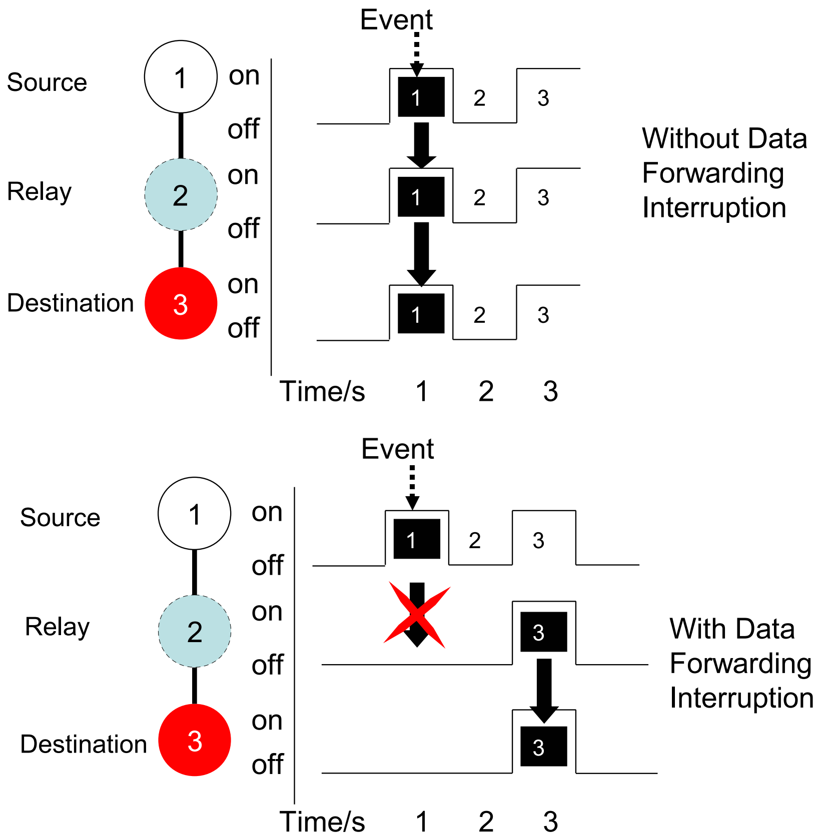


Figure 2.5: Data Forwarding Interruption Problem

Another problem which increases delay in Periodic Scheduling techniques is the Broadcast Storm problem [SYTCS99]. The Broadcast Storm problem occurs when an event is detected by several nodes at the same time in the same geographical region as shown in Figure 2.6. As their radio signals overlap, the sensor nodes experience high contention for a share of the wireless medium. This congestion leads to increased collisions as shown in Figure 2.6, thus necessitating retransmissions of data. As a result of the need for retransmissions, further delay is experienced before data can be successfully disseminated from a source to a destination. As will be discussed in the next section, the effects of increased delay caused by the Data Forwarding Interruption problem and the Broadcast Storm problem can be alleviated to a certain degree using Adaptive Scheduling protocols.

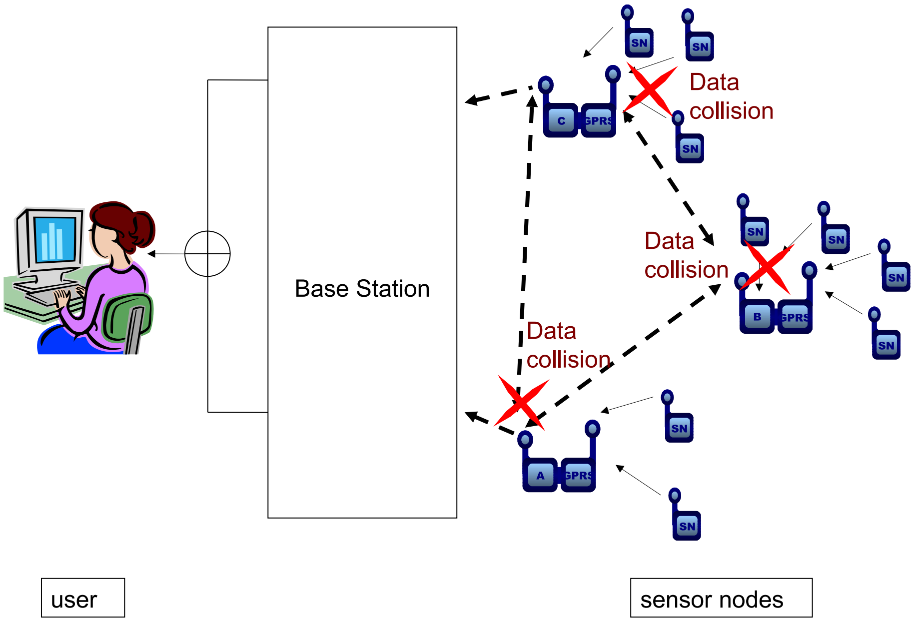


Figure 2.6: Broadcast Storm Problem

2.3.3.2 Adaptive Scheduling

Adaptive Scheduling protocols adjust the communication frequency of a network in order to match the energy and delay requirements of an application during data collection [MSG05, CPR03, vDL03]. For example in [MSG05] by *Miller et al.*, the energy-delay trade-offs are explored using PBBF, a probabilistic Adaptive Scheduling protocol for medium access control. PBBF follows the idea that for a given *reliability requirement* during data collection, the energy consumed in a WSN is inversely proportional to delay incurred. Based on the *reliability requirement*, PBBF probabilistically adapts a sensor's wakeup schedule so that its communication unit is activated less often, thus reducing energy consumption.

PBBF would be more suitable for event detection in monitoring application if sensor nodes were spatially co-ordinated. The implementation of such an idea was shown in [ZLN07] where the authors designed CAS. CAS worked by collaboratively adapting the wakeup schedule of a group of sensor nodes around an event in order to minimise monitoring before an event is detected. CAS therefore relies strongly on an application having a high density of nodes around an event area so that redundant nodes may be used for adaptive scheduling. In monitoring applications, such a high density of nodes around an event is not always possible. Moreover CAS does not address the problem of minimising end-to-end delay between a source and a remote base station.

Another method of achieving energy savings is when sensor nodes adjust their scheduling rates using any correlation found in collected data. This was exemplified in [GLY07] where *Gedik et al.* proposed ASAP - an adaptive sampling approach to energy efficient data collection in sensor networks. ASAP increased a WSN's lifetime while maintaining the quality of collected data by adaptively varying a subset of active nodes in a network. These active nodes are the ones which collect sensor readings. Other sensor nodes can remain asleep thus conserving energy. The value of readings from these sleeping sensor nodes can be predicted using a probabilistic model of the environment where data is collected. ASAP relies on a high level of spatial and temporal correlation in the monitoring environment so that data can be predicted in order to decrease energy consumption in a sensor network. In some monitoring applications with a high level of noise, such correlations between data may be difficult to detect or may be altogether absent.

Jain et al.'s Kalman Filter-based estimation model described in [JC04] is an Adaptive Scheduling technique which works well in noisy environments. The authors used a Kalman Filter (KF) to adjust the sampling rate during data collection in a noisy environment. This was carried out while optimising usage of energy and communication bandwidth throughout the WSN. Results demonstrated that the KF Adaptive Scheduling technique produced better performance with regard to resource utilisation in comparison with a periodic scheduling method while also minimising the error produced during prediction over all active sensor nodes. However, the KF model assumes all nodes can always communicate with the base station and that the base station manages the allocation of resources centrally. As will be discussed in section

3.3 of Chapter 3, such a centralised system differs from the system required for data collection in a sensor-based monitoring application.

An approach which does not use a centralised system for data collection is AMS (Adaptive Model Selection) presented by *Yann-Ael Le Borgne et al.* in [BSB07]. AMS works by exploring the trade-off between the fitness of a model for predicting environmental data and the resulting overhead caused by the need to update a model. AMS autonomously selected the best model for a data collecting sensor network from a set of candidate models. This was done using a *racing* algorithm whose task was to discard poorly performing models. The framework used in AMS is similar to that used in TiNA [SBLC03] in which a sensor node and a base station employ a prediction model. This framework allows energy to be conserved because the communication unit of a sensor node can be switched off more often. Although AMS achieves reductions in energy consumption while restricting inaccuracies incurred during the prediction of data, AMS cannot detect events.

In contrast to AMS, a Near Optimal Sleep Scheduling (NOSS) protocol is shown in [CAHS05] by *Cao et al.* for event detection in environmental monitoring applications. NOSS minimised detection delay by first selecting a subset of nodes that rotate periodically to give full coverage over an event area. Then end-to-end delay was reduced by applying a streamlined wakeup sequencing algorithm that transports data efficiently by constructing data routes between a source node and a base station. These data routes allow intermediate relay sensors to wakeup in time for the arrival of any data.

This idea of having a streamlined wakeup sequence of nodes was also employed in another Adaptive Scheduling protocol called DMAC [LKR07]. DMAC was designed by *Lu et al.* to solve the Data Forwarding Interruption problem in a one way tree topology. This was done by giving the active-sleep schedule of a sensor node an offset that varied according to a sensor node's location in relation to a base station. This offset allowed a staggered wakeup schedule to be created in which all nodes in a chain from a source to a base station can be notified of the impending arrival of data and wakeup just in time to receive and forward data. DMAC also addressed the Broadcast Storm problem by adaptively varying the number of active slots in a sensor node's schedule according to the traffic load in the network. Although DMAC provides energy savings and reductions in delay during the dissemination of data, DMAC is limited to one-way data gathering trees and thus is unlikely to be useful for a wide variety of monitoring applications where topologies can be highly variable.

An alternative to DMAC and NOSS is Alert, an adaptive low-latency event-driven MAC protocol for WSNs constructed by *Namboodiri et al.* [NK08]. Alert was primarily designed to address the Broadcast Storm problem in sensor nodes without requiring the type of prescheduled offset used in both DMAC and NOSS. This was done by minimising the contention among sensor nodes for a share of the wireless medium using a combination of time and frequency multiplexing techniques. By controlling the selection probability of each channel, this multiplexing technique allowed multiple frequency channels to be used so that contention for a share of the wireless medium is reduced.

Another adaptive scheduling protocol which addresses the Broadcast Storm problem during event detection is Best Effort Synchronisation (BES) presented by *Olston et al.* in [OW02]. Unlike Alert, BES is used to ensure that data collected and stored at a base station is accurate and up-to-date. This is done by optimally allocating bandwidth resources through scheduling. BES is therefore especially relevant to the Broadcast Storm problem because this bandwidth allocation improves communication among sensor nodes and therefore also improves the data collection rate. A *divergence* property was used to measure the accuracy of data at sensor nodes compared with data stored at a base station. Results carried out using real world data indicated that BES improves synchronisation between data stored at sensor nodes and a base station. Olston's *divergence* property, however, is geared towards the problem of cache assignment and does not consider energy efficiency in the data collection environment.

An energy efficient protocol that considers both Data Forwarding Interruption and Broadcast Storm problems is SSMTT [JRC08], an energy aware sleep scheduling algorithm for concurrent tracking of multiple targets. Designed by *Jiang et al.*, the main idea behind SSMTT was to use a proactive transmission mechanism which negated the need for nodes to send multiple messages to targets in a shared subarea. This proactive transmission mechanism improves energy savings by using sensor nodes already awakened for tracking so that unused nodes not within the surveillance area can be asleep. SSMTT however is unsuitable for environmental monitoring because this type of adaptive sleep scheduler allows a large number of events to remain undetected. Additionally, the number of undetected events increases as the speed of the target increases.

Despite offering many advantages, the protocols mentioned above are unsuitable for event detection in many monitoring applications because they do not efficiently detect events while limiting energy consumption and end-to-end delay. For example the aforementioned SMAC reduces energy consumption but incurs high delay during data collection because of the effects of the Data Forwarding Interruption problem. Therefore an adaptive scheduling protocol, called ADAMAC (Adaptive Detection-driven Ad hoc Medium Access Control), was developed in Chapter 5. Not only does ADAMAC reduce end-to-delay and alleviate the effects of the Data Forwarding Interruption problem, but it is also compatible with other energy efficient algorithms such as DPPS which enables energy consumption to be minimised.

A summary of the survey conducted in this section is shown in Table 2.2.

Table 2.2: Summary of Model-based techniques. Notice that the combination of DPPS and ADAMAC has the potential for reducing energy consumption in both the communication and sensing units. The DPPS/ADAMAC combination also would potentially have the capability for event detection in an application where limiting end-to-end delay is critical

Algorithm	Model-based technique	Primary energy saving unit(s)	Event detection Capability	Delay limiting Capability
LEACH	Compression	Communication	No	No
PEGASIS	Compression	Communication	No	No
TiNA	Compression	Communication	No	No
PCA	Compression	Communication	No	No
SBR	Compression	Communication	No	No
DIMENSIONS	Compression	Communication	No	No
CC-MAC	Compression	Communication	Yes	No
BINOCULAR	Compression	Communication	No	No
BBQ	Compression	Communication	No	No
DSC	Compression	Communication	No	No
ARIMA	Compression	Communication	No	No
PAQ	Compression	Communication	No	No
APS	Compression	Communication	No	No
QUASAR	Compression	Communication	No	Yes
DPM	Load Balancing	Processor	No	No
MSUS	Load Balancing	Processor/ Sensing	Yes	No
TSDVFS	Load Balancing	Processor	No	No
TSDRPM	Load Balancing	Processor	No	No
RightSpeed	Load Balancing	Processor	No	No
DKF	Load Balancing	Communication	No	No
LMS	Load Balancing	Communication	No	No
FA	Scheduling	-	Yes	Yes
CM	Scheduling	Communication	Yes	No
SMAC	Scheduling	Communication	Yes	No
PBBF	Scheduling	Communication	No	No
CAS	Scheduling	Communication	Yes	No
ASAP	Scheduling	Communication	Yes	No
KF	Scheduling	Communication	No	No
AMS	Scheduling	Communication	No	No
NOSS	Scheduling	Communication	No	Yes
DMAC	Scheduling	Communication	No	Yes
BES	Scheduling	Communication	No	No
SSMTT	Scheduling	Communication	Yes	No
eSENSE	Compression/Load Balancing	Sensing	Yes	No
DPPS with ADAMAC	Compression/Load Balancing	Communication/Sensing	Yes	Yes

2.4 Summary of Benefits and Limitations in Data Collection Protocols

This section summarises the main features of data collection protocols discussed in section 2.3 along with their key merits and limitations. Table 2.3 outlines the advantages and disadvantages of the major techniques reviewed in the last section.

Table 2.3: Summary of model-based data collection techniques in WSN

Techniques	Advantages	Disadvantages
Compression	•Saves energy by fusing data from different sources	•Limited effectiveness in networks with low node density •Poor energy management at the sensor unit •May place no distinction between event and non event data and therefore miss important events •May require high data correlation to be effective
Load Balancing	•Saves energy by reducing the amount of work done by components in a sensor node	•May lead to poor energy management at the sensor unit and/or the communication unit because these components may need to be active for extended durations •Energy savings can be relatively small compared with the overall energy consumption of a sensor node •Limited to applications with potential for high loss of data quality
Scheduling	•Saves energy by switching off components of a sensor node intermittently	•May not be adequate for some environmental monitoring applications because of the Data Forwarding Interruption problem •Important events may be missed while a sensor node is switched off •May only be effective in specific network topologies

There are three main classes of model-based data collection techniques; Compression, Load Balancing and Scheduling. The Compression class includes techniques such as Aggregation, Time Series Modelling and Approximate Caching. Generally the family of compression methods save energy by using some form of redundancy among nodes. In Aggregation, (examples of such methods include LEACH [HCB00] and PEGASIS [LR02]) cluster-based techniques are used to save energy consumption by compressing communication between source nodes and a destination.

Compression is shown to be useful when carried out on time series data using Time Series Modelling techniques. Such techniques rely on some form of correlation existing in historical datasets so that future data can be forecasted using an appropriate prediction model. Lazaridis' Piece-wise Constant Approximation (PCA) [LM03] and Deligiannakis's Self Based Regression (SBR) [DKR04] algorithms are good examples. When there are strong data correlations, further compression can be achieved using spatio-temporal correlation functions. For example in CC-MAC [VA06] such a spatio-temporal correlation function was used to compress data at a node's communication unit. A similar approach is exemplified in *Liu et al.*'s EEDC framework where an ARIMA model was incorporated into the data collection protocol in order to reduce energy consumption. Further compression can be carried out using probabilistic models which seek to approximate future values thus eliminating the need to collect them as shown in Desphande's Barbie Q (BBQ) [DGM$^+$04] and Tulone's probabilistic adaptive query (PAQ) [TM06] algorithms. None of these algorithms detect events efficiently and therefore would be unsuitable for environmental monitoring applications. Time Series Modelling techniques may also be disadvantageous because too much adaptation of the prediction model causes increased energy consumption.

An alternative solution to Time Series Modelling which does not use a prediction model is Approximate Caching. Here, a caching width is set up which limits the communication from a sensor node to a base station; only measurements with values that exceed the caching width are transmitted. Examples include *Christian Olston*'s APS [OLW01] and *Qi Han*'s QUASAR [HMV07]. Approximate caching protocols save energy in the communication unit but are heavily parameterised and require assumptions on unknowns, such as the interarrival time between events, in order to be effective.

The second class of model based techniques, Load Balancing techniques, are useful when little is known about the characteristics of the data being collected. Load Balancing techniques limit the workload done by a sensor node's components by shedding or scaling down excesses. There are two main types of Load Balancing techniques; Load Shedding and Dynamic Power Management (DPM). *Santini et al.*, using an LMS Load Shedding method, reduced the amount of data collected by 90% when compared with Continuous Monitoring [SR06]. Load Shedding was also shown to be effective in TiNA (Temporal in-Network Aggregation) where it eliminated 50% of the required communications while preserving data collection quality [SBLC03]. DPM is also a Load Balancing technique because, like Load Shedding, it scales down processing power in order to decrease the amount of work needed during data collection. This was exemplified by TSPM (Time Series based Power Management) which scaled down voltage settings during idle processor periods [LSG04]. In spite of the advantages of reduced energy consumption in the processing unit of a sensor node, traditional Load Balancing techniques require the sensing unit to be active all the time. Such an approach is ineffective when the power consumption of a sensing unit is high.

Compression and Load balancing techniques also have limitations which affect their ability to collect data efficiently, especially when event sampling and sensor usage are important metrics. eSENSE [LCS06] was developed in order to limit energy consumption in sensing units by using a stochastic scheduler during data collection. However, improvements in eSENSE are needed to reduce the number of missed events and false alarms during data collection. Therefore a Dual Prediction and Probabilistic

Scheduler (DPPS) is developed in Chapter 4; DPPS reduces energy consumption while simultaneously minimising missed events and false alarms by guaranteeing the precision of the data collection process.

The final class of model-based techniques is that which utilises scheduling to achieve improved data collection efficiency. In environmental monitoring applications where delay-critical event detection is carried out in multi-hop sensor networks, scheduling techniques are a useful alternative to using a Fully Active network. There are two main categories of scheduling: Periodic Scheduling and Adaptive Scheduling. The most popular periodic scheduling protocol for sensor networks is SMAC [YHE02]. However SMAC suffers from the effects of the Data Forwarding Interruption problem. Using Adaptive Scheduling techniques such as DMAC [LKR07] or NOSS [NK08], a prescheduled offset can be introduced so that sensors become awake just in time for the arrival of data thus limiting the effect of the Data Forwarding Interruption problem. However, protocols such as DMAC and NOSS can only be used in specific network topologies which are unsuitable in many monitoring applications. In Chapter 5, the thesis develops an Adaptive Detection-driven Ad hoc Medium Access Control (ADAMAC) protocol. ADAMAC, an adaptive scheduling protocol which alleviates the effects of the Data Forwarding Interruption problem, was constructed to be flexible in a wide variety of delay-critical data collection environments. ADAMAC was also designed to be fully compatible with other algorithms, such as eSENSE and DPPS in order to further enhance energy savings.

What should also be considered, while developing protocols such as DPPS or ADAMAC, is the cumulative effect of improvements to energy efficiency in the long term. According to what has come to be known as Jevons paradox, technological innovations which produce energy efficiency gains tend to increase rather than decrease overall energy consumption because any improvement leads to greater demand for that technology [Alc06]. This type of effect is well understood in economic terms where efficiency yields lower costs and prices. The decrease in price can serve to drive up sales and thus increase total energy consumption needed to produce such higher sales. Jevon's argument means that efficiency gains in data collection protocols are likely to be followed by new rounds of pursuit of efficiency because each success story will ultimately necessitate the development of new energy-saving technologies. The pursuit of energy efficient techniques and technologies is therefore likely to remain a subject of continuing research interest.

2.5 Chapter Summary

This chapter reviewed literature on hardware and model-based techniques for optimising energy efficiency in wireless sensor networks. This covered discussions of Compression, Load Balancing and Scheduling techniques for enhancing data collection efficiency. The combining of these techniques, though potentially beneficial for data collection applications, is still uncommon among protocols developed so far. The amalgamation of several model-based techniques could lead to the development of new protocols which address both the issue of energy efficiency and end-to-end delay. This will allow the deployment of wireless sensor networks across a wider variety of applications than had been hitherto readily possible.

Chapter 3

Self-Organised Network Architecture

3.1 Introduction

The first fruits of self-organisation were born shortly after the second world war when researchers were actively seeking to understand the workings of the human brain and to mathematically describe the complex logic underpinning the process of thought [Ric94, Ash62, For65]. Decades later, Eigen and Schuster's seminal work on self-organisation led to the currently held theory that complex systems consisting of smaller subsystems require a controlled autonomy for reliable performance [ES79]. At the heart of their discovery were the principles of self-organisation, that is, the idea of having 'creation' control itself without influence from a 'creator' [Bre94].

This chapter begins with a review of the characteristics of self-organised systems and goes on to outline the benefits and limitations of such systems. A framework is then presented which can be used to facilitate self-organised data collection in a wireless sensor network. Such a framework is necessary in order to improve the reliability of data collection within self-organised systems.

3.2 Self-Organisation: Concept and Characteristics

In [Dre06], *Dressler* defined self-organisation as a concept, used in systems, that enables a large number of autonomously operating subsystems to perform a collective task. Self-organising architectures are necessary in monitoring applications because in complex environments, where conditions are highly variable, centralised systems of control are inadequate [GS97].

Self-organisation can be clearly observed in a colony of ants: each ant acts autonomously in order to perform the global task of foraging food [KE01]. Self-organisation is also prevalent in organisms during mitosis; cellular signalling occurs during replication without any one cell taking overall responsibility. Inter-cell communication is also self-organising in nature because there is no overall global signalling plan existing among cells [ABL+94]. Self-organised signalling behaviour is also used to co-ordinate the immune system in mammals; when infection occurs, antibodies are deployed by the body in a self-organised fashion without any global control [JWT01]. Similarly, the authors in [McG04] argue that societal systems, such as geographical patterns in the arrangement of a population over a landscape, are self-organised.

Dressler also notes that self-organising systems show an overall behaviour that cannot be easily predicted or pre-progammed. Resulting complexity occurs because individual components in the system behave randomly and independently. The autonomous behaviour of components in self-organised systems results in the scalability of the system; components can be added or removed without drastically affecting overall performance. The fundamental characteristics in self-organising systems are summarised in Table 3.1.

Table 3.1: Properties of self-organisation

Property	Description
• No central control	• Global state information is unavailable or unused as each component of the system operates autonomously
• Emerging structure	• Autonomous subsystems perform a collective task
• Resulting complexity	• Individual components behave randomly
• Scalability	• Components may be added or removed without affecting the performance of a system

3.3 Standard Architecture in Data Collection Systems

As computing systems develop and progress, architectures have necessarily become more complex and the demand for methods of managing and controlling resources in such systems has substantially increased. Ideas for management and control have evolved from monolithic and centralised systems to distributed and self-organised systems. Figure 3.1 shows the movement from a traditional centralised control management architecture to a decentralised fully distributed system.

A monolithic system is a centralised system in which a single computer is used to control a subsystem. Generally, poor data transparency and limited scalability in centralised systems mean that the number of subsystems which can be managed under the supervision of a central computer is constrained. These inherent weaknesses led

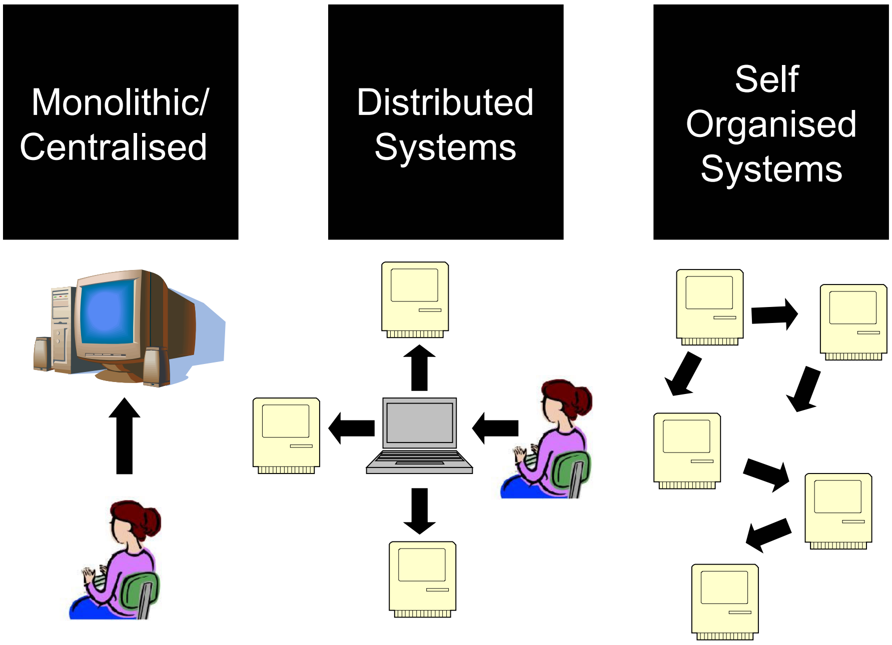


Figure 3.1: The evolution from traditional centralised control to self-organised decentralised systems

to the development of distributed systems in which one computer is used to control a group of subsystems. Within a distributed system, control is administered over multiple systems using a middleware architecture[1] [Tv02]. Although distributed systems have many advantages over centralised systems in terms of adaptability, improved fault tolerance and scalability, these benefits are limited when compared with the robustness and scalability of self-organised systems. Distributed systems also rely on synchronisation between subsystems in order to operate; this can be impractical and, unlike self-organised systems may require human intervention.

Self-organisation allows complex systems to become both manageable and controllable while increasing scalability. Self-organised systems permit global order through local interactions without the need for any central control making the system's architecture flexible and therefore desirable for use in monitoring applications. Also, whereas distributed systems require a control centre to operate, self-organised systems use a completely decentralised system which not only provides increased robustness and

[1]Middleware resides between the application layer and the physical layer of the protocol stack in order to support heterogeneous subsystems

fault tolerance properties, but also limits any overheads arising from the need for central control.

3.4 Self-Organisation and Wireless Sensor Networks

The integration of the concept of self-organisation into communication systems originated from research done to create control mechanisms for managing Internet traffic [ZG04]. It was however recognised that these control mechanisms would require modification in order to handle data collection and dissemination efficiently in variable environments. The integration of self-organisation into wireless sensor networks requires the development of models, whose features reflect the general characteristics of self-organised systems; these models can then be installed into devices, such as sensor nodes in data collection and dissemination applications, to complement and control complex networks while enhancing functionality and energy efficiency.

Four general characteristics inherent in self-organised communication systems, as summarised in [PB05], are outlined below:

i Emergent behaviour: Local behaviour rules in the network lead to the achievement of global goals. This is exemplified in the way shoals of fish use co-ordinated individual behaviour to protect the group against predators. Agglomeration behaviour of individual fish at microscopic levels results in an overall system behaviour at macroscopic levels. This emergent behaviour protects the group.

ii Implicit co-ordination: When using self-organisation in communication systems, information is not only communicated explicitly through signalling messages but nodes also detect and analyse transmitted information from neighbouring nodes. An example of this kind of implicit co-ordination is illustrated in Figure 3.2 along with an example of explicit co-ordination. As shown in the diagram, node A sends a message to node B at time t_1 which in turn sends it to node C at time t_2. Since node A is in the vicinity of node C it overhears node B's message to node C (at time t_2) and this acts as an implicit acknowledgement that node B received the initial message from node A. This differs from communication in a centralised system where communication is solely based on the explicit exchanging of signalling messages.

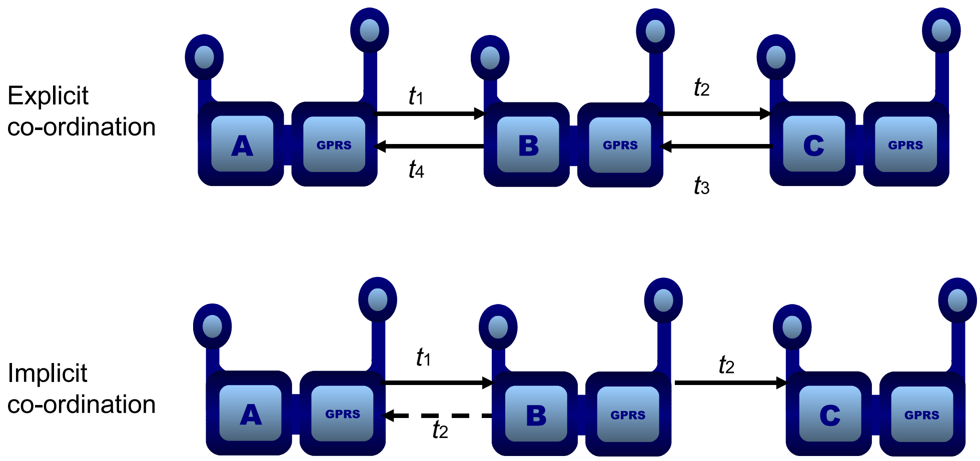


Figure 3.2: Communication exchanged among nodes are acknowledged either explicitly using dedicated message acknowledgments or indirectly through overhearing neighbouring transmissions.

iii Limited longevity of state information: A third characteristic is the limited amount of time that information on the *network state* lives in the system; localised interaction between nodes means less co-ordination is needed and therefore minimal state information regarding the condition of a network is maintained.

iv Adaptability of nodes: The fourth characteristic of self-organising WSNs is the capacity of nodes to adapt individually to changes in a network and its local environment. There are three distinct types of changes to which nodes can react adaptively. Firstly nodes are designed to be able to cope with changes in a network such as the failure or movement of a neighbouring node. Secondly, nodes are able to adapt to changes in parameters, such as cluster size, in order to avoid unnecessary monitoring and communication and thus optimise system performance. Thirdly, nodes are designed to recognise when changes in an environment are too frequent; too much adaptation of nodes would ultimately compromise the energy efficiency of a network.

For a WSN to self-organise, a control model based on these four characteristics should be integrated into its hardware. Firstly, the control model brings about emergent behaviour in the network by grouping nodes together into clusters based on a particular parameter. Secondly, the control model minimises energy overheads during communication by taking advantage of implicit co-ordination among sensor nodes. Thirdly, the control model encourages local interactions between nodes, rather than global control, leading to minimal storage of state information. Finally, the model controls adaptation of nodes to optimise performance by avoiding too frequent changes. These principles of self-organisation guided the development of the models in this thesis.

3.5 Limitations of Self-Organised Systems in Monitoring Applications

It has already been noted that the advantages of self-organised systems include their scalability, robustness and ability to make complex systems more manageable. However, self-organised systems have certain disadvantages which must be recognised as they could affect the usability of such systems in a monitoring environment. Firstly, moving from a centralised system of control to one that is decentralised decreases the level of determinism and predictability as summarised in Figure 3.3. This unpredictability is a problem when the need for strict quality guarantees on collected data is important. To address this problem, researchers are actively seeking ways of implementing statistical models into self-organised systems in order to limit the effects of unpredictability [LK00].

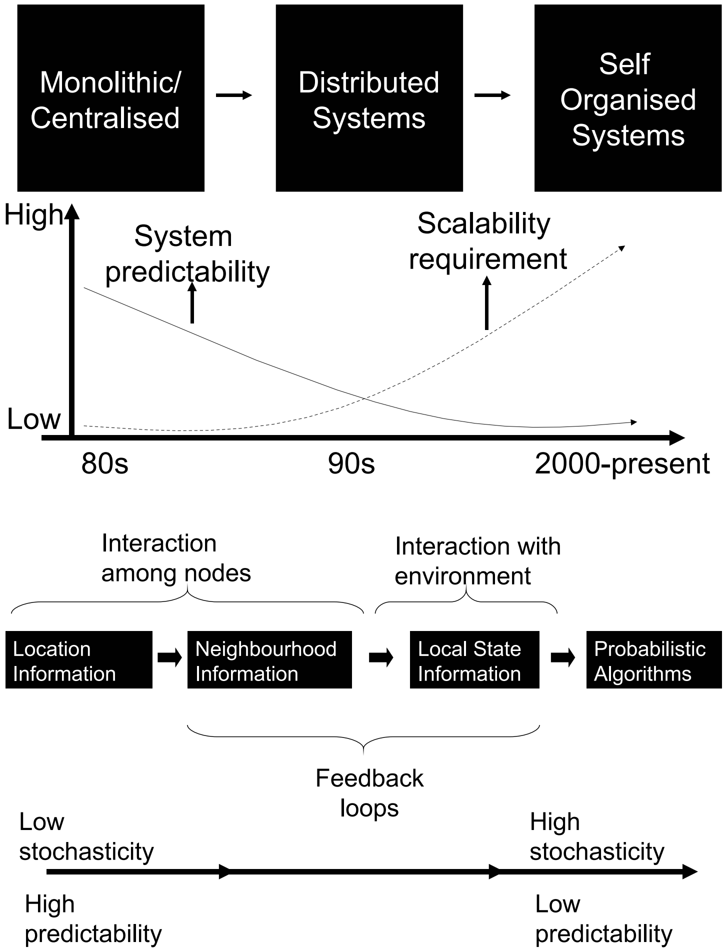


Figure 3.3: Illustration of the inherent unpredictability within self-organising systems

Another drawback of self-organised systems arises because of the lack of global state control. This means that only a local view of an environment is available from any given subsystem and therefore only suboptimal results are obtainable. In some application scenarios where the environment is highly variable, suboptimality is less of a disadvantage because optimum settings are subject to constant variation. Nevertheless, in general, rapid convergence to a satisfactory suboptimal operational point is critical in order to satisfy application requirements.

A third disadvantage of self-organising systems is that it is difficult to replicate the exact behaviour of a network while testing components in a laboratory environment because such systems are inherently random and unpredictable. The natural world is intrinsically non-linear and therefore much flexibility is needed within self-organised systems in order for them to adapt to the properties of the deployment environment.

The aforementioned limitations of self-organising systems can be largely alleviated using the framework proposed in the next section.

3.6 Framework for Self-Organised Data Collection and Dissemination

Given the decentralised nature of self-organising systems, statistical and time series protocols are required to alleviate some of the problems mentioned in Section 3.5. To support such protocols, frameworks are needed to deal with unpredictability and provide statistical quality guarantees on data collected. Section 3.6.1 describes a Dual Prediction Scheme (DPS) which can be used as a framework for supporting the efficiency requirements in self-organised systems. The DPS is incorporated into a management and control plane in order to facilitate the interoperability of adaptation, sensing and prediction operations for more efficient data collection. Simulations were carried out in order to demonstrate the effectiveness of DPSs in a self-organised WSN. Section 3.6.2 outlines the specifications used in simulations throughout this thesis.

3.6.1 Dual Prediction Scheme

Previous work [BSB07, JC04, SR06] has shown that an effective framework for providing quality guarantees is one which predicts approximate values of a reading at a base station while guaranteeing bounds on any divergence from the true value of a sensor reading. One such framework that incorporates this idea is a Dual Prediction Scheme (DPS) [BSB07]. DPSs compare a prediction $\hat{X}_t$ against real values X_t, so that any deviations are bound by a maximum error threshold e_{max}, at the base station. This is possible because each sensor node both measures and forecasts readings. For example, at time t, a forecast function $f(X_t)$ predicts reading $\hat{X}_t$ at both the sensor and base station ends. The sensor then takes an actual reading X_t. If the deviation $\hat{e}_t = \left|X_t - \hat{X}_t\right|$ is within the predetermined error threshold e_{max}, the forecast $\hat{X}_t$ is accepted as satisfactory. At the base station, $\hat{X}_t$ is used as a reading in which $\hat{X}_t \approx X_t$, and no transmission is made from the sensor node. Energy is saved because transmission to the base station only occurs when the deviation $\hat{e}_t$ at the sensor node exceeds e_{max}.

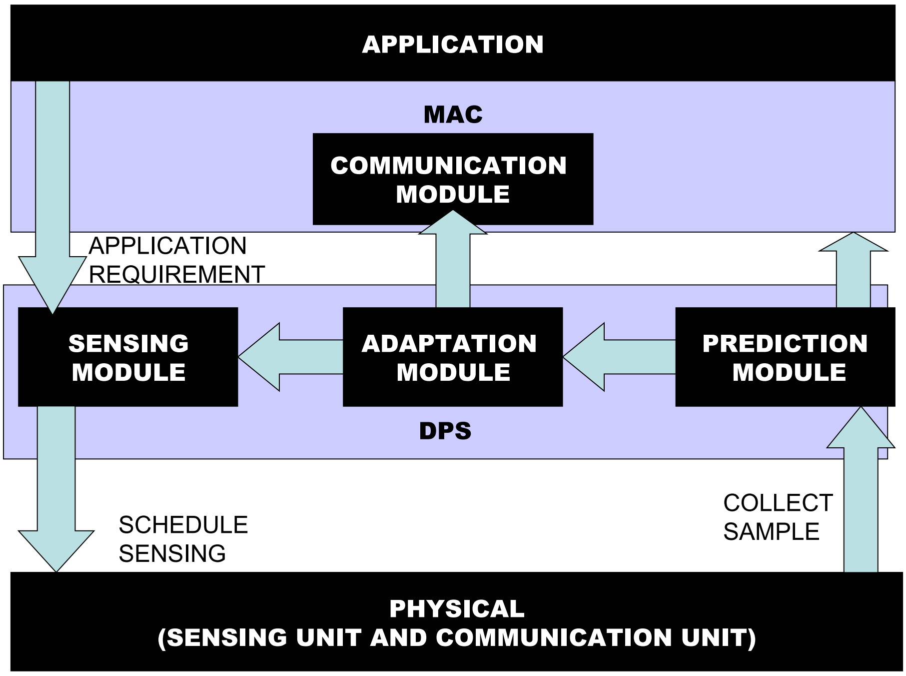


Figure 3.4: Management and control of data collection is facilitated by interoperability of three modules for adaptation, sensing and prediction

To further conserve energy, DPS can be combined with a management and control plane as shown in Figure 3.4. Figure 3.4 comprises of four layers: Application, MAC, DPS and Physical layers. Excluding the physical layer, each layer is supported by functions in an adjacent lower layer; as an illustration the MAC layer is supported by the DPS layer. This layered architecture in the management and control plane addresses two of the aforementioned limitations in self-organised systems: the lack of global state control leading to suboptimality of results; and the need for flexibility because of the difficulty in recreating the properties in a deployment environment.

The framework addresses suboptimality by introducing an application requirement into the application layer so that as the quality of data collection is traded off against energy savings at the physical layer, the overall application requirements are not compromised. The application layer must be connected to the physical layer via the sensing module in order to adjust the sleep-wake cycle of the sensing unit. Flexibility, required within self-organising systems as they adapt to the properties of a deployment environment, is achieved using an adaptation module. Readings collected from a sensing unit are delivered to the prediction module which uses the data to determine forecasts. Any deviations between forecasts and true readings are registered and processed by the adaptation module. The adaptation module also uses the event occurrence rate to tune a system's responses according to the monitoring environment in order to minimise the number of adjustments made by a sensor node.

The adaptation module is half of the interface between the MAC layer and the DPS layer. The other half is comprised of the communication module which schedules transmission/reception in the communication unit. The interoperability of the modules for prediction, adaptation and sensing are described in further detail in Chapter 4 while Chapter 5 presents more details on the communication module.

3.6.2 Self-Organised Wireless Sensor Network: System Specifications

Owing to the large variety of applications in which wireless sensor networks are used, the specifications in self-organised systems can vary widely. This means that testing and evaluating DPSs in self-organised WSNs may be difficult because of the variety between systems. For the purposes of this study, particular characteristics of self-organised WSNs are assumed. This allows for the development of simulations which evaluate the parameters most relevant to this thesis. It is important to note that after small alterations these specifications could be easily adapted to serve a wider variety of sensor-based applications.

In order to develop simulations that demonstrate the effectiveness of DPSs in a self-organised WSN, the following specifications are used:

- Short range communication units: Sensor nodes use a common short range communication frequency to communicate with each other using an omnidirectional antenna. All sensor node communication units have identical communication ranges and require a limited bandwidth for either transmission or reception.

- First order radio model: A first order radio model of the type in [HCB00, YWZ06] is used to determine the energy required to both transmit and receive data packets. More specifically, $\varepsilon_{elec} = 50nJ/bit$ and $\epsilon_{amp} = 100pJ/bit/m^2$ denote the energy consumption requirements for the electronic and amplification components of the communication unit respectively in a sensor node. Each sensor node uses $E_{Rx}(k) = \varepsilon_{elec} \cdot k$ Joules of energy for reception and $E_{Tx}(k, d) = \varepsilon_{elec} \cdot k + \epsilon_{amp} \cdot k \cdot d^2$ for transmission of a k bit packet over a distance of d metres.

- Restricted inter-node communications: In order to alleviate the effects of the Broadcast Storm problem, previously mentioned in Section 2.3.3.1 of Chapter 2, the communication range is restricted so that data can only be broadcast to nodes that are within one hop from the transmitting node. This limits the inter-node communication distance to a maximum of one hop in each sleep-wake cycle of a sensor node. Although this restriction increases the delay in event reporting, because nodes can only communicate with their immediate neighbours, *Wang et al.* illustrated that such an approach was an effective solution to the Broadcast Storm problem [Wan03].

- Active sensor state: When a node is active, its processor and RF components are also active. It is assumed that in each active period, a node's receiver is always operational to receive data. It is also assumed that data is stored in a transport buffer with a large capacity. At the end of an active period, data from a node's transport buffer is transmitted in a first-in-first-out (FIFO) schedule and then deleted from the transmitting node. Furthermore, within an active period, a node may add its own data packets to the transport buffer for transmission. It should be emphasised that while transmission and reception only occur in active states, these functions are decoupled and thus do not have to occur simultaneously. For example a node's receiver can be active while its transmitter is asleep. This may occur when a node's transport buffer is empty.

- Sleep sensor state: In a sleep state, it is assumed that all components of the sensor including the processor, radio and measurement sensor are switched off except for a wake-up timer which consumes a negligible amount of energy [GM04].

- Multiple sensors: Sensor nodes have multiple sensor units and each sensor unit can be turned on or off independently from other sensor units.

- Random node placement: Unless otherwise stated, sensor nodes are deployed randomly and each deployment forms a connected network.

- Limited or no node mobility: nodes either have limited movement or are immobile. Therefore the location of each node either remains constant or does not change to the extent of impairing connectivity between nodes.

- No obstacles: An outdoor deployment area with no obstacles in the event detection area is assumed.

These specifications allow a sensor network to be simulated using a 2-dimensional Unit Disk Graph (UDG) [LWF03]. A UDG is a special form of a geometric random graph which is useful for simulating the random nature of sensor node's deployment. In UDGs two nodes are said to be connected when they are located within a specified communication range from each other.

In comparison with real-life sensor network deployments, UDG models would appear simple but they are effective nonetheless [LWF03]. Furthermore, UDG models can be reinforced with path loss models of the type proposed in [Rap02] in order to incorporate real-life characteristics of an environment into a simulation.

3.7 Chapter Summary

In this chapter the fundamental principles of self-organised systems were discussed and the advantages of such systems, including robustness, scalability, flexibility of deployment and the absence of central control, were outlined. These advantages make self-organising systems desirable for use in environmental monitoring, as well as in other application areas, because they can be deployed in a wide variety of deployment environments and they allow systems to function autonomously while reducing communication overheads when compared with centrally controlled or distributed systems.

The limitations in self-organising systems were also discussed including the unpredictability of such systems and the lack of global state. Because self-organising systems are inherently unpredictable, the quality of the data collection process cannot be guaranteed; it also means that it is difficult to replicate the exact behaviour of a network while testing components. Furthermore, the lack of global state means that only suboptimal results can be obtained. A Dual Prediction Scheme was then outlined which can provide guarantees on the quality of the data collection process. It was noted that a DPS could be incorporated into a management and control framework in order to address the aforementioned disadvantages and thus make self-organising WSNs a more reliable and effective tool for data collection and dissemination. This framework serves as a basis for the models proposed in Chapter 4 and Chapter 5 of this study.

Chapter 4

Dual Prediction and Probabilistic Scheduler

4.1 Introduction

Energy efficient data collection protocols are required in order to improve the energy efficiency and processing capabilities of sensor networks. Efficient management of a sensor node's communication unit has traditionally assumed critical significance because communication is energy intensive. However, specialised sensors exist which, over time, can consume more energy than the communication unit. This chapter proposes a Dual Prediction and Probabilistic Scheduler (DPPS) to be used for event detection, with improvement of energy efficiency as a central purpose. By combining Compression and Load Balancing techniques in a Dual Prediction Scheme, DPPS monitors event data more efficiently in comparison to previous protocols; energy is conserved in the sensing unit while stronger quality guarantees of the data monitoring and collection process are provided.

4.2 Motivation

Data collection frameworks used in wireless sensor networks can be applied to a plethora of different environments including habitat monitoring [SCV+06], target tracking [AKP08], and monitoring buildings [DGM05]. In these diverse deployments, energy consumption is highlighted as a critical drawback because when a sensor node's limited battery supply has been completely discharged, replacements in such environments is expensive or impossible.

The most common view expressed within the research community is that it is most advantageous to limit energy consumption in the communication unit of a sensor node as it is thought that the communication unit accounts for the highest proportion of energy consumption [PK00]. Although this is the case in some types of sensor nodes, specialised sensing units exist, such as airflow sensors, pressure sensors and accelerometers which, over time consume equal or more energy than a communication unit [hon08]. For example XBow's Heading Sensor which measures azimuth angles consumes $375mW$ of power during sensing compared with the $60mW$ used for transmitting in MICA2 nodes [LCS05]. Effective management of the sensing unit is therefore essential because it can decrease the energy consumption of a sensor node and thus increase the lifetime of a network.

The most popular method of managing a sensing unit is through the use of Scheduling techniques. Scheduling involves switching off the sensing unit between measurements, thus saving energy. However switching off the sensing unit also potentially increases the number of missed events and false alarms thus compromising the quality of event detection.

eSENSE, a classic sensor unit scheduler, trades off energy consumption in the sensing unit with an application's underlying data quality requirements. eSENSE used an average wake-up rate to save energy while simultaneously satisfying an application's quality constraint. The disadvantage, however, of using a system with an average wake-up rate is that this type of constraint introduces some ambiguity into the collection process; two or more sampled signals could satisfy the same data quality constraint but have significantly varied mean square errors in comparison to the actual data. In applications where high precision in monitoring is essential, the wake-up rate of sensors may have to be increased to the extent that any energy savings are negligible.

The problem of ensuring the quality of results while minimising energy consumption can be addressed using the Dual Prediction and Probabilistic Scheduler (DPPS) developed in this chapter [EY09]. Rather than using an average wake-up rate, DPPS combines a Dual Prediction Scheme (DPS), previously presented in Chapter 3, with a combination of Compression and Load Balancing techniques. This synthesis of techniques facilitates the collection of event data in an energy efficient manner while providing bounded guarantees on the mean square error of the reconstructed sample data thus providing stronger guarantees on the quality of collected data and reducing the number of false alarms and missed events.

DPPS was developed using the eSENSE framework as a foundation. The fundamental framework of eSENSE was adapted to incorporate a Dual Prediction Scheme and a mean square error constraint. The mathematical formulation of the sensing efficiency problem is outlined below in Section 4.3.

4.3 Problem Formulation

Table 4.1: Notation of parameters used in DPPS

Parameter	Definition (Value)
Generic Parameters	
X_t	Real time series data
$\hat{X}_t$	Predicted time series data
N	Total number of data samples
δN	Baseline sampling interval
e_{pred}	Prediction error: e_{pred} is the difference between real and predicted time series data
DPPS Parameters	
e_{max}	Event threshold: e_{max} is the prediction error threshold that leads to a state change
E^2_{max}	Mean square error requirement
p_i	(Adjusted) probability of sensing a state change: p_i is the probability of sensing an event at time i
p^*_i	Unadjusted probability of sensing a state change
q_i	Probability of a state change occurring: q_i is probability of sensing an event at time i
f_p	Calculated false positive
f_n	Calculated false negative
F_p	False positive requirement
F_n	False negative requirement
S_i	Event sample at time i
k	Sampling interval of a sensing unit
θ_k	IMA prediction co-efficient at sampling interval k
D_{max}	Maximum sampling interval of a sensing unit

4.3.1 System Model

Consider a system of sensor nodes which measure and collect data using a total number of samples N at a particular temporal resolution δN. The temporal resolution defines the granularity of changes that can be detected. Indicated in Figure 4.1, δN is the baseline sampling interval which represents the smallest resolution for event detection. Detection at δN corresponds to the highest sampling frequency and hence the most accurate digitised approximation of the data being sampled, X_t. Sensor nodes are assumed to be in one of two states: active or sleeping. When a sensor node is active, the sensing unit, processing unit and radio are active. This allows the sensor node to measure samples and compute predictions. Active sensor nodes also calculate *prediction error*, e_{pred}, which is the difference between an actual measurement and the predicted measurement. Although a sensor node may be active and ready to receive an incoming packet, the transmitter may be idle. This is because the transmitter only becomes active when there is data to send. This occurs in two different cases: *send-on-requirement* and *send-on-sample*. In the case of *send-on-requirement*, collected data is discarded if its equivalent estimate at the base station is accurate. This saves energy because it means the transmitter in the communication unit can be off for longer. Conversely, in a *send-on-sample* policy, sensing and transmission are combined so that nodes transmit all collected data without recourse to suppression. When the sensor node is sleeping, the sensing, processing and communication units are all off.

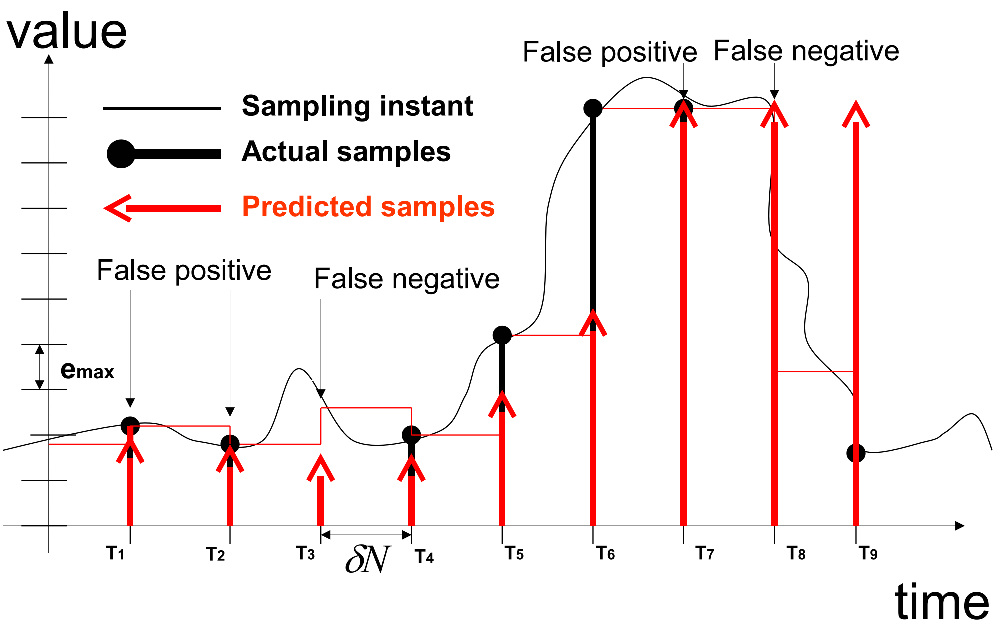


Figure 4.1: Prediction, false negatives and false positives

As illustrated in Figure 4.1, a state change occurs when the *prediction error* exceeds the event threshold, e_{max}, signifying an event of interest to an application. Because they occur randomly, state changes may be missed when a sensor node's sensing unit is switched off. Such missed state changes are referred to as missed events or false negatives as shown at T_3 and T_8 in Figure 4.1. Conversely any unnecessary measurement by a sensor when no state change has occurred is defined as a false alarm or false positive because a sample was measured in the false expectation that an event would be detected as illustrated at T_2 and T_7 in Figure 4.1.

The false negative ratio, f_n, is defined as the number of false negatives, n_f, divided by the length of the baseline sequence, $\sum \delta N$: $f_n = \frac{n_f}{\sum \delta N}$. Similarly, the false positive ratio, f_p, refers to the number of false positives, n_p, divided by the length of a baseline sequence: $f_p = \frac{n_p}{\sum \delta N}$. An increase in the false negative ratio is symptomatic of an increase in the number of missed events and therefore a loss in data quality. Conversely an increase in the false positive ratio indicates that data is being sampled at a rate beyond the requirements of the application and therefore energy is being wasted unnecessarily.

4.3.2 Objectives

The objective of DPPS is to minimise the total energy required to measure events whilst providing statistical guarantees on the quality of data collected. This is done by minimising the chance of a sensor being active when no relevant event occurs. At each detection point over time N, let vectors $\bar{p} = [p_1, \ldots, p_N]$ represent the probability of the sensor being active and $\bar{q} = [q_1, \ldots, q_N]$, the probability that a state change occurs. Thus p_i represents the probability of sensing an event at a sampling instant i and q_i represents the probability of an event occurring at a sampling instant i. Where ν is a positive constant that determines the total average energy $\bar{E}$ used, the optimisation problem of DPPS becomes:

$$\underset{p_i}{Min}\,\bar{E} = \nu \sum_{i=1}^{N} p_i : \tag{4.1}$$

$$\frac{\sum_{i=1}^{N}(1-p_i)q_i}{N} \leq F_N \tag{4.2}$$

$$\frac{\sum_{i=1}^{N}(X_i(p_i)-\hat{X}_i(p_i))^2}{N} \leq E_{max}^2 \tag{4.3}$$

Although it is impossible that all events are captured, the inequality in Equation 4.2 provides a statistical guarantee on the quality of data collected by limiting the expected miss ratio to within a tolerance level F_N. Similarly the inequality in Equation 4.3 limits inaccuracy in terms of the mean square error constraint to within E_{max}^2.

4.4 Event Detection

4.4.1 Sensing Probability

In order to adequately measure events it is necessary to set a minimum bound on the sensing probability. If p_i^* defines the unadjusted sensing probability at time i, the calculation of p_i^* is achieved by considering what value is required for optimality. For example, $p_i^* = 1$ is the maximum sensing probability which minimises the number of missed events. However using this value would require that the sensing unit is active all the time thus consuming high amounts of energy. p_i^* would therefore need to be continually adjusted in order to enhance energy savings. Given that the miss ratio is defined by the number of missed events divided by the size of the baseline sequence, and assuming that events occur randomly, it follows that $\bar{p}$ and $\bar{q}$ are independent and that the missed ratio at i is defined by $(1-p_i^*)q_i$. If however $\bar{p}$ and $\bar{q}$ were correlated as in [LCS06], then it follows that the miss ratio, f_{n_i}, at detection point i, should be smaller in comparison to when $\bar{p}$ and $\bar{q}$ were independent and uncorrelated i.e. $f_{n_i} \leq (1-p_i^*)q_i$. This prevents any individual miss ratio deviating above the upper bound F_N. Stronger quality guarantees are provided because f_{n_i} is required at each sampling point thus satisfying the tolerance constraint. As inspired by [LCS06], the L_∞ miss ratio bound on p_i^* becomes:

$$p_i^* = \begin{cases} 0 & 0 < q_i < F_N \\ 1 - \frac{F_N}{q_i} & F_N < q_i \leq 1 \end{cases}$$

The adjusted sensing probability at point i, p_i, is calculated by considering the false positive rate f_p. f_p is approximated from a FIFO (first-in-first-out) queue of length W where $f_p = \frac{n_p}{\sum \delta N} \approx \frac{n_p}{W}$. As df_p/dW increases, it indicates that the sampling rate is high in relation to the data quality requirements of the application. Hence p_i^* should be decreased so that the sensor spends more time asleep. Alternatively as events occur more often, df_p/dW decreases and p_i^* should be increased so that a sensor node is active more often in order to catch potential events. Thus it follows that p_i can be determined from p_i^* using:

$$p_i = \begin{cases} p_i^* - \zeta & \text{if } \frac{n_p}{W} \geq F_p \\ \eta \, p_i^* & \text{if } \frac{n_p}{W} < F_p \end{cases}$$

where $|\eta| > 1$ and $|\zeta| < 1$ are positive values with F_p being a constant denoting the false positive threshold. By decreasing p_i^* linearly and increasing p_i^* non-linearly, events are captured more aggressively. During simulations, the results of which are shown later in this chapter, the parameters $\eta = 1.1, \zeta = 0.1, W = 50$ and $F_p = 0.8$, were used because of the superior energy savings at these settings. The next subsection discusses how to estimate the probability of detecting an event using historical data.

4.4.2 Event Detection Probability

At both sensor nodes and the base station, a data stream prediction model uses historical data to forecast sensor readings. While other models can be used for prediction purposes, a first order Integrated Moving Average (IMA) model was selected for DPPS; diagnostic checks were carried out using autocorrelation and partial autocorrelation functions from a set of training data (see Appendix A for details). IMA models are widely used for modelling non-stationary time series data because they offer low computational and memory overhead which allows easy practical implementation [BK92]. Asumming $\{X_1, X_2, X_3...\}$ represent the real data stream sequence of sensor node and $\{\hat{X}_1, \hat{X}_2, \hat{X}_3...\}$ represent the predicted data stream sequence of a sensor node, the IMA prediction model used in DPPS is as follows:

$$\hat{X}_i = X_{i-k} + \theta_k e_{i-k} \tag{4.4}$$

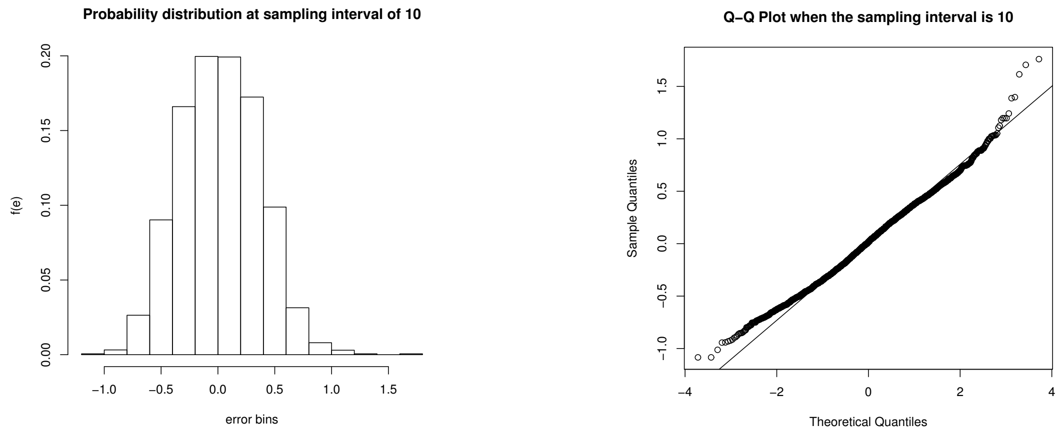
In Equation 4.4, θ_k is the IMA coefficient used for a k step ahead predictor. By definition the *k-step ahead predictor* is the same as the sampling interval and thus henceforth both are used synonymously. Equation 4.4 may be represented more generally by:

$$\hat{X}_{i+k} = X_i + noise \tag{4.5}$$

Let the *noise* term be described by Q_k; Q_k is a distribution that represents the probability of changes in prediction error. For clarity we introduce subscript k which indicates that each sampling interval has an associated error distribution. Q_k is formed from the differences between the actual and predicted samples in a datastream sequence and hence may be expressed as:

$$Q_k = \{X_{i+k} - \hat{X}_{i+k} \quad \forall i | k \leq D_{max}\} \tag{4.6}$$

If the length of the sequence of differences forming Q_k is large (typically greater than 50), then it is expected that Q_k becomes normally distributed in accordance with the central limit theorem. Thus $q_i \in Q_k$ and $q_i \sim N(\mu_k, \sigma_k^2)$. Take Figure 4.2 as an example where the prediction error at an interval $k = 10$ is outlined. At this sampling interval, the errors follow normal distributions as seen in Figure 4.2(a) and is further confirmed by the Q-Q plot in Figure 4.2(b).



(a) Probability distribution of error at sampling interval $k = 10$

(b) QQ Plot comparing the distribution of errors to a Gaussian normal distribution

Figure 4.2: Error distribution and the Q-Q plot at $k = 10$

If e_{pred} is a random variable which denotes the size of the prediction error, then it follows that the probability of state change in the prediction error is given by:

$$\mathbb{P}[e_{pred}(i) > e_{max}] = q_i \sim N(\mu_k, \sigma_k^2) = \frac{1}{\sqrt{2\pi}\,\sigma} \int_{e_{max}}^{\infty} e^{\frac{-(e_{pred}-\mu_k)^2}{2\sigma_k^2}} \, de_{pred} \qquad (4.7)$$

Equation 4.7 shows the probability of the prediction error at a particular sampling interval k; the state change probability, q, can be calculated using a combination of μ and σ^2. In addition, for a particular lead time the normal distribution can be transformed into a standard normal distribution using $z = \{\frac{x-\mu}{\sigma}\}$, as shown in Figure 4.3.

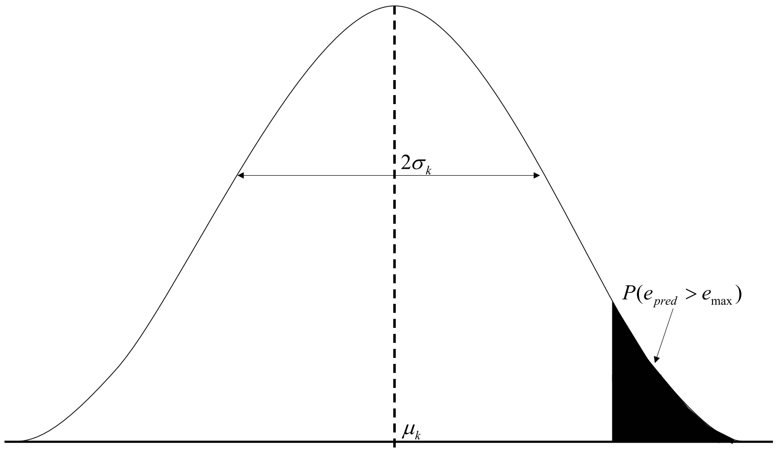


Figure 4.3: The Gaussian distribution for mean μ and standard deviation σ; the shaded area is the probability of observing a state change greater than e_{max}

4.4.3 Mean Square Error Accuracy Constraint

In order for DPPS to satisfy an accuracy constraint, the requirement of Equation 4.3 is that the mean square error is bounded below e_{max}. Furthermore, satisfying an L_∞ bound on the constraint as shown in Equation 4.8 would constitute an even stricter condition. The goal of this section is to answer the following: What maximum sampling interval, D_{max}, is required in order to satisfy an application's mean square error constraint? Let us begin the analysis by assuming the following to be strictly true:

$$\left| X_i - \hat{X}_i \right| < E_{max} \forall i \in N \tag{4.8}$$

Squaring both sides of Equation 4.8 and substituting for the forecasting function gives:

$$\begin{aligned}
\left(X_i - \hat{X}_i \right)^2 &< E_{max}^2 \\
\left(X_i - X_{i-k} + \theta_k e_{i-k} \right)^2 &< E_{max}^2 \\
\left(X_i - \hat{X}_{i-k} \right)^2 + 2 \left(X_i - X_{i-k} \right) \left(\theta_k e_{i-k} \right) + \left(\theta_k^2 e_{i-k}^2 \right) &< E_{max}^2
\end{aligned}$$

Taking the expectation of both sides gives:

$$E\left[(X_i - \hat{X}_{i-k})^2\right] + 2\theta_k E\left[(X_i - X_{i-k})\right] E\left[e_{i-k}\right] + \theta_k^2 E\left[e_{i-k}^2\right] < E_{max}^2$$

It can be assumed that $E(e_{i-k}) = 0$ (see Figure 4.2(a)) $\therefore$

$$E\left[(X_i - \hat{X}_{i-k})^2\right] + \theta_k^2 E\left[e_{i-k}^2\right] < E_{max}^2$$

In [BJ70], it is shown that $E[(X_i - X_{i-k})^2] = \gamma_0 + \gamma_1$ where:

$$\begin{aligned} \gamma_0 &= \left((1+\theta_1)^2 + (k-1)(1-\theta_1)^2\right)\sigma_1^2 \\ \gamma_1 &= -\theta_1\sigma_1^2 \end{aligned} \tag{4.9}$$

Substituting γ_0 and γ_1 into $E[(X_i - X_{i-k})^2]$ above gives:

$$\begin{aligned} \gamma_0 + \gamma_1 + \theta_k^2\sigma_k^2 &< E_{max}^2 \\ \left((1+\theta_1)^2 + (k-1)(1-\theta_1)^2\right)\sigma_1^2 - (\theta_1\sigma_1^2) + \theta_k^2\sigma_k^2 &< E_{max}^2 \end{aligned}$$

The above may be rewritten as:

$$\begin{aligned} \sigma_1^2\,k - 2\theta_1\sigma_1^2\,k + 2\theta_1\sigma_1^2 - \theta_1^2\sigma_1^2\,k - \theta_1\sigma_1^2 + \theta_k^2\sigma_k^2 &< E_{max}^2 \\ k\left(\sigma_1^2 - 2\theta_1\sigma_1^2 - \theta_1^2\sigma_1^2\right) &< E_{max}^2 - \theta_1\sigma_1^2 - \theta_k^2\sigma_k^2 \end{aligned}$$

Rearranging for k gives:

$$\lfloor k \rfloor < \frac{E_{max}^2 - \theta_1\sigma_1^2 - \theta_k^2\sigma_k^2}{(\sigma_1^2 - 2\theta_1\sigma_1^2 - \theta_1^2\sigma_1^2)} \tag{4.10}$$

Therefore $D_{max} = \lfloor k \rfloor$ should be the maximum sleep interval if the precision constraint in Equation 4.3 is to be satisfied.

4.5 Overview of DPPS

The operation of DPPS can be divided into three stages: *initialisation*, *sensing-adaptation* and *prediction*. During *initialisation* the parameters required in sensor nodes are initialised. Next, nodes enter a *sensing-adaptation* stage in which data collected by a sensing unit is used for the adaptation of the sensing unit's sleep wake cycle. When a sensor node is asleep, the value of readings that would have been measured are predicted; this occurs in the *prediction* stage. A structural overview of the *initialisation*, *sensing-adaptation* and *prediction* stages are shown in Figure 4.4.

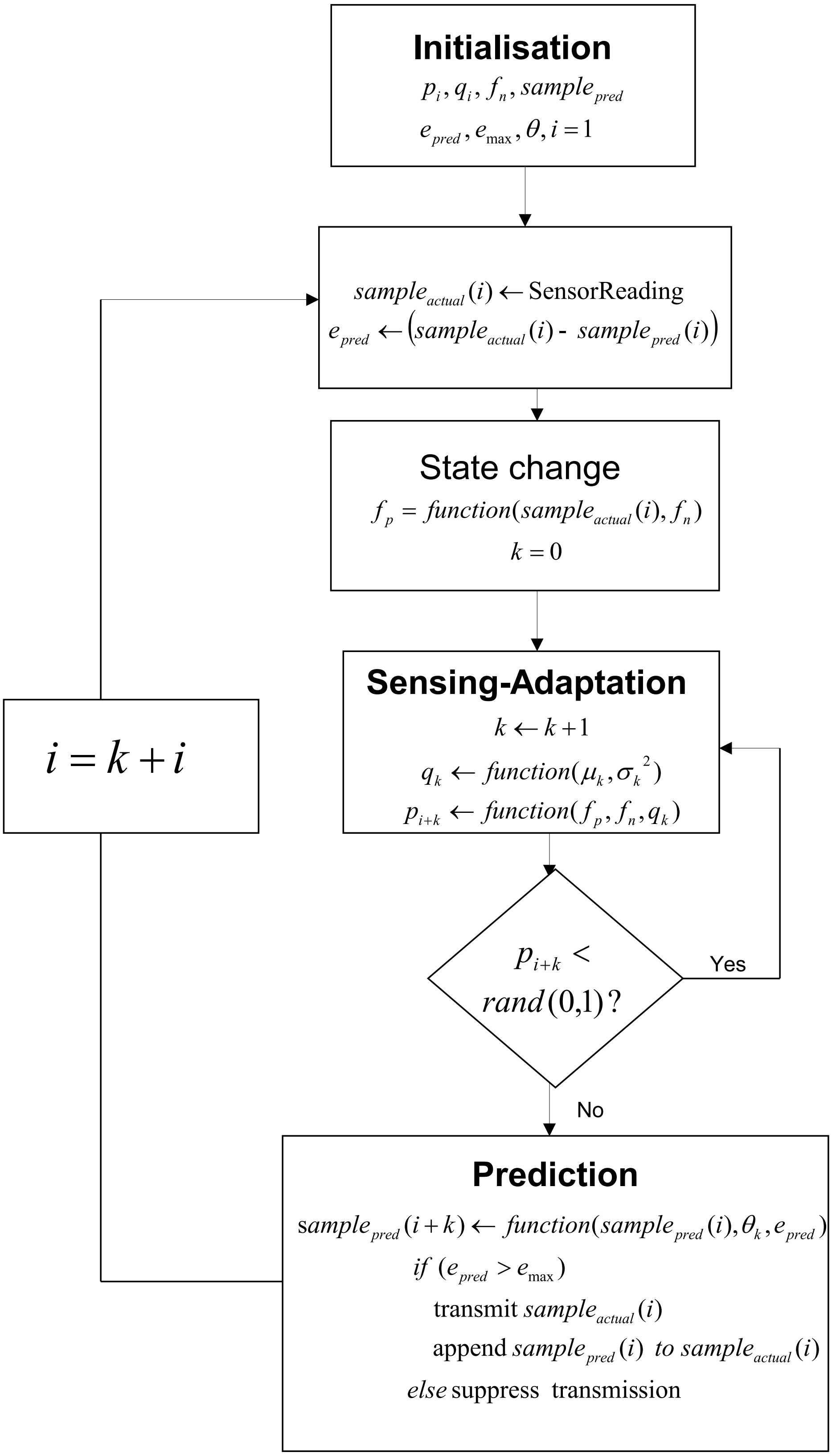


Figure 4.4: DPPS structural overview

The pseudo-code for the *initialisation*, *sensing-adaptation* and *prediction* stages are presented in three separate algorithms. Algorithm 1, outlines details of the *initialisation* stage where the real time series X and the application's maximum delay D_{max} are used as inputs; D_{max} was calculated using Equation 4.10. X was sampled at an interval k then the *first-order difference* of this series was used to calculate the moving average coefficient required in the prediction model of Equation 4.4 (see lines 2-4 of Algorithm 1). Using the prediction model, the predicted time series, $\hat{X}$, was generated and compared with the real time series, X. The difference between X and $\hat{X}$ gives the prediction error. Over time, prediction errors form a time series from which a mean, μ_k, and standard deviation, σ_k, can be obtained (see lines 9-10 of Algorithm 1).

input : time series $X = \{X_1, X_2, \ldots, X_M\}$
Maximum delay D_{max}
output: Model Parameter Sequence(MPS)
MPS=$(\theta_1, \mu_1, \sigma_1), \ldots, (\theta_{D_{max}}, \mu_{D_{max}}, \sigma_{D_{max}})$

1 **for** $k \leftarrow 1$ **to** D_{max} **do**
2     $X \leftarrow$`seq`(*from=1,to=M,by=k*);
3     $X \leftarrow$`diff`(*X,lag=k*);
4     $\theta_k \leftarrow$`MovingAvg`(X);
5   **for** $j \leftarrow 1$ **to** M **do**
6         $\hat{X}_j \leftarrow$`Forecast`(θ,*interval* = k);
7   **end**
8     $\hat{e}_{pred} \leftarrow$`diff`($X$,$\hat{X}$);
9     $\mu_k \leftarrow$`meanCalc`($\hat{e}_{pred}$);
10     $\sigma_k \leftarrow$`stdCalc`($\hat{e}_{pred}$);
11 **end**

Algorithm 1: Initialisation Stage

Algorithm 2 describes the adaptation of the sampling rate in the sensing unit during the *sensing-adaptation* stage. This adaptation begins when the sensing unit measures a reading. Following a measurement, the value of the reading is examined for false positivity by calculating f_p using the method discussed in Section 4.4.1. Lines 5-12 of Algorithm 2 are used to determine the sampling interval k; assuming p_i is less than a generated random number between 0-1, k is incremented after every cycle in the *If loop*. In each cycle, q_i and p_i are calculated using the methods outlined in Sections 4.4.1-4.4.2. The maximum sampling interval used is D_{max} in order to maintain the inequality constraint shown in Equation 4.3. Therefore the maximum value of k is limited to D_{max} as shown in lines 10-11.

input : initialise i, k, $sample_{pred}$, e_{pred}, e_{max}, F_N,f_p
$\theta = \{\theta_1, \ldots, \theta_{D_{max}}\}$
$\mu = \{\mu_1, \ldots, \mu_{D_{max}}\}$
output: k, p_i, q_i

1 $k = 0$;
2 $p_i = 0$;
3 $sample_{actual}$ ←take sensor reading;
4 f_p ←`adaptSchedule(`$sample_{actual}$,F_N`);`
5 **if** p_i `<randUniform` **then**
6   $k \leftarrow k + 1$;
7   q_i ←`senseEventStreamProbability(`μ_k,σ_k`);`
8   $p_i \leftarrow$ `stateEventDetectionPrediction(`f_p, F_N, q_i`);`
9   $i \leftarrow i + 1$;
10 **if** $k > D_{max}$ **then**
11   $k \leftarrow D_{max}$;
12 return k

Algorithm 2: Sensing-Adaptation Stage

At the *prediction* stage, illustrated in Algorithm 3, the prediction model determined during *intialisation* is used for forecasting as well as determining which real sensor readings should be transmitted in order to enhance data quality. Data quality is preserved while energy is saved because readings are transmitted only when the prediction error e_{pred} is above the error threshold e_{max} (see lines 2-3 Algorithm 3). Transmission of readings below the error threshold are suppressed. At line 4 of Algorithm 3, the false alarm rate is updated. This false alarm rate accelerates or decelerates adaptation so that the efficiency of data collection is improved. The combination of Algorithms 1-3 creates the Dual Prediction and Probabilistic Scheduler.

input : i, k, e_{max}, $sample_{pred}$, $sample_{actual}$,f_p
output: `transmit(`*false*`)` or `transmit(`*true*`)`

1 $sample_{pred}$ ←`Forecast(`θ,$interval = k$`);`
e_{pred} ←`diff(`$sample_{actual}$,$sample_{pred}$`);`
2 **if** $e_{pred} > e_{max}$ **then**
3   `transmit(`*true*`);`
4   `updateFalsePositive(`f_p`);`
5 **else**
6   `transmit(`*false*`);`

Algorithm 3: Prediction Stage

4.6 DPPS Simulation Setup

The simulations in this chapter were carried out using real world soil moisture datasets; these datasets corresponded to several weeks worth of soil moisture data each sampled at 30 second intervals. Data were collected at the Ecole Polytechnique Fédérale de Lausanne campus during March 2007 as part of the Sensorscope project, an environmental monitoring project in Switzerland [sen07].

For model construction purposes, the first 10,000 data points of a dataset were used as a training sequence and the remaining data were used for verification of the performance of the model. During the initialisation stage the coefficients required in the IMA prediction model described in Equation 4.4 were calculated offline using the training sequence. At a sampling interval k, the corresponding IMA co-efficient θ_k, was calculated. Table 4.2 shows θ_k, μ_k and σ_k for various sampling intervals ranging from 1 to 10.

Table 4.2: Parameters for DPPS as calculated from the training data sequence

k	θ	μ	σ
1	-0.8661	-0.00029257	0.38787
2	-0.7963	-0.00058245	0.37261
3	-0.7483	-0.004035	0.36936
4	-0.7185	-0.014145	0.35915
5	-0.6568	-0.014436	0.35389
6	-0.6413	-0.0041954	0.34782
7	-0.5963	-0.019497	0.34596
8	-0.5863	-0.035925	0.34226
9	-0.5253	-0.013804	0.33866
10	-0.4850	-0.011083	0.33728

Section 4.7 evaluates the performance of DPPS against eSENSE using the quality metrics of miss ratio, usage percentage, transmission percentage and sampling efficiency. The results of these simulations, evaluated using the data analysis package MATLAB,

were obtained by varying e_{max} at a constant miss ratio threshold F_N. Each simulation run was repeated 50 times to reduce any effects of pseudo randomness [BJ84] and the mean along with associated error bars were plotted in all results that follow.

4.7 DPPS Results and Analysis

In this section the performance of DPPS is compared with eSENSE and CM (see Chapter 2 for more details on eSENSE and CM protocols).

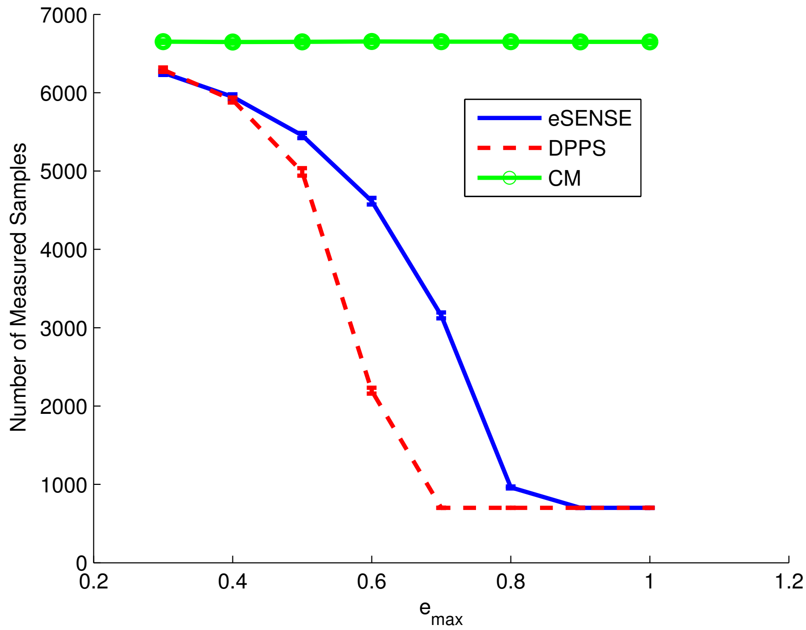


Figure 4.5: Number of measurements using DPPS, eSENSE and CM ($F_N = 5\%$)

A simulation was done to establish the relationship between e_{max} and the sampling rate for each of the three protocols. Figure 4.5 illustrates that DPPS collects a smaller number of samples than eSENSE over the same duration. This means that more energy is saved in the sensing unit using DPPS thus enabling increased battery life for sensor nodes in a network.

The next simulations were conducted to determine the usage percentages of the sensing units. As discussed previously in Section 4.3, a send-on-sample policy transmits all collected data without recourse to suppression and therefore is synonymous with usage percentage. The usage percentage is thus an indicator of the level of sensor sampling. The results of these simulations, shown in Figure 4.6, reveal that high usage percentages are connected with high sampling rates.

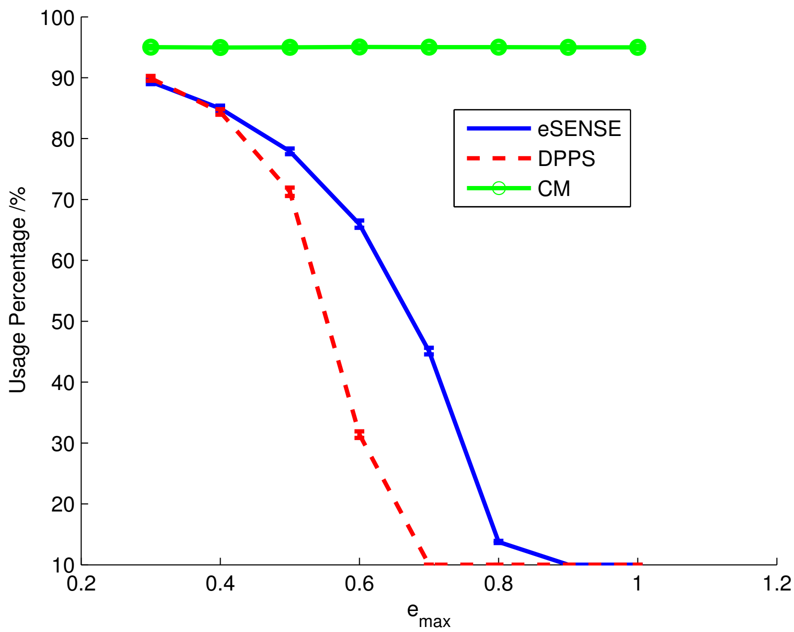


Figure 4.6: Usage percentage of DPPS, eSENSE and CM ($F_N = 5\%$)

In order to guarantee a miss ratio of 5%, CM must send 95% of all sampled data. Therefore CM collects the highest number of samples and consequently has the highest sensor usage percentage in comparison to DPPS and eSENSE. Figure 4.6 reveals that DPPS reduces the usage percentage by up to 85% compared with CM, and by up to 35% compared with eSENSE.

The send-on-requirement policy is synonymous with the transmission percentage. Figure 4.7 shows the results of simulations done to examine the transmission percentage of the three protocols. Figure 4.7 shows that DPPS has a higher transmission percentage than eSENSE indicating that a higher proportion of collected measurements contain relevant data. On the other hand, CM has the highest transmission percentage because, unlike DPPS and eSENSE, all samples in CM are transmitted regardless of their relevance, with no distinction between event and non-event sampling.

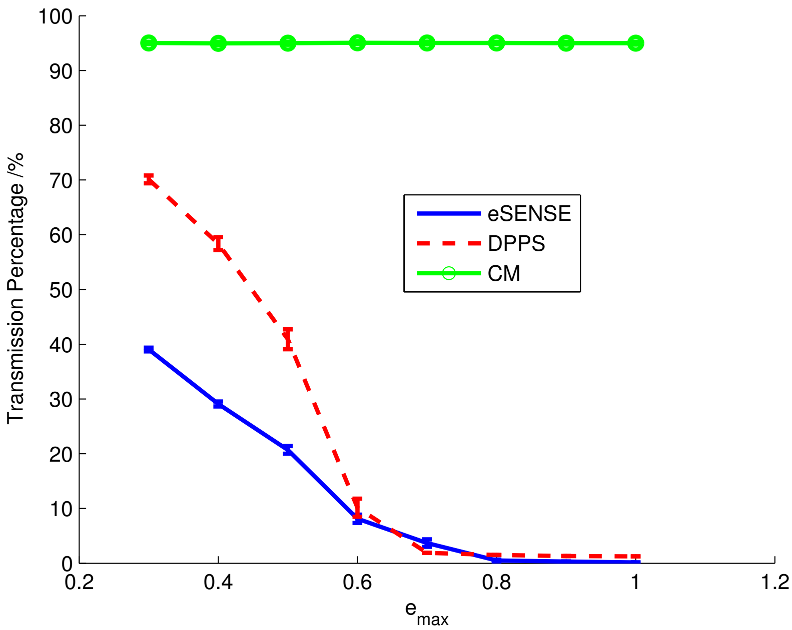
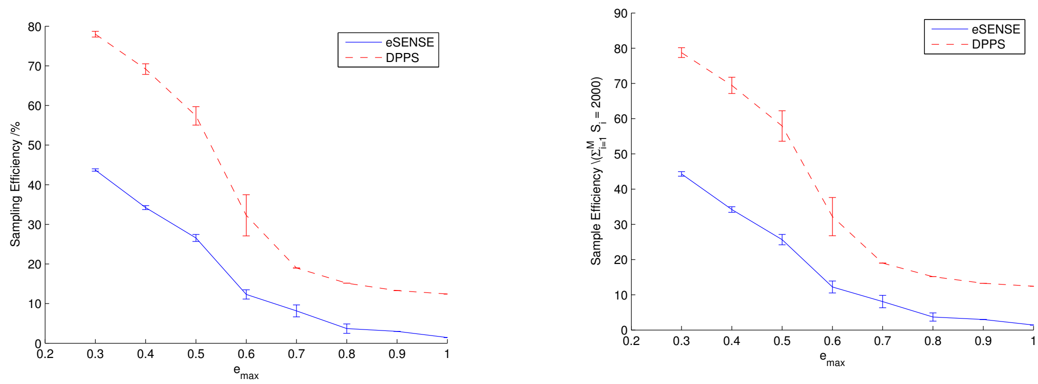


Figure 4.7: Transmission percentage of DPPS, eSENSE and CM ($F_N = 5\%$)

As expected, the usage and transmission percentages decrease in DPPS and eSENSE as e_{max} increases because increasing e_{max} also reduces the likelihood that all events will be measured. Another explanation of the decreased usage and transmission percentages is that as e_{max} increases and thus the error threshold increases, the conditions required for event capture are less strict. Therefore it is less likely that measurements will be taken which exceed the error threshold and thus transmissions are reduced.

The next set of simulations carried out examine the sampling efficiencies of the protocols in two different cases; firstly where the total time N is constant, and secondly where the total number of samples $\sum_{i=1}^{N} S_i$ is constant. In both cases e_{max} was varied and the sampling efficiency was calculated. Since CM does not distinguish between event and non-event sampling, the sampling efficiency using CM is trivially 0% and is therefore not considered.



(a) Sampling efficiency calculated over time

(b) Sampling efficiency calculated after 2000 measurements

Figure 4.8: Sampling efficiency of DPPS and eSENSE ($F_N = 5\%$)

Results shown in Figure 4.8 reveal that using DPPS increases the sampling efficiency by up to 30% compared with eSENSE. Given that the sampling efficiency is the event-to-measurement ratio, the improvement offered by DPPS over eSENSE can be explained by further examining the usage and transmission percentage plots in Figures 4.6-4.7. For example in Figure 4.6, the lower usage percentage in DPPS in comparison to eSENSE suggests that, on average, DPPS collects less samples and therefore has a larger sampling interval. A large sampling interval allows the sensor to spend more time asleep thereby saving more energy. Also examining the transmission percentages in Figure 4.7 reveals that a higher proportion of measurements taken using DPPS are events. This means DPPS offers improved efficiency in event detection in a sensor unit.

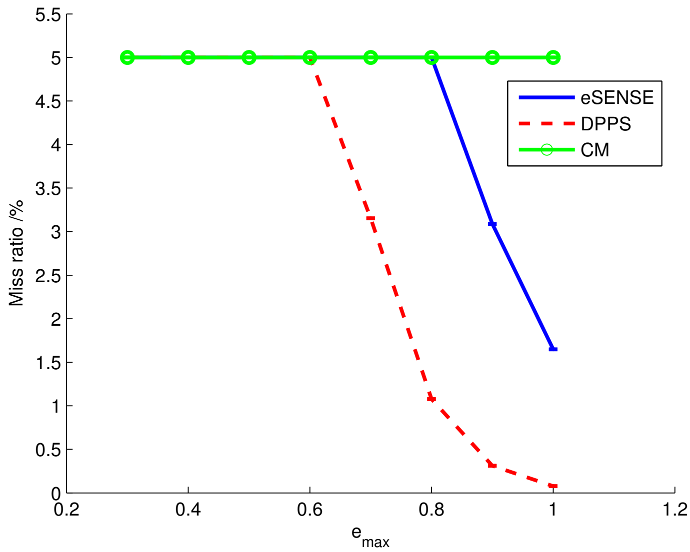


Figure 4.9: Expected miss ratio of DPPS compared to eSENSE and CM ($F_N = 5\%$)

Increased sampling efficiency and lower sensor usage are not the only advantages offered by DPPS over eSENSE. Figure 4.9 shows that DPPS has a lower miss ratio compared with eSENSE when e_{max} is between 0.6 and 1. It is also clear from Figure 4.9 that the miss ratio decreases as e_{max} increases. This is because as e_{max} increases it is less likely that an event will occur, thus it is less likely that an event will be missed. Most importantly the graph reveals that a lower number of missed events occur when using DPPS (compared to eSENSE or CM) and therefore DPPS offers stronger missed ratio guarantees.

The next simulations were done to evaluate the mean square error of DPPS, eSENSE and CM. As Figure 4.10 reveals, DPPS has, on average, a similar mean square error compared to eSENSE at the same e_{max}. More specifically, at an e_{max} value of 0.7 and 0.8, DPPS and eSENSE have the same mean square error. However, at e_{max} values of 0.5 and 0.6, DPPS had a slightly higher mean square error than eSENSE. Owing to the plethora of data collected and transmitted, CM has the lowest mean square error.

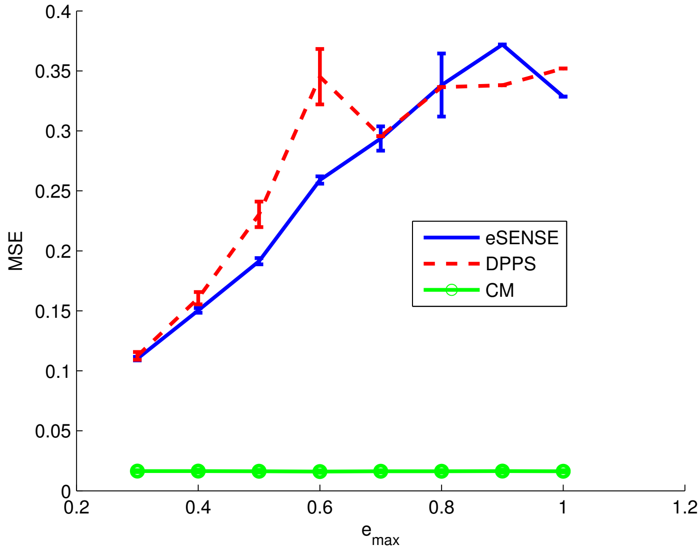


Figure 4.10: Mean square error of DPPS compared to eSENSE and CM ($F_N = 5\%$)

Results corresponding to another time series dataset are shown in Figures 4.11-4.16 when $F_N = 10\%$. These results confirm earlier findings about the improvements offered by DPPS. For example, from analysing Figure 4.11 it can be concluded that DPPS reduces the *number of measured samples* in comparison with eSENSE and CM.

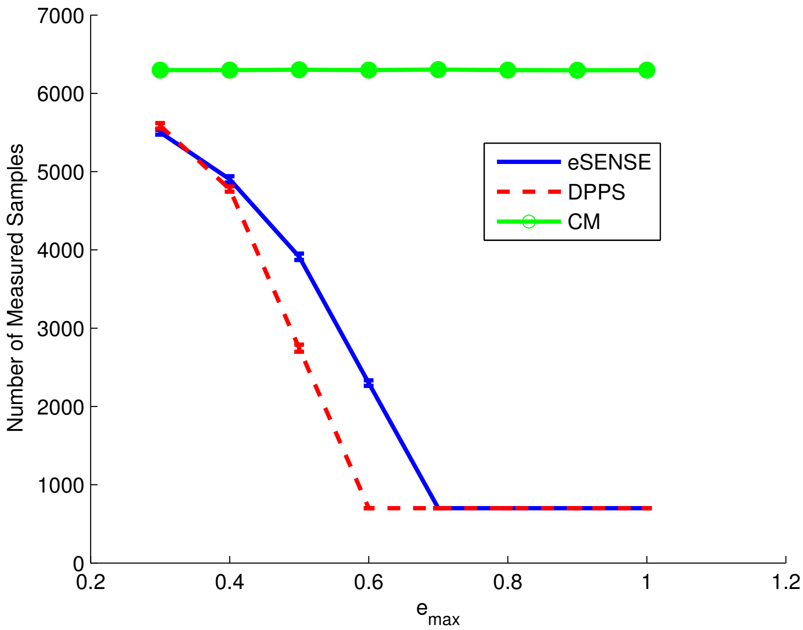


Figure 4.11: Number of measurements using DPPS, eSENSE and CM ($F_N = 10\%$)

As well as confirming earlier findings, these results provide further insights. For example it is interesting to note that the results in Figures 4.11-4.16 are, for the most part, lower than those in Figures 4.5-4.10. This is exemplified in Figure 4.12 where both DPPS and eSENSE have a maximum usage percentage of around 80% at $e_{max} = 0.3$, compared with 90% at the same error threshold in Figure 4.6. This can be explained by the fact that data in Figure 4.12 is collected at $F_N = 10\%$ in contrast with Figure 4.6 where data is collected at $F_N = 5\%$. As F_N increases, a sensor network application becomes more tolerant to missed events and allows sensor units in a network to be asleep for longer thus minimising usage percentages. Therefore it can be concluded that increasing F_N from 5% to 10%, increases energy savings.

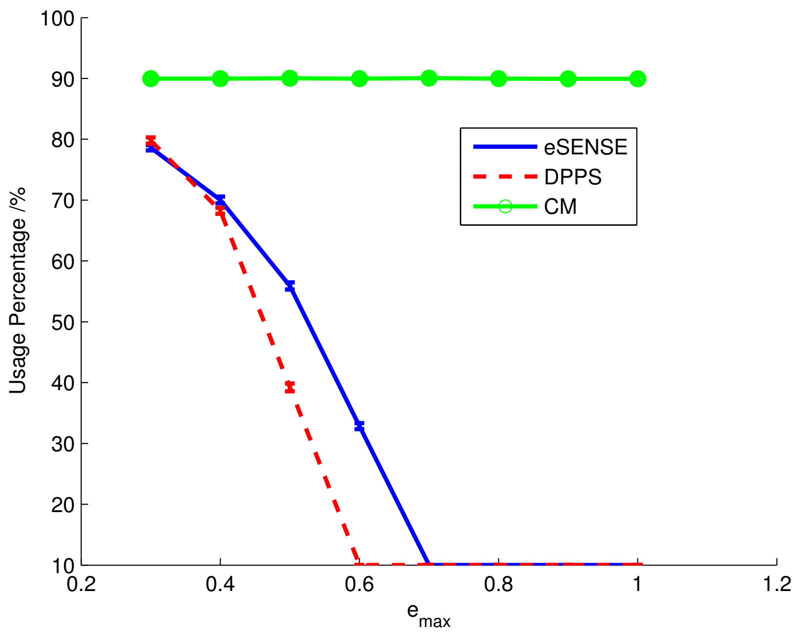


Figure 4.12: Usage percentage of DPPS, eSENSE and CM ($F_N = 10\%$)

Figure 4.12 also shows that DPPS converges more quickly to an optimal operational point than eSENSE or CM thus further enhancing energy savings. As F_N increases, the rate of this convergence also increases and therefore so does DPPS's advantage over eSENSE and CM in terms of energy savings.

Figure 4.13 demonstrates again that DPPS has a higher transmission ratio compared with eSENSE and consequently is more suitable for detecting events efficiently especially when e_{max} is small. As e_{max} increases however, fewer events are detected because the application becomes more tolerant of missed events. Above $e_{max} = 0.70$, both DPPS and eSENSE become so tolerant that events are undetected and therefore the transmission ratio is 0%. This contrasts with the results shown for CM where the transmission ratio is at a constant level of 90%, because as previously mentioned, CM does not distinguish between event and non-event data.

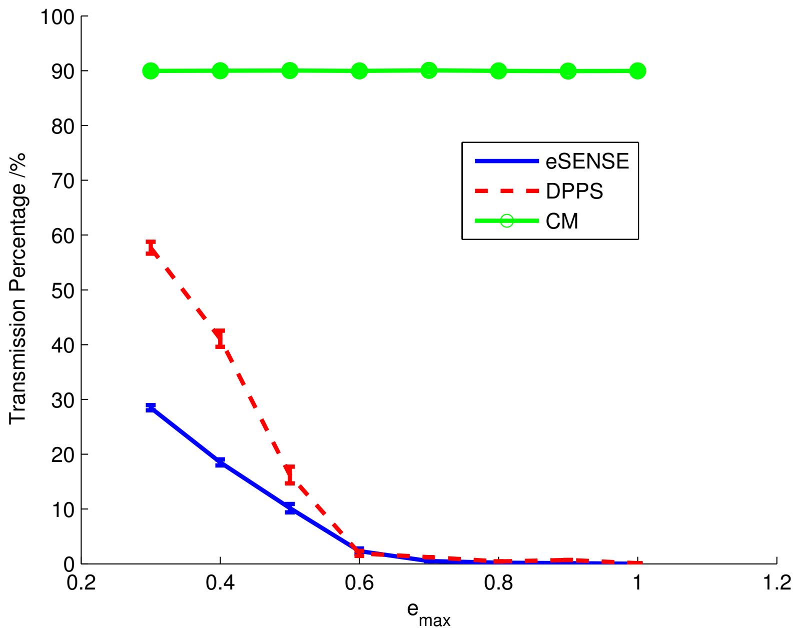
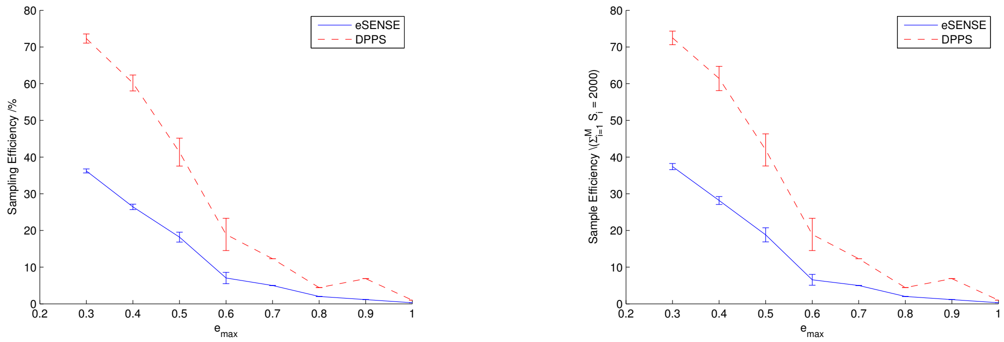


Figure 4.13: Transmission percentage of DPPS, eSENSE and CM ($F_N = 10\%$)

As revealed in Figure 4.8, Figure 4.14 demonstrates that DPPS offers an improvement of up to 35% in terms of sampling efficiency in comparison with eSENSE. This is the case both when the total time is constant and when the total number of samples is constant.



(a) Sampling efficiency of DPPS compared to eSENSE calculated over time

(b) Sampling efficiency of DPPS compared to eSENSE calculated after 2000 measurements

Figure 4.14: Sampling efficiency of DPPS and eSENSE ($F_N = 10\%$)

Figure 4.15 shows that when $F_N = 10\%$ DPPS again offers the lowest miss ratio when compared with eSENSE and CM. This confirms that DPPS reduces the amount of missed events and therefore provides stronger quality guarantees than eSENSE or CM.

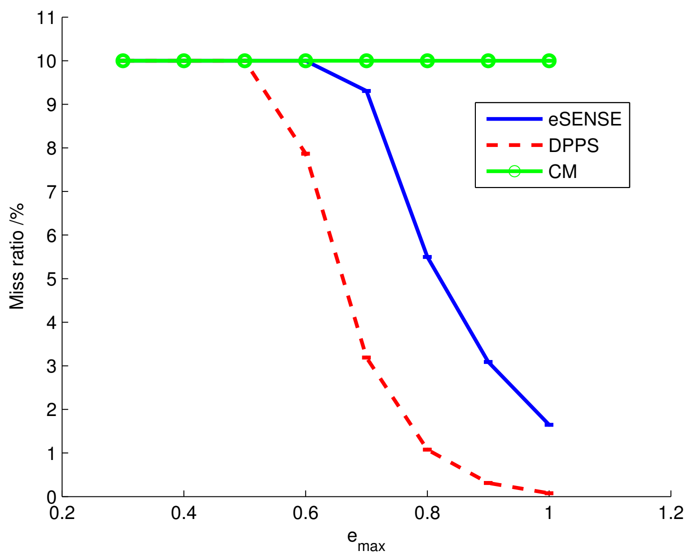


Figure 4.15: Expected miss ratio of DPPS compared to eSENSE and CM ($F_N = 10\%$)

Figure 4.16 shows that both DPPS and eSENSE have comparable mean square errors. Critically, Figure 4.16 also shows that DPPS, despite having a higher mean square error than CM, satisfies the mean square error requirement outlined in Equation 4.3. The results of further simulations using additional datasets are provided in Appendix B. The next section contains preliminary results from empirical experiments which were conducted to demonstrate DPPS's potential as an effective protocol for data collection applications.

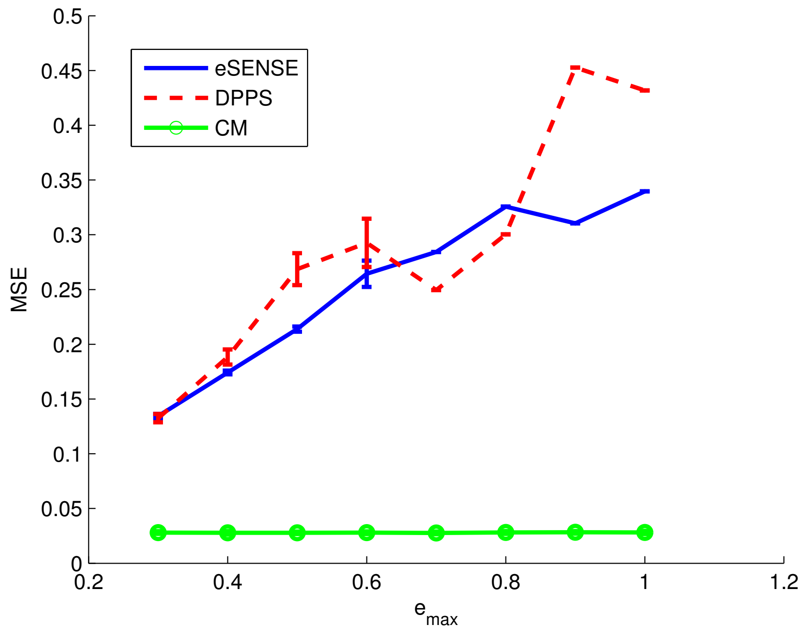


Figure 4.16: Mean square error of DPPS compared to eSENSE and CM ($F_N = 10\%$)

4.8 DPPS Initial Experimental Demonstration

This section looks at the empirical implementation of the algorithms for data collection within an indoor laboratory environment.

4.8.1 Hardware

The time series in this regard were collected using Microchip PICDEMZ boards as shown in Figure 4.17. The PICDEMZ demonstration kit is a user friendly platform for ZigBee application design and development. The demonstration board is fitted with an RJ11 connector which provides an interface to Microchip's MPLAB ICD 2 Debugger. The ICD 2 Debugger allows developers to reprogram or modify code on board the PIC 18F4620 microcontroller unit flash memory. Also part of the demonstration kit is

Microchip's MPLAB IDE which provides the platform for writing and debugging code for application development. Every PICDEMZ board was powered by 9V-170mAh rechargeable batteries. The MCU contained 1.5K RAM, 32K programme memory and 1024 bytes EEPROM. The EEPROM memory was the most critical storage component because it was used for programming the scheduling algorithms as well as storing 360 bytes worth of parameters.

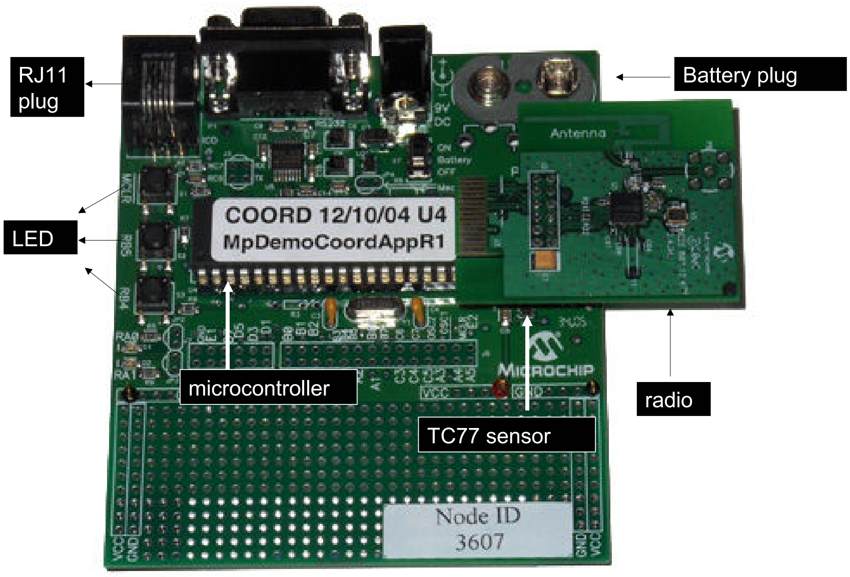


Figure 4.17: The main components in a PICDEM Z board include microcontroller, radio and a measurement sensor

Other important peripheries located on the board are:

- Temperature sensor - A TC77 thermal sensor was used for temperature data collection.
- LED - The LED was also used to check the integrity of the data collection algorithms. This was done by setting the LED to blink when measurements were being taken.
- Radio - A CC2420 radio was used to send and receive packets at 2.4 GHz. The radio consumed 18.8 mA and 17.4mA for receiving and transmitting data respectively.

4.8.2 Firmware

A simplified version of Senceive Ltd.'s FlatMesh firmware was used for implementing the various algorithms onboard the PIC18F4620 microchip. Senceive Ltd is a company which manufactures wireless sensor networks typically for industrial and environmental monitoring. Having developed their own proprietary mesh networking protocol, the people at Senceive Ltd. were interested in developing intelligence into their networks by using algorithms that could adapt data collection in order to limit network traffic and also further extend battery life.

Senceive Ltd. assisted with this study by providing both the hardware and a simplified version of their FlatMesh firmware for testing of our algorithm. The stripped down FlatMesh firmware was geared for point to point communications in a star topology (see Figure 4.18).

The firmware was used because its modular architecture abstracted the physical layer processes and networking features from the functions in upper layers. In particular the top level application layer allowed the development and customisation of application based algorithms without knowledge of the underlying framework in lower layers. The firmware was written in C using Micropchip's MPLAB integrated development environment with a C18 compiler [Goi08].

The core module customised for this study was an application layer module used as an interface for data sampling. The module was augmented to include DPPS for adaptive data sampling and transmission. This was done for both a send-on-sample and a send-on-requirement policy.

4.8.3 Experimental Setup

In experiments, six PICDEMZ boards including one gateway node were deployed in an indoor laboratory environment. One node collected data using Continuous Monitoring (CM) at a baseline granularity of one second and the other two pairs of nodes collected data using eSENSE and our proposed algorithm DPPS respectively. The sampling interval on each node varied between 1 and D_{max}. Because the interval between sampling is small (i.e. in seconds) compared to the rate of temperature variation in the room being monitored, a small miss ratio of $F_N = 10\%$ was used in conjunction with $e_{max} = 0.1, 0.2, 0.3$. In both cases the algorithms were implemented to test the usage percentage and data transmission ratio at the selected error thresholds. Figure 4.18 shows the experimental setup. Two hours of data collected first at the baseline sequence was used to build up the model. Next, the newly created model was applied to the next two hours of data in order to evaluate the performance. This procedure was done on all sensor nodes on which the proposed algorithm was implemented for both eSENSE and DPPS.

Figure 4.18: Experimental Hardware: the main setup components included a 9V power supply, Microchip PICDEMZ boards fitted with Chipcon CC2420 transceivers, ICD 2 debugger and a laptop computer

All parameters θ_k, μ_k and σ_k, were calculated using the method described in Section 4.5. A send-on-sample policy where all measured samples are transmitted was used to collect the usage percentage. Conversely a send-on-requirement policy, involving only measurements determined to be significant events, were used to determine transmission percentage. All the data collected were grouped into the fields as shown in table 4.3.

Table 4.3: Data fields reported to the sink from sensors

Data Field
node ID
$date$
$time$
report number
$temperature$

The report number is the total number of samples measured. It is incremented after every sensor measurement and therefore was used to determine the sampling efficiency which is the ratio of the total number of samples transmitted to the total number of samples observed.

4.8.4 Experimental Results and Analysis

The empirical evaluation of performance compared DPPS to eSENSE using the usage percentage, transmission ratio and the sampling efficiency metrics.

Figure 4.19 shows the time series data as collected using a send-on-sample policy for both the probabilistic algorithm DPPS and eSENSE. Figure 4.20 shows that DPPS has a lower usage percentage than eSENSE between $e_{max} = 0.1$ and $e_{max} = 0.2$. For example at $e_{max} = 0.15$, DPPS has an average usage percentage of 6.1% compared to 11% for eSENSE. This means that the sensor is used less and hence more energy is conserved. When $e_{max} > 0.2$, the error threshold is too large and the sampling interval remains constant using both algorithms.

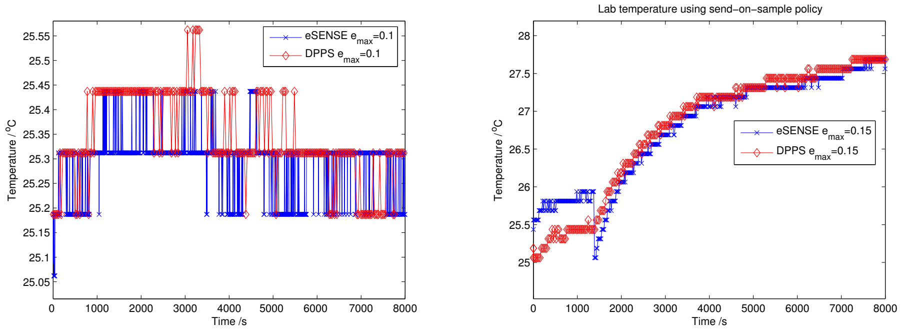


Figure 4.19: Temperature time series as collected using DPPS, in comparison with eSENSE as acquired empirically for a send-on-sample policy at various e_{max} values

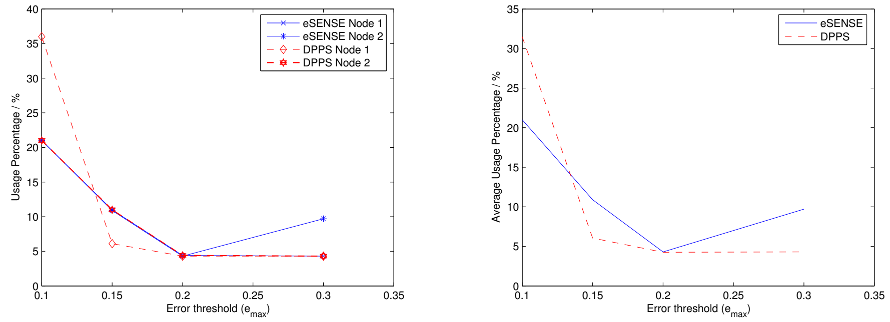


Figure 4.20: The usage percentage and the average usage percentage of DPPS in comparison with eSENSE as acquired empirically

Figure 4.21 shows the transmission percentage recorded using our proposed algorithm DPPS in comparison to eSENSE and CM between $e_{max} = 0.1$ and $e_{max} = 0.3$.

Figures 4.22(a)-4.22(b) show the corresponding transmission percentage and the sampling efficiency as derived empirically. At an error threshold of $e_{max} = 0.1$, Figure 4.22(a) shows that using a send-on-requirement policy significantly reduces the transmission percentage of DPPS to 8% from the average usage percentage of 31% shown in Figure 4.20. This is because the send-on-requirement policy only sends data recorded as events rather than all measurements taken. Figure 4.22(b) shows that DPPS has a higher sampling efficiency than eSENSE because less false positives are measured using DPPS. This indicates that, although more samples are sent to the base station using DPPS than eSENSE, a higher proportion of the samples taken are events. The

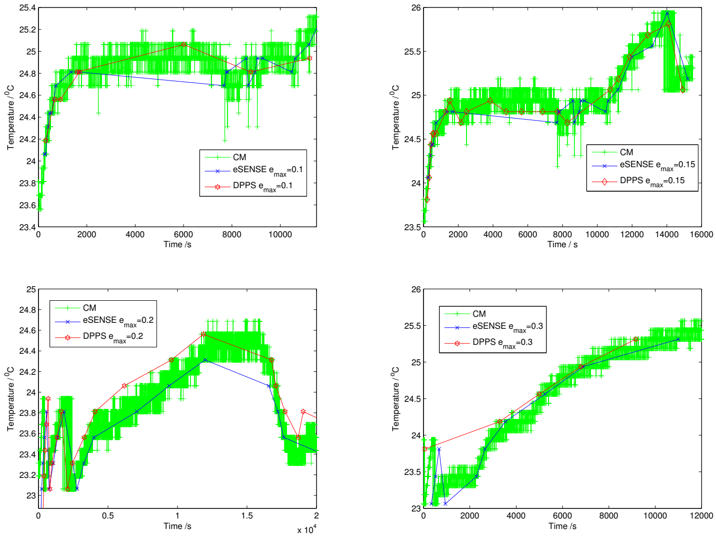
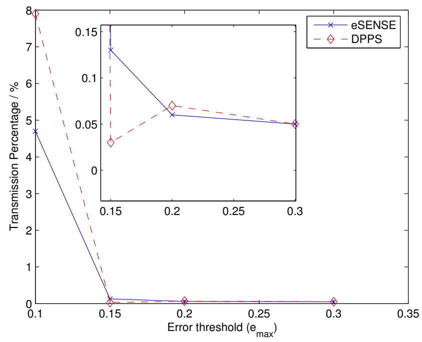
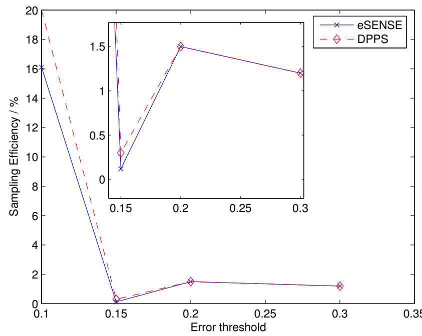


Figure 4.21: Temperature time series as collected using DPPS in comparison with eSENSE and CM

results in Figure 4.22 also indicate that a sensor running DPPS is used less often for event measurement compared with a sensor running eSENSE.



(a) Experimental transmission percentage



(b) Experimental sampling efficiency

Figure 4.22: Experimental transmission percentage and sampling efficiency

This is exemplified when $e_{max} = 0.15$; both the transmission percentage (0.015%) and the sampling efficiency (0.25%) of DPPS provides a marginally improved performance over eSENSE. Figure 4.22 also demonstrates that the transmission percentage and the sampling efficiency decreases with increasing e_{max}. This occurs because an increase in e_{max}, decreases the chance of event capture and leads to a decrease in the sampling rate thus causing a sensor's communication unit to be less active over time. Measurements acquired with $e_{max} \geq 0.2$ produced identical transmission and sampling efficiencies using both algorithms because the error threshold is too large to distinguish their event capture performance.

4.9 Chapter Summary

This chapter presented and reviewed DPPS, a Dual Prediction and Probabilistic Scheduler for sensor network applications. DPPS extends the eSENSE framework by combining both Compression and Load Balancing techniques in a Dual Prediction Scheme. Simulation results were presented which revealed that DPPS provides improved sensor usage, higher sampling efficiency and a higher transmission percentage of events in comparison with eSENSE and CM. More specifically simulation results showed that DPPS offered reductions of up to 35% in sensor usage and an increase of up to 35% in the efficiency of sampling events in comparison with eSENSE. In essense this means DPPS offers improved efficiency in event detection in a sensor unit. Additionally DPPS has, on average, a lower miss ratio than eSENSE or CM while simultaneously satisfying the mean square error constraint required in an application.

Chapter 5

Adaptive Detection-driven Ad hoc Medium Access Control

5.1 Introduction

In sensor networks, medium access control protocols, such as SMAC, minimise energy consumption by scheduling sensor nodes to sleep periodically. However, the effectiveness of periodic scheduling protocols in large multi-hop networks is affected by the Data Forwarding Interruption problem which increases end-to-end delay. In this chapter, the Data Forwarding Interruption problem is addressed using an Adaptive Detection-driven Ad hoc Medium Access Control (ADAMAC) algorithm. ADAMAC limits end-to-end delay in a multi-hop sensor network while reducing energy consumption by combining the probability of an event occurring with an early warning framework in order to adapt the duty cycle of a network.

5.2 Motivation

Periodic scheduling algorithms such as SMAC [YHE02] offer a simple means of controlling the energy consumption in a sensor network. Such protocols cause sensor nodes to become active at fixed intervals throughout the lifetime of a network. Energy is thus conserved because components in a sensor node are intermittently active and can be asleep for long periods of time.

Although periodic scheduling protocols increase energy savings, their effectiveness is affected by the Data Forwarding Interruption problem. This makes such protocols inadequate for some monitoring applications because the Data Forwarding Interruption problem leads to increased delay as data is reported across a network. In some environmental monitoring applications increased delay is a problem because following the occurrence of a natural disaster such as a landslide, data may need to be forwarded quickly and efficiently so that emergency services can undertake remedial actions faster.

In order to address the Data Forwarding Interruption problem and thus facilitate the expanded applicability of sensor networks into wider areas, an Adaptive Detection-driven Ad hoc Medium Access Control algorithm (ADAMAC) has been designed in this chapter. ADAMAC adapts the communication frequency of a network both before and after the occurrence of an event using early warning event indicators. These event indicators are provided by probabilistic algorithms such as DPPS in Chapter 4. By adjusting the communication frequency of a network ahead of time, the effects of the Data Forwarding Interruption problem are minimised. ADAMAC can therefore limit both energy consumption and end-to-end delay during the detection and reporting of events. In order for ADAMAC to successfully adjust the communication frequency of a network, the challenge of efficiently adapting the duty cycle of a network must be addressed.

5.3 Adaptive Duty Cycling: A Challenge

The challenge of adaptive duty cycling concerns the optimal adjustment of the sleep-wake cycle of a network in order to detect events while limiting both energy and delay. Although it is impossible to know exactly when an event will occur, estimates of an event occurrence time can be calculated ahead of time using trends in historical data. Though such estimates are impossible with regards to unexpected events, such as a sudden bridge collapse, other events such as forest fires, landslides or embankment failures do follow trends from which accurate estimates may be obtained.

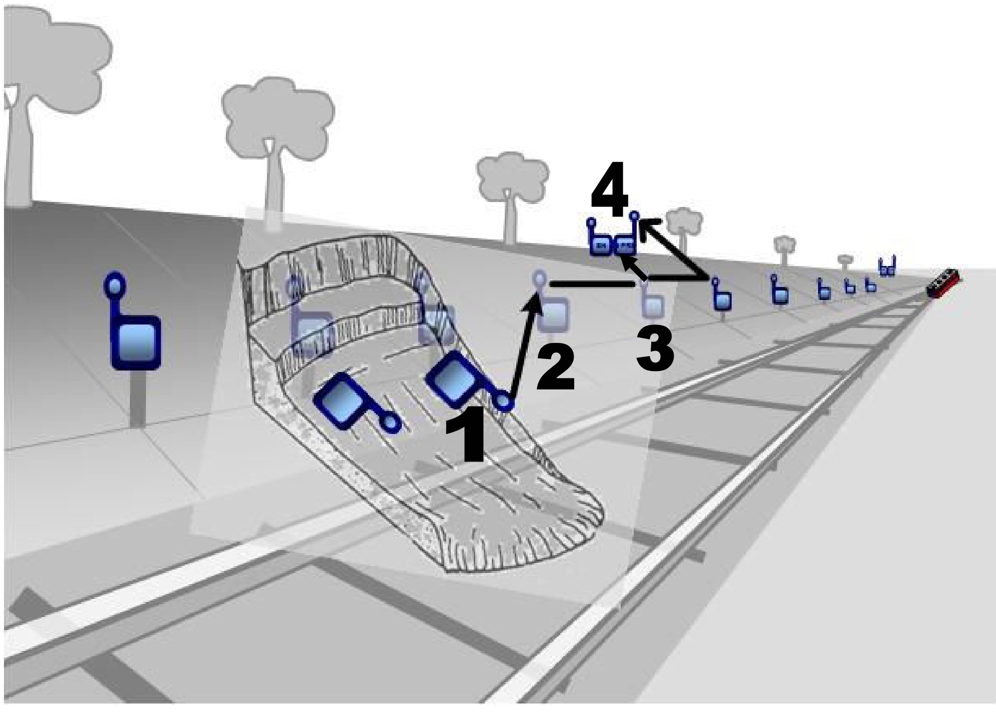


Figure 5.1: A rail-side embankment failure. Prior to the embankment failure, the tilt angle from the sensors trigger early warning signals. The nodes used in this picture are courtesy of Senceive Ltd.

Figure 5.1 shows the collapse of a rail-side embankment being monitored by a sensor network. The sensor nodes along the embankment rotate in proportion to the severity of the embankment's collapse. Event data is reported to emergency services when a sensor node's rotation exceeds a certain threshold. This is illustrated in Figure 5.1

where, following the collapse, event data from node 1 is reported to node 2; node 2 forwards the data to node 3 and node 3 similarly reports the data to node 4, the base station (BS). The base station, which has long-range communication capabilities, can then alert emergency services.

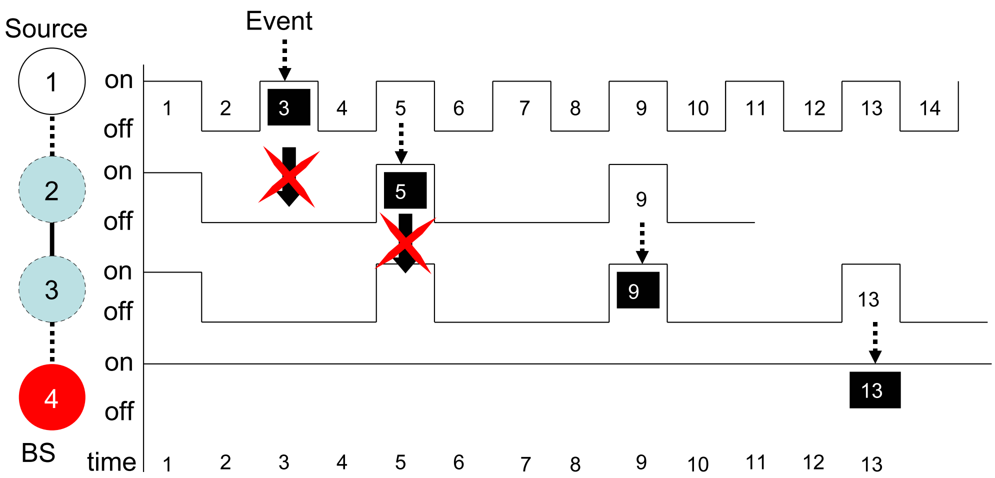


Figure 5.2: The effect of differing sleep-wake cycles on end-to-end delay and energy consumption. An event occurs at time t = 3 seconds and is reported to the base station with an end-to-end delay of 10 seconds

In order to illustrate the adaptive duty cycling problem, a 1 dimensional outline of the embankment collapse is shown in Figure 5.2. Assume that node 1 has a 50% sleep-wake cycle, nodes 2 and 3 use a 25 % sleep-cycle and node 4, the base station, is always awake. If an event were to occur at time $t = 3$ seconds as shown, the end-to-end delay from node 1 to node 4 would be 10 seconds. This is because when node 1 detects the event at time $t = 3$ seconds, it cannot immediately report the event to node 2 because of the Data Forwarding Interruption problem. Node 1 can only pass on data when node 2 is awake at time $t = 5$ seconds. Upon receipt of event data from node 1 at time $t = 5$ seconds, node 2 cannot immediately send the data onto node 3, even though node 3 is awake, because of the communication restriction imposed to prevent the Broadcast Storm problem (see Section 3.6.2 in Chapter 3). Node 2 must wait until its next wake cycle at time $t = 9$ seconds before the message is reported to node 3. Again the effects of the Data Forwarding Interruption problem mean that node 3 must wait until time $t = 13$ seconds before the event data can finally be reported to the base station thus incurring a total end-to-end delay of 10 seconds.

Figure 5.3 shows the effect of adjusting the sleep-wake cycle of nodes 2 and 3 to 50% prior to the event occurrence at time $t = 3$ seconds. Figure 5.3 reveals that the end-to-end delay is reduced by four seconds when compared with Figure 5.2. However, more energy is consumed because sensor nodes are active more often.

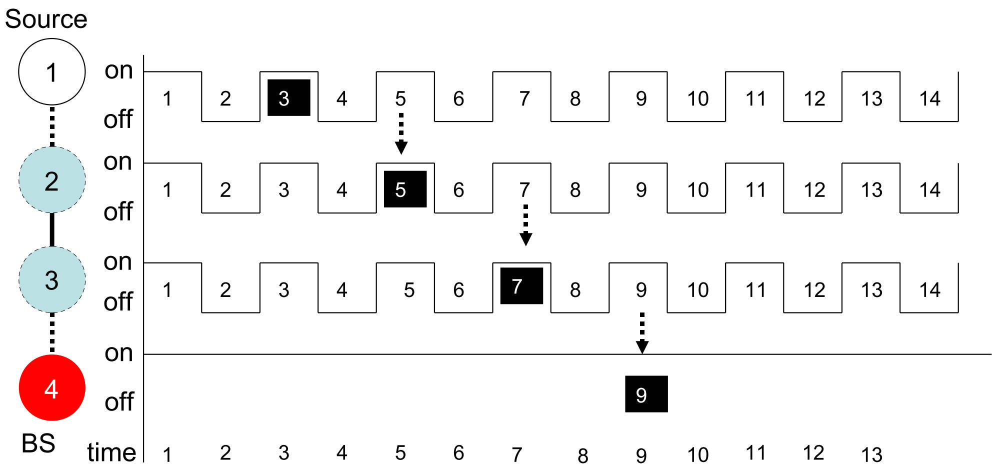


Figure 5.3: Adjusting the sleep-wake cycle to reduce end-to-end delay

In order to address the challenges of adaptive duty cycling while minimising energy consumption, an algorithm is needed which allows all nodes to sleep for as long as possible before an event occurs. Just before an event is likely to occur, the algorithm must adjust the communication frequency of a network in order to reduce end-to-end delay. The algorithm must also re-adjust the communication frequency after the event has been reported. In the next section, the problem of adaptive duty cycling is formulated mathematically in order to develop ADAMAC, an algorithm that facilitates the reduction of both end-to-end delay and energy consumption.

5.4 Adaptive Duty Cycling: Problem Formulation

Table 5.1: Notation of parameters used in ADAMAC

Parameter	Definition (Value)
	Generic Parameters
h	Number of hops from a source node to a base station
TP_{max}	Maximum toggling period in an application
	ADAMAC Parameters
$X_{warning}$	Event warning threshold
X_{event}	Event occurrence threshold
α	Event coefficient
ϕ	Adaptation policy
ϕ_b	Adaptation policy at *breakdown*
θ	Event occurrence rate
θ_b	Critical *breakdown* rate
q	Event probability
$\beta(q, \phi)$	Toggling period adaptation function
M	Maximum number of warning levels. M is also the maximum number of toggling frequencies
WL_i	i^{th} event warning level
TP_i	Toggling period at the i^{th} warning level
f_i	Toggling frequency at the i^{th} warning level
Ω	Total number of active cycles in a network
ρ	Ramp down constant
t_r	Transition time

Figure 5.4 illustrates the conceptual relationship between a Toggling period (TP) and a Duty Cycle (DC). In each TP, nodes adopt either an active or sleep state. The duty cycle is adapted by changing the length of time a node remains in a sleep state between two consecutive active states.

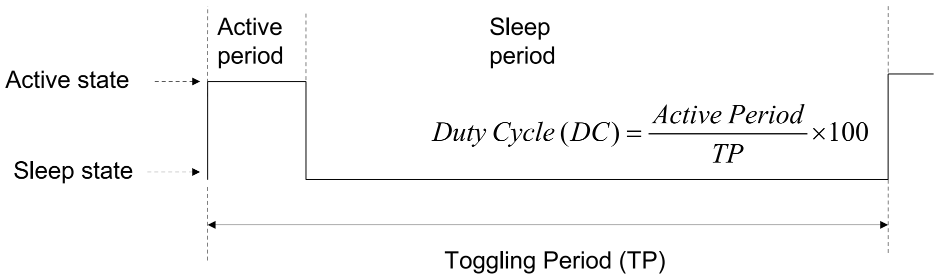
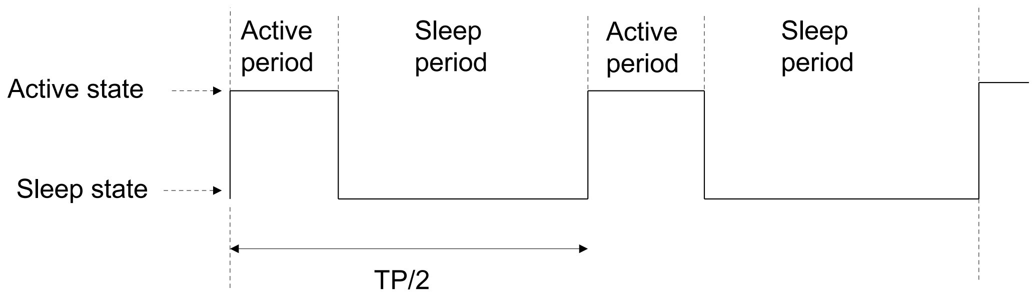


Figure 5.4: The relationship between toggling period and duty cycle

As an example, in Figure 5.5 the duty cycle is doubled in comparison to Figure 5.4, by re-activating the sensor after an interval $\frac{TP}{2}$. The maximum toggling period is TP_{max} and re-activation can take place at M toggling periods $\{TP_i = 2^{i-1} | i \in [1, M]\}$ corresponding to M communication frequencies $\{f_i = \frac{2^{i-1}}{TP_{max}} | i \in [1, M]\}$.



$$Duty\ Cycle\ (DC) = \frac{2 \cdot Active\ Period}{TP} \times 100$$

Figure 5.5: Adapted toggling period

Toggling period adaptation is triggered when measurements X_t exceed predetermined event warning levels $\{WL_i | i \in [1, M]\}$ where $WL_i \leq X_{event} \forall i \in [1, M]$. These event warnings occur within an input measurement range defined by:

$$X_t = \begin{cases} X_{init} & 0 < \mathrm{t} \leq t_{init} \\ \theta t & t > t_{init} \end{cases}$$

As time t_{init} elapses, X_t increases from an initial reading level X_{init} at the event occurrence rate θ. At the event occurrence time, t_{event}, X_t reaches the event occurrence level X_{event}. In algorithms such as eSENSE [LCS06] and DPPS [EY09], the occurrence of an event is indicated by the probability of an event occurring, q, which maps a reading to an event probability. In ADAMAC, after q has been obtained, it is correlated to a toggling period using a *toggling period adaptation function*, $\beta(q, \phi)$, which allows the duty cycle of a network to be adjusted. The problems of adaptive duty cycling in a sensor network with limited energy supply may be expressed as:

Given an event with an occurrence rate of θ, how can the duty cycle of a sensor network be adapted using the toggling period adaptation function $\beta(q, \phi)$ in order to limit both energy consumption and delay during event detection and dissemination?

Providing a solution to this question is challenging because it requires:

- The development of an adaptation function - A *toggling period adaptation function*, $\beta(q, \phi)$, must be developed so that the toggling period can be adjusted using an adaptation mechanism.
- The development of an adaptation mechanism - Shortly before the onset of an event, the duty cycle of a sensor network must be high in order to minimise end-to-end delay. When the event has passed, the sensor network must decrease its duty cycle so that energy consumption is reduced.
- The determination of the event occurrence time - When using applications in certain environments, event occurrence times are unknown and therefore estimates must be calculated using trends from historical data.

5.5 Development of ADAMAC

In this section, the development of ADAMAC is discussed. Firstly in Section 5.5.1 a *toggling period adaptation function* is developed. Secondly in Section 5.5.2 the *toggling period adaptation function* is examined for *breakdown*; *breakdown* leads to increased inefficiencies in event detection applications because end-to-end delay and energy savings can no longer be traded off effectively. The conditions that cause *breakdown* are therefore discussed in detail. Thirdly in Section 5.5.3 the parameters required to avoid *breakdown* of the *toggling period adaptation function* are outlined and then used to propose an adaptation policy which would increase energy savings while minimising end-to-end delay and avoiding *breakdown*. Details of a prediction model for estimating the event occurrence rate, θ, are also provided in order that ADAMAC could be used in applications where the event occurrence time is unknown. Finally in Section 5.5.4, an overview of ADAMAC is illustrated using the example of an embankment collapse. This overview also includes the pseudo-code for ADAMAC.

5.5.1 Toggling Period Adaptation Function

As previously outlined in section 5.3, an energy efficient solution to the challenges of adaptive duty cycling necessitates that all nodes in a network sleep for as long as possible before an event occurs. This means that when the event probability q is close to 0, the toggling period of a network should be high so that the communication frequency is low. Using a low communication frequency increases energy savings in a network. Alternatively, just before the onset of an event when q is close to 1, the toggling period of a network should be low so that the communication frequency is high. This high communication frequency minimises delay when data is reported across a network. These characteristics are precisely matched by the power function shown below:

$$1 - q^{\phi} \tag{5.1}$$

In the expression of 5.1 the adaptation policy, ϕ, is a non-negative real number. The expression was inspired from [MS90] where such power functions are shown to be effective methods of adapting the communication frequency of a self-organised system. When q is close to 0, 5.1 places more priority on increasing energy savings by decreasing communication frequency. Conversely when q is close to 1, 5.1 decreases end-to-end delay by increasing communication frequency. In order for 5.1 to adapt the toggling period of a sensor network with M warning levels, the *toggling period adaptation function* below can be used:

$$2^{(M-1)(1-q^{\phi})} \tag{5.2}$$

The *toggling period adaptation function* of 5.2 ensures that the toggling frequency is in the required range $\{f_i = \frac{2^{i-1}}{TP_{max}} | i \in [1, M]\}$ as discussed previously in Section 5.4. Increasing or decreasing the toggling frequency of a network is controlled by adjusting the adaptation policy, ϕ. Adjustment of the toggling period using four different *toggling period adaptation functions* is illustrated in Figure 5.6.

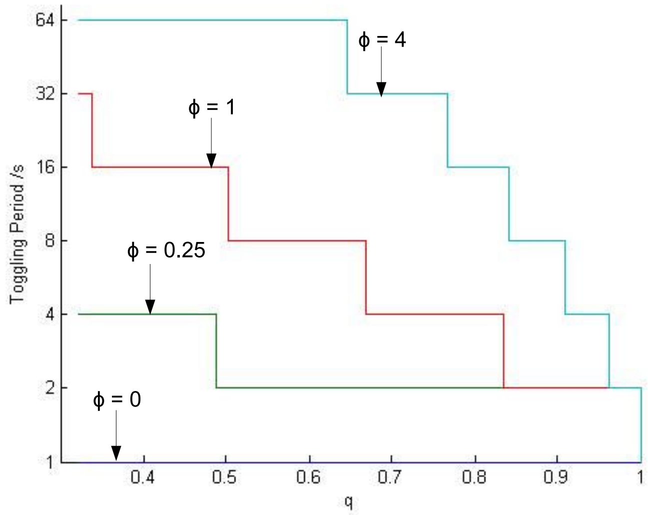


Figure 5.6: Relationship between ϕ and the toggling period

$\phi = 0$ corresponds to a Fully Active network with a toggling frequency of $f_0 = \frac{1}{2^0} = 1$. Using $\phi = 0$ the end-to-end delay is minimised but energy consumption is maximised. As ϕ increases to values of 0.25 and 1, the energy savings increase but so too does the end-to-end delay. Toggling period adaptation when $\phi = 4$ saves the most energy but using this adaptation policy produces the highest end-to-end delay. Adjustment of ϕ is therefore critical because it controls the sleep-wake cycle of a sensor node, managing the trade-off between energy savings and delay across a sensor network.

5.5.2 *Breakdown*

As ϕ increases there is a higher likelihood of experiencing *breakdown*, an effect in which end-to-end delay sharply increases during the adaptation of a network's toggling period. *Breakdown* makes it more difficult to efficiently trade-off energy and delay. To further explain *breakdown*, consider a network that uses an adaptation policy of $\phi = 4$. As illustrated in Figure 5.6, such a network has a toggling period of 64 seconds when $q \leq 0.65$. Thereafter the toggling period rapidly decreases to 0 as q increases towards 1. When $q = 1$ a toggling frequency of f_0 is immediately deployed by the event node for event reporting. However, a minimum transition time is required for this toggling frequency to reach all nodes in the sensor network. The length of this transition time varies depending on the average toggling frequency of all nodes between the source and the base station. When this transition time is too long in comparison to the event occurrence rate, the forwarded event data is delayed in some parts of a network because of the Data Forwarding Interruption problem. The agglomeration of such delays causes the *toggling period adaptation function* to *breakdown*.

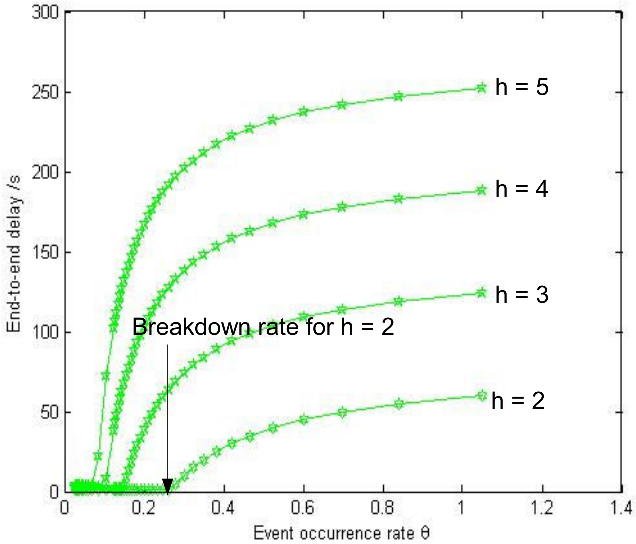


Figure 5.7: Breakdown in a fully active network at different hop counts, h.

In ADAMAC, *breakdown* is especially noticeable after a particular point known as the Critical Breakdown Rate, θ_b, when the end-to-end delay sharply increases. In order to further evaluate the effects of *breakdown*, Figure 5.7 shows the end-to-end delay at various event occurrence rates. For example when hop count $h = 2$ in Figure 5.7, breakdown occurs at $\theta_b = 0.3$. Each h hop network was simulated with MATLAB in order to form a connected linear network of $h + 1$ nodes in a 100 m^2 plane. Although nodes use a fully active schedule in order to report data, the network is assumed to have an initial sampling period of 64 seconds. End-to-end delay is the time taken for an event packet from a source to reach its destination h hops away.

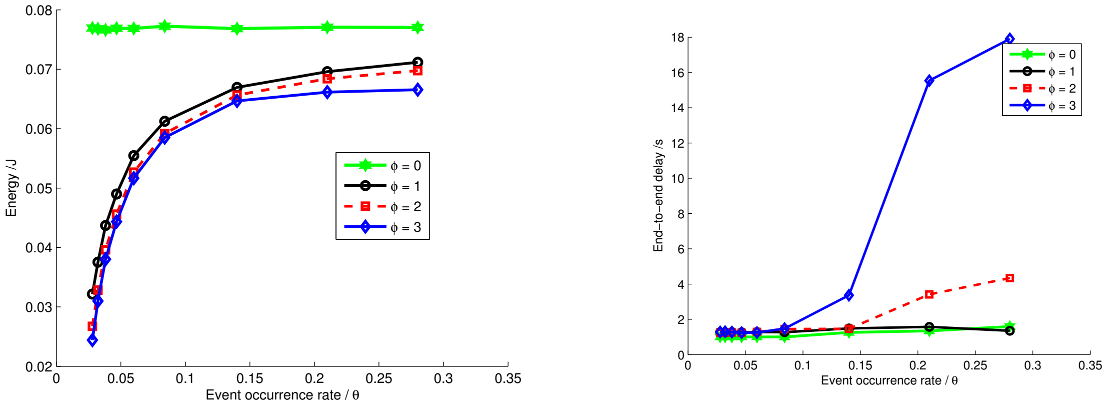
Figure 5.7 clearly shows that increasing the hop count, in a network substantially increases the end-to-end delay after the Critical Breakdown Rate (CBR). The increased end-to-end delay occurs because as the network size increases, more sensor nodes suffer from the effects of the Data Forwarding Interruption problem. As a result *breakdown* is more apparent in sensor networks where the distance between a source and a base station is large. More formally, *breakdown* occurs when a network's transition time, t_r, and the event occurrence rate, θ, violates the following inequality:

$$\frac{dX}{t_r} \geq \theta \tag{5.3}$$

$dX = X_{event} - X_t$ is the increase necessary for a reading at time t to reach the event occurrence level. The transition time, t_r, in a h hop network where all the nodes have the same toggling frequency, f_i, is given by:

$$t_r = \frac{h-1}{f_i} \tag{5.4}$$

In order to further explore the *toggling period adaptation function* and demonstrate the importance of avoiding *breakdown*, preliminary simulations were done using the expression of 5.2 at various event occurrence rates ranging from $0.0280 < \theta \leq 0.28$. Figure 5.8 shows the results of these preliminary simulations and confirms previous assertions that increasing ϕ decreases the energy consumption. This is explained by the fact that as ϕ increases the average toggling period and therefore the proportion of sleeping nodes in a network increases. Most importantly Figure 5.8 also shows that, while increasing ϕ decreases energy consumption, end-to-end delay is limited when *breakdown* is avoided. For example in Figure 5.8(b) end-to-end delay using $\phi = 1$ is similar to end-to-end delay using $\phi = 0$.



(a) Energy against θ at various adaptation policies ϕ

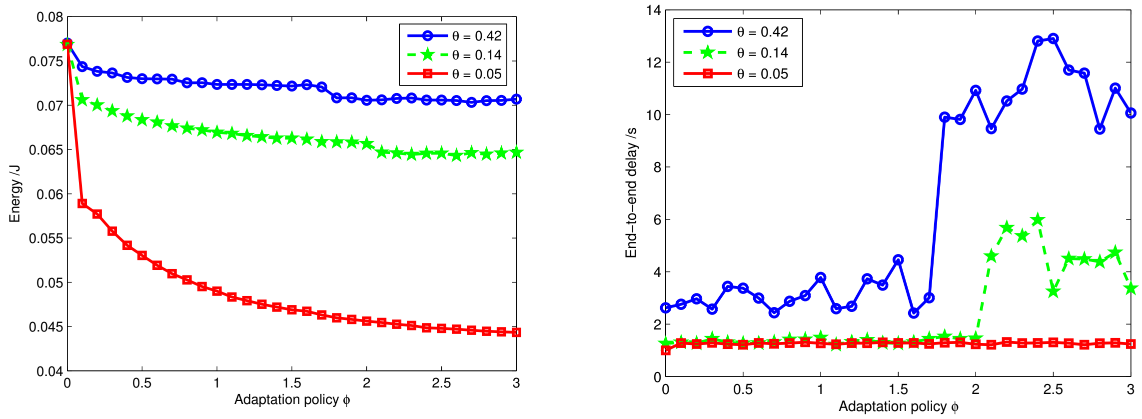
(b) Delay against θ at various adaptation policies ϕ

Figure 5.8: Energy and delay plots against θ. Simulations were obtained using a randomly deployed sensor network. Results shown are the mean from 100 random simulation runs of a 10 node network at various values of ϕ

The *toggling period adaptation function* of 5.2 makes it possible to limit both energy consumption and end-to-end delay when the highest adaptation policy that avoids *breakdown*, ϕ_b, is used. Figure 5.8(b) shows that as the event occurrence rate increases, the value of ϕ_b decreases. For example $\phi_b = 4$ when $\theta = 0.028$ and $\phi_b = 1$ when $\theta = 0.28$. A summary of these and other relevant findings revealed by Figure 5.8 are outlined below:

- Increasing ϕ decreases energy consumption
- As ϕ increases, energy consumption decreases and the rate of change of this decrease in energy consumption converges
- Increasing ϕ does not alter end-to-end delay if *breakdown* has not occurred
- Increasing ϕ increases end-to-end delay if *breakdown* has occurred
- Beyond a Critical Breakdown Rate, θ_b, the end-to-end delay increases
- Increasing ϕ for any given event occurrence rate, θ, will eventually lead to *breakdown* when $\phi > \phi_b$

Another preliminary simulation was done in order to determine the critical *breakdown* rate, θ_b, after which events cannot be detected or reported efficiently. This was done by evaluating the value of θ_b at various adaptation policies as shown in Figure 5.9.



(a) Energy against ϕ at various event occurrence rates θ

(b) Delay against ϕ at various event occurrence rates θ

Figure 5.9: Energy and delay plots against ϕ at various event occurrence rates. Results were obtained using the mean of 100 random simulation runs of a 10 node network at various values of θ

Figure 5.9 shows that before θ_b, increasing θ also increases the energy consumption but the end-to-end delay remains largely unaffected. Above the θ_b, increasing θ increases the energy consumption and also increases end-to-end delay. Essentially as ϕ increases, the value of θ_b decreases, making it more difficult to detect events efficiently. These characteristics along with other findings revealed from Figure 5.9 are outlined below:

- Increasing θ increases energy consumption
- As θ increases, energy consumption increases and the rate of change of this increase in energy consumption converges
- Increasing θ at any given ϕ, does not alter end-to-end delay if *breakdown* has not occurred
- Increasing θ at any given ϕ, increases the end-to-end delay if *breakdown* has occurred
- Increasing θ at any given ϕ will eventually lead to *breakdown* when $\theta > \theta_b$

From findings in both Figure 5.8 and 5.9, it can be deduced that ϕ should be large before *breakdown* in order to maximise energy savings. Conversely ϕ should be relatively small after *breakdown* in order to minimise end-to-end delay. This can be explained intuitively: a sensor network should sleep as much as possible when the event occurrence rate is low so that more energy is conserved before the occurrence of an event. Alternatively after the occurrence of an event, the sensor network should become more active in order to reduce end-to-end delay.

Management of the trade-off between conserving energy and minimising end-to-end delay is paramount, not only for elongating the lifetime of a network, but also for improving the efficiency of data collection. Such management can be enhanced by avoiding *breakdown* whenever possible in order to reduce the effects of the Data Forwarding Interruption problem.

5.5.3 Breakdown Avoidance

In order to avoid *breakdown* while maximising energy savings, Equation 5.3 must firstly be adjusted so that:

$$\frac{dX}{t_r} = \theta \tag{5.5}$$

Then, substituting $dX = X_{event} - X_t$ and $t_r = \frac{h-1}{f_i}$ from Equation 5.4 gives:

$$\begin{aligned} \frac{(X_{event} - X_t)f_i}{h-1} &= \theta \\ \frac{X_{event} - X_t}{2^i(h-1)} &= \theta \end{aligned}$$

where $2^i = 2^{(M-1)(1-q^{\phi})}$ as seen in the expression of 5.2. Therefore:

$$\frac{X_{event} - X_t}{2^{(M-1)(1-q^{\phi})} \times (h-1)} = \theta$$

Rearranging the above gives:

$$\phi_b = \frac{\log\left(1 - \frac{\log\left(\frac{X_{event}-X_t}{\theta(h-1)}\right)}{(M-1)\log 2}\right)}{\log(q)} \tag{5.6}$$

Although adapting ϕ using the above equation maximises energy savings before event occurrence, after event occurrence energy savings may be adversely affected. This can happen when a high communication frequency is used for extended periods after an event has been reported. This energy consumption can be represented by the total number of active sensor wake-up cycles, Ω, used when monitoring over a period of time T as shown below:

$$\Omega = \gamma + \left(\sum_{i=1}^{i=h-1} (h-i)\frac{f_i}{f_j} + i\right) + (h+1)\left\lfloor (T - \tau + t_0)\, f_i \right\rfloor \tag{5.7}$$

Note that $\gamma = (h-1)\frac{f_i}{f_j} + 1$ and $\tau = \frac{(h-1)}{f_j}$. From Equation 5.7, when $T \gg \tau$, Ω can be very large thus increasing energy consumption (see Appendix C for details). In order to minimise Ω and therefore reduce energy consumption, the toggling period of a network should be increased to TP_{max} as soon as the event data has been reported to the base station.

In order for Equation 5.6 to be useful in applications in which the event occurrence time is unknown, θ can be estimated using historical data. This estimation should place more value on newer measurements in comparison with older measurements so that the toggling period can be adjusted more efficiently when an event occurs.

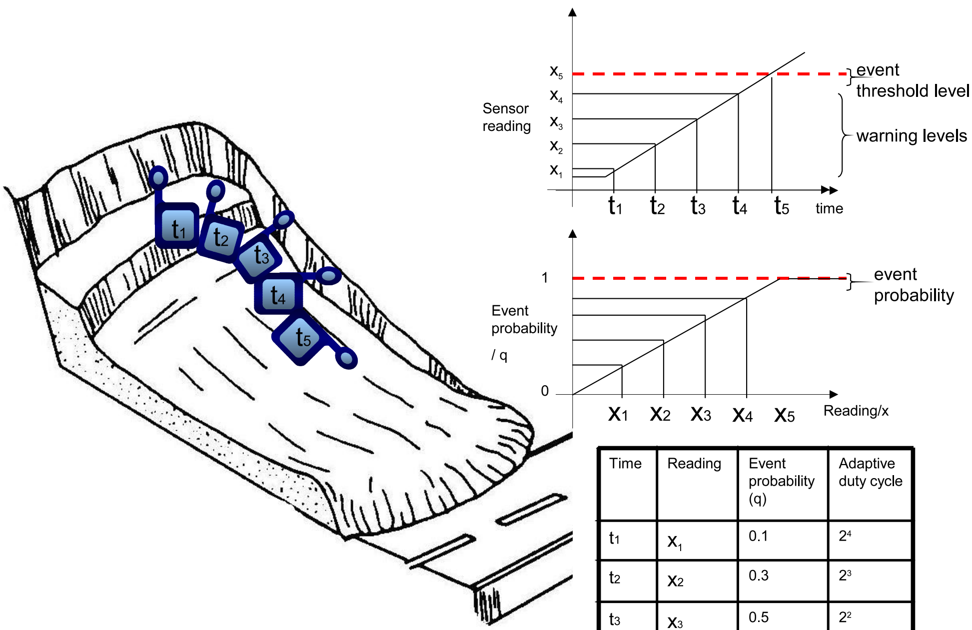
These requirements are satisfied by an Exponentially Weighted Moving Average (EWMA). EWMA applies exponentially increasing weight factors to more recent parts of historical data while simultaneously decreasing weight factors on older data points. An EWMA model with a weight factor, α, can be applied to the *toggling period adaptation function* using the equation illustrated below:

$$\bar{\theta}_t = (\alpha)\theta_t + (1-\alpha)\bar{\theta}_{t-1} \tag{5.8}$$

Where $0 \leq \alpha \leq 1$, $\bar{\theta}_t$ is the estimated value of θ at time t and θ_t is the most recent value of the event occurrence rate at time t.

5.5.4 *Overview of ADAMAC*

Figure 5.10 illustrates the operation of ADAMAC during an embankment failure.



Time	Reading	Event probability (q)	Adaptive duty cycle
t_1	x_1	0.1	2^4
t_2	x_2	0.3	2^3
t_3	x_3	0.5	2^2
t_4	x_4	0.7	2^1
t_5	x_5	1.0	2^0

Figure 5.10: Minimising end-to-end delay during an embankment failure requires the use of warning levels which inject duty-cycle policies into the network before an event occurs. The nodes used in this picture are courtesy of Senceive Ltd.

As the embankment collapses at a particular rate θ between time t_1 to t_5, ADAMAC adapts the communication frequency in order to minimise end-to-end delay and limit energy consumption. At t_1, the collected reading is unchanged because the embankment has yet to start moving. Between t_2 and t_4, the embankment steadily collapses as revealed through rising readings and thus causes higher event probabilities. Therefore at times t_2, t_3 and t_4 corresponding to three different warning levels, ADAMAC calculates a toggling period which increases the communication frequency of the network as an event becomes more likely. At t_5 when an event occurs, $q = 1$ and an event notification packet is reported with minimal delay towards the base station.

More details on the operation of ADAMAC are presented in the pseudo-code of ADAMAC shown in Algorithm 4 below. After measuring a reading X_t, Algorithm 4 changes a node's toggling frequency to a new toggling frequency,f_i.

To determine f_i, the values of X_{event}, X_t, M, h, θ, TP_{max} and ρ must first be obtained during initialisation. When the event occurrence rate, θ, is unknown, an estimated event occurrence rate can be calculated using Equation 5.8. Next the event occurrence probability, q, corresponding to X_t is calculated using algorithms such as DPPS or eSENSE (see Line 2 of Algorithm 4). This event occurrence probability is then applied to an `adaptDutyCycle Function` as shown in Line 3. If $q = 1$, an event is imminent therefore the `adaptDutyCycle Function` applies an adaptation policy of $\phi = 0$ corresponding to a Fully Active toggling frequency in order to limit end-to-end delay (see Lines 7 - 9). Conversely if $q < 1$, the toggling frequency, f_i, corresponding to $\phi = \phi_b$ is obtained using Equation 5.6 (see Lines 10 - 11). The value of this toggling frequency is optimal in the sense that it maximises energy savings while avoiding *breakdown*. In order to minimise energy savings, after a sensor node has spent ρ seconds using the same toggling frequency, a `rampDownStatus Function` as shown in Line 5 is used to increase the toggling period of a network towards TP_{max} (see Lines 13 - 27 for more details). After the `rampDownStatus Function` has been implemented, the new toggling frequency is updated as shown in Line 5 and then broadcast to neighbouring nodes. Nodes that receive differing values of f_i compare these and adopt the one with the highest toggling frequency.

input : X_t
output: f_i

1 Initialise X_{event}, M, h, θ,TP_{max},ρ
2 $q_t \leftarrow$ `DPPS` (X_t,X_{event});
3 $f_i \leftarrow$ `adaptDutyCycle` (X_t,X_{event},M,h,θ,q_t);
4 $f_{array}(t) \leftarrow f_i$;
5 $f_i \leftarrow$ `rampDownStatus` (f_{array}, f_i, ρ);
6 `adaptDutyCycle` ()
7 **if** *q==1* **then**
8 $\quad \phi \leftarrow 0$;
9 **else**
10 $\quad \phi \leftarrow \frac{\log\left(1 - \frac{\log\left(\frac{X_{event}-X_t}{\theta(h-1)}\right)}{(M-1)\log 2}\right)}{\log(q)}$
11 $\quad f_i \leftarrow \frac{1}{\beta(q_t,\phi)}$
12 **end**
13 `rampDownStatus` ()
14 $rampDown \leftarrow 0$;
15 **for** *time* $\leftarrow$ *t-ρ to t* **do**
16 $\quad$ **if** f_i==f_{array}*(time)* **then**
17 $\quad\quad rampDown \leftarrow rampDown + 1$;
18 $\quad$ **else**
19 $\quad\quad rampDown \leftarrow 0$;
20 $\quad$ **end**
21 **end**
22 **if** $rampDown$==ρ **then**
23 $\quad$ **if** $\frac{1}{f_i}$ == TP_{max} **then**
24 $\quad$ **else**
25 $\quad\quad f_i \leftarrow \frac{f_i}{2}$;
26 $\quad$ **end**
27 **end**

Algorithm 4: Adaptive Detection-driven Ad hoc Medium Access Control (ADAMAC)

5.6 ADAMAC Simulation Setup, Results and Analysis

This section presents a comparison and evaluation of the performance of ADAMAC compared with SMAC and a Fully Active (FA) network. Relevant parameters were obtained from a Chipcon CC2420 radio and simulations were carried out using MATLAB v7.0.1. A 2-dimensional unit-disk graph model in a 100 square meter area was used in all experiments. As outlined in Section 3.6.2 of Chapter 3, it is assumed that nodes operate in an open lossless space with no physical obstructions and that each node knows the location of neighbouring nodes.

To facilitate event reporting, a unique message is broadcast from an event source node which, when received by a base station, alerts users of the occurrence of an event. An event occurs when a reading is greater than or equal to $X_{event} = 31$. The EWMA equation, $0.85\bar{\theta} + 0.15\theta_t$, with $M = 7$ were used because experimental tests showed that they produced the best results in regard to limiting energy consumption and end-to-end delay.

The performance metrics used to evaluate ADAMAC, SMAC and FA not only include *energy consumption* and *end-to-end delay*, but also include *event detection time*:

- End-to-end delay (seconds): The amount of time between an event occurring and the event being reported at a destination base station
- Event detection time (seconds): The amount of time between an event occurring and the event being detected. Figure 5.11 further illustrates the distinction between event detection time and end-to-end delay.
- Energy consumption (Joules): Energy consumed by a sensor node during sensor operation, calculated using a first order radio model

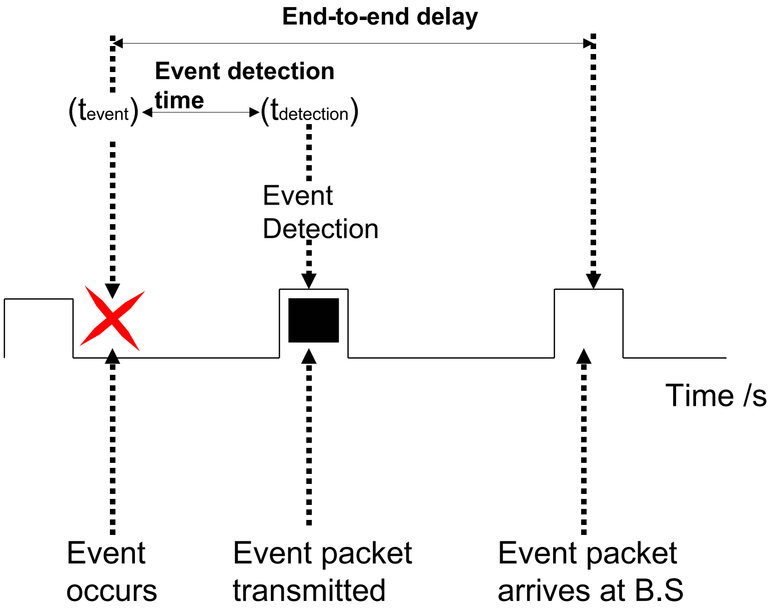


Figure 5.11: The distinction between end-to-end delay and event detection time

5.6.1 The effects of breakdown on delay and energy consumption

This section explores the effect of *breakdown* on end-to-end delay. The simulations were carried out using a linear chain of nodes; the source and destination nodes were located at the beginning and end of the chain.

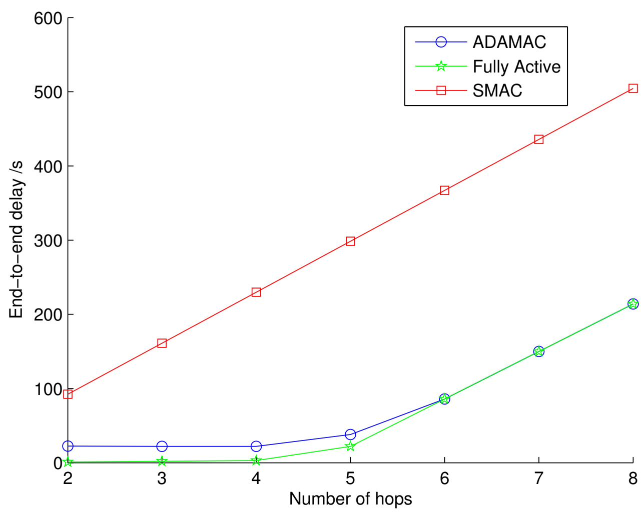


Figure 5.12: The effect of hop count on end-to-end delay

Simulations were carried out to show the effects of increasing the number of hops between an event node and a base station on end-to-end delay and energy consumption. At a fixed event occurrence rate of $\theta = 0.084$, Figure 5.12 demonstrates that as the network size increases, *breakdown* occurs immediately using SMAC, and after five hops using ADAMAC and FA. Figure 5.12 illustrates that using ADAMAC, the CBR occurs after five hops, four hops more than is possible with SMAC. This means that delay is substantially reduced when using ADAMAC. Furthermore, even after *breakdown*, ADAMAC reduces end-to-end delay by over 200 seconds in comparison to SMAC for the same network.

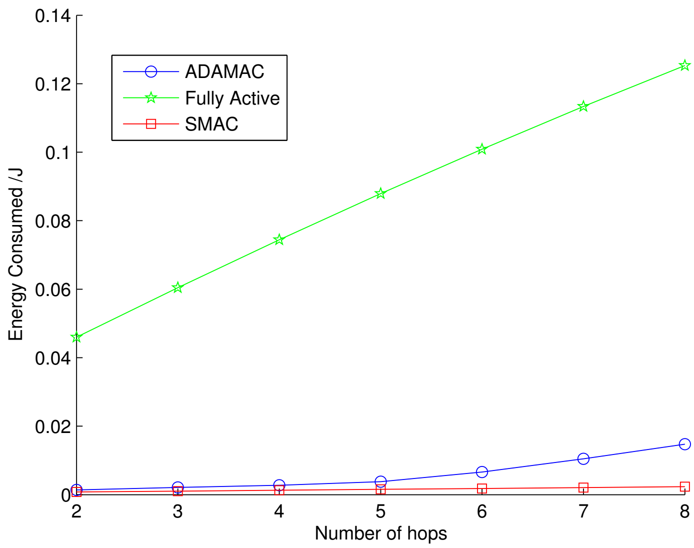


Figure 5.13: The effect of hop count on energy consumption

Figure 5.13 demonstrates that before five hops, the energy consumption using ADAMAC is almost as low as levels obtained using SMAC. Because *breakdown* has not occurred when the number of hops is less than five, ADAMAC can limit both end-to-end delay and energy consumption efficiently. After five hops, *breakdown* causes both energy consumption and delay to increase. Inspite of the increase in energy consumption, ADAMAC uses significantly less energy in comparison with FA.

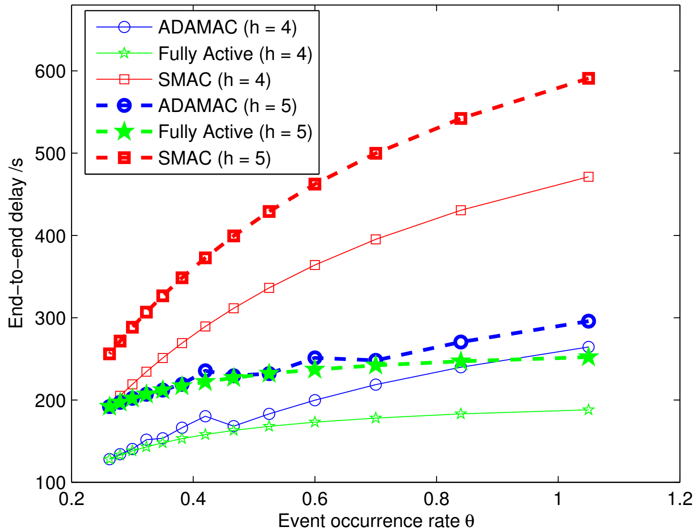


Figure 5.14: End-to-end delay variation with network size

Figure 5.14 also demonstrates the advantages of using ADAMAC; as the event occurrence rate increases beyond the CBR, using ADAMAC leads to a smaller delay than using SMAC when networks with four or five hops are used. It can also be observed that the increase in delay caused by increasing the number of hops from four to five hops affects ADAMAC less than SMAC. Thus Figure 5.14 shows that delay incurred by ADAMAC for a network with five hops still offers an improvement over SMAC for a network with four hops. This means that ADAMAC is less prone to the effects of the Data Forwarding Interruption problem and can therefore reduce end-to-end delay substantially in comparison to SMAC. It should also be noted that for networks with either four or five hops, ADAMAC's end-to-end delay is minimised to levels comparable to FA when $\theta < 0.4$ because *breakdown* has yet to occur.

5.6.2 The effect of event occurrence rate in a large network

More simulations were carried out to discover the effect of event occurrence rate on both event detection time and end-to-end delay in large networks; networks with over 50 nodes. The simulation results obtained in this section used networks of random topology containing 60 and 80 nodes deployed in a 100 m^2 plane. Results shown correspond to a mean of all simulations in which nodes use a fixed communication range of 30 m to form a connected network.

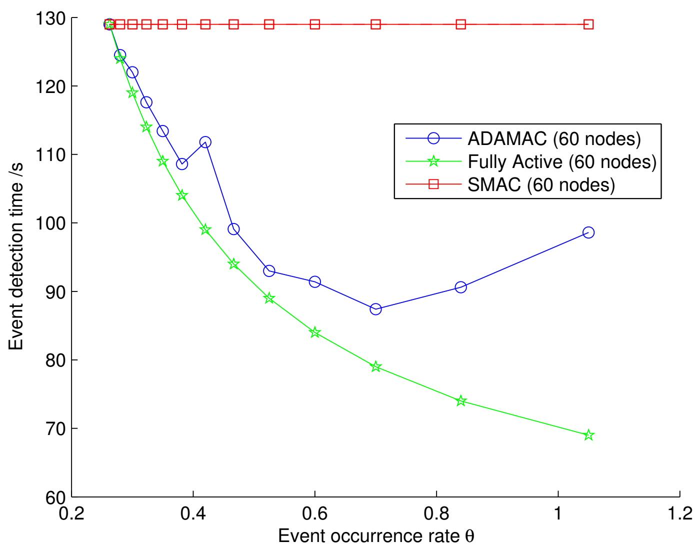


Figure 5.15: The effects of the event occurrence rate on event detection time in a network containing 60 nodes

Figure 5.15 shows that ADAMAC has a shorter event detection time in comparison to SMAC; after an event occurs, the event is detected earlier using ADAMAC in comparison with SMAC. As θ increases to between $\theta = 0.2625$ and $\theta = 0.7$, the event detection time decreases. This is because the event occurs shortly before the sensor becomes active and is therefore detected sooner. Although ADAMAC has a shorter event detection time than SMAC, the event detection time starts to increase after $\theta = 0.7$. This increase happens because the event occurs just after a node has gone to sleep and the event is therefore not detected until the next active cycle of the node.

Simulations were also done to examine the effects of the event occurrence rate on end-to-end delay and energy consumption. Figure 5.16 reveals that using ADAMAC reduces end-to-end delay in comparison with SMAC. Although ADAMAC and SMAC

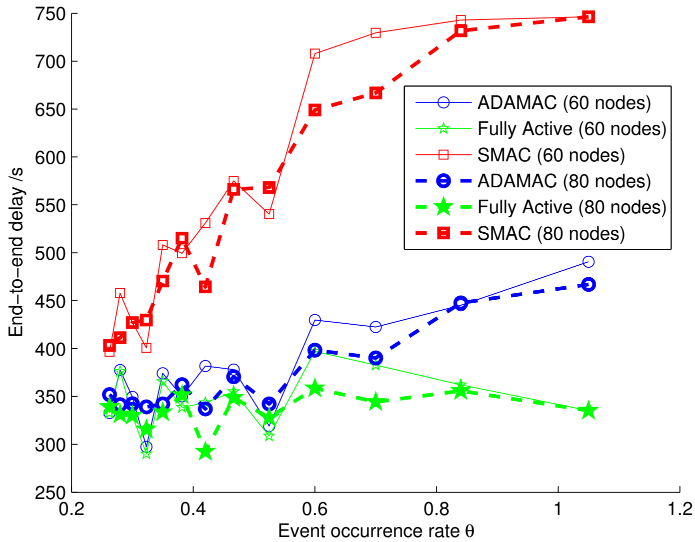


Figure 5.16: The effects of event occurrence rate on end-to-end delay in networks containing 60 and 80 nodes

have the same event detection time at $\theta = 0.2625$, as can be seen in Figure 5.15, Figure 5.16 reveals that ADAMAC, at $\theta = 0.2625$, has a shorter end-to-end delay. This is because the adaptive duty cycling technique employed in ADAMAC uses more warning levels in comparison to FA and SMAC as shown in Figure 5.17. In order for relevant events to be reported to a base station with limited end-to-end delay, these warning levels increase the duty cycle of a network before an event occurs.

Another advantage of ADAMAC's adaptive duty cycling technique is that as the event occurrence rate increases, ADAMAC offers progressive performance improvement with regard to end-to-end delay. As an example, in Figure 5.16 when $\theta = 0.2625$, the difference in end-to-end delay between ADAMAC and SMAC is 1% in a 60 node network but by $\theta = 1.05$ this difference increases to over 40%. It is also evident from Figure 5.16 that as the density of nodes increases from 60 to 80, end-to-end delay decreases. This is because as the number of nodes increases, more connections are formed between nodes which offer shorter pathways to the base station thereby reducing the delay.

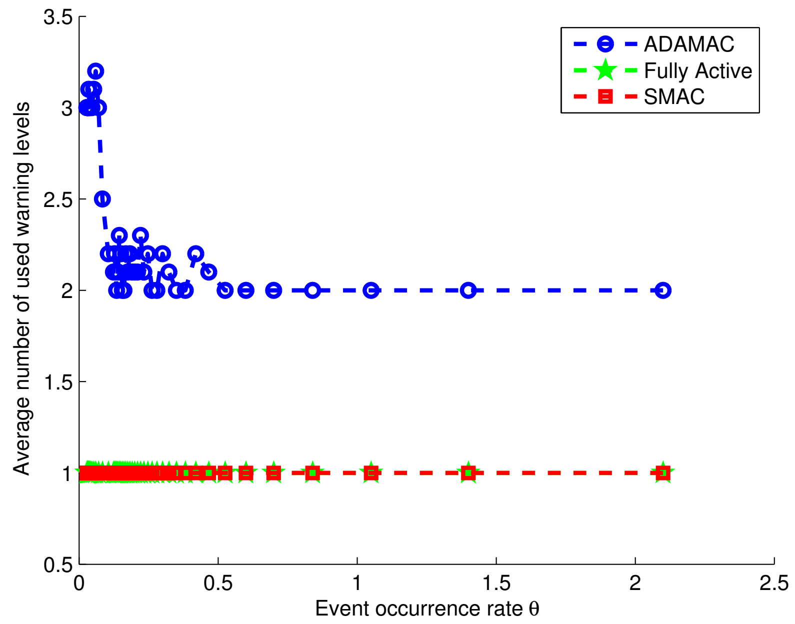


Figure 5.17: Average number of warning levels used in a network with 80 nodes

Figure 5.18 shows the energy consumption at various event occurrence rates in networks with 60 and 80 nodes. Although ADAMAC consumes more energy than SMAC, ADAMAC limits energy consumption in comparison to FA by over 90% in all event occurrence rates. This is possible because after the event has been reported to the base station, the network's duty cycle is increased to TP_{max} leading to increased energy savings. Figure 5.18 also shows that a network with 80 nodes consumes more energy in comparison with a network with 60 nodes. Nevertheless increased energy consumption between ADAMAC and SMAC is not substantial when compared with the increase observed between FA and SMAC. It can be concluded therefore that ADAMAC limits the effects the Data Forwarding Interruption problem while simultaneously limiting energy consumption. To further outline the benefits of ADAMAC, the next section examines the effects of node density on a sensor network in more detail.

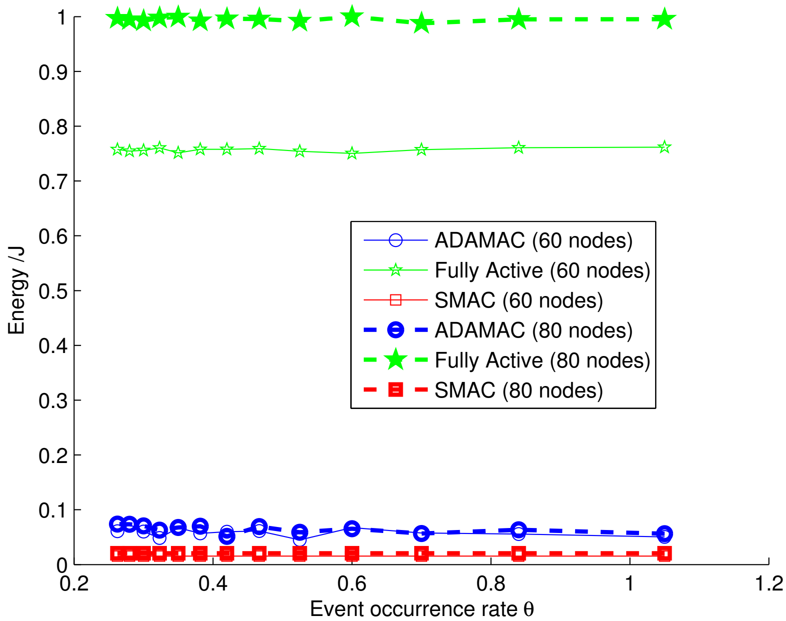


Figure 5.18: The effects of event occurrence rate on energy consumption in networks containing 60 and 80 nodes

5.6.3 The effect of network density on delay and energy consumption

This section examines the effect of an increase in the number of nodes on the performance of a network. Simulations were carried out using two different topologies. The first used a source and base station pair at random locations; the second used a source and base station pair at fixed locations diagonally across opposite ends of a square deployment area.

Figure 5.19 shows the results for the first case where the source and destination base station locations were random. Using a random source and destination has the effect of averaging the number of hops between a source and destination. End-to-end delay noticeably increases between 40 and 50 nodes when $\theta = 0.2625$ because the effects of the Data Forwarding Interruption problem are more prevalent in larger networks where convergence to a duty cycle can take longer in comparison with smaller networks. While it can be observed that SMAC only reduces the delay by less than 10% as the

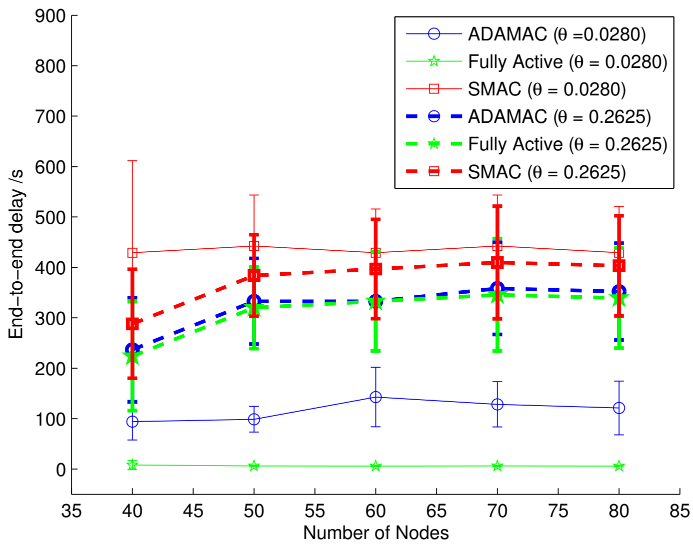


Figure 5.19: The effect of node density on end-to-end delay using a random source and destination pair

event occurrence rate decreases from $\theta = 0.2625$ to $\theta = 0.0280$, within this same event occurrence range ADAMAC decreases end-to-end delay by 200%.

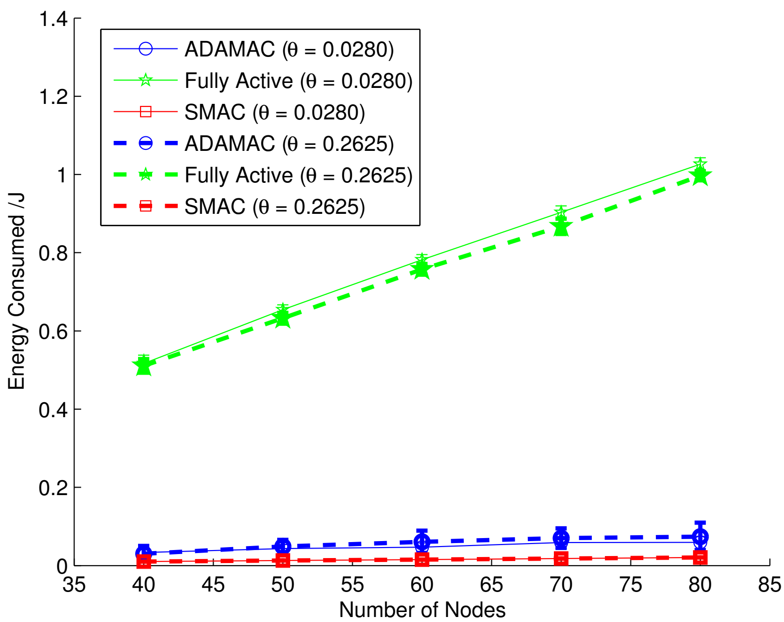


Figure 5.20: The effect of node density on energy consumption using a random source and destination pair

The effect of node density on energy consumption where the source and base station locations were random is shown in Figure 5.20. Energy consumption using ADAMAC is marginally higher than SMAC, but significantly lower than FA. Furthermore as the number of nodes increases in ADAMAC, the rate of increase in energy consumed is substantially lower than FA and only fractionally higher than SMAC. The fact that this result is obtained across varying network sizes and event occurrence rates demonstrates that ADAMAC is a scalable data collection tool.

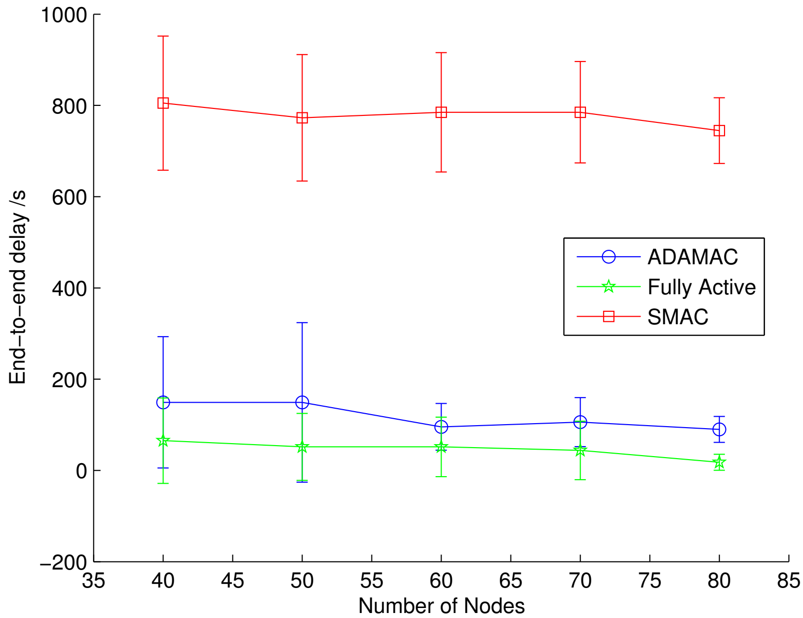


Figure 5.21: The effect of node density on end-to-end delay using a source and destination pair at a fixed located at $\theta = 0.0280$

Figure 5.21 presents end-to-end delay for an event occurrence rate of $\theta = 0.0280$ when the location of the source and base station are fixed at opposite ends of the diagonal of a square deployment area in order to elongate the network's diameter. As the number of nodes increases, the end-to-end delay decreases because an increased number of nodes make more connections forming shorter paths between the source node and the base station, thus allowing messages to be transmitted more quickly. It may also be noticed that values of end-to-end delay in Figure 5.21 are higher than values in Figure 5.19.

This is because the average distance between the source and base station of the fixed network is higher than the average distance of the random network.

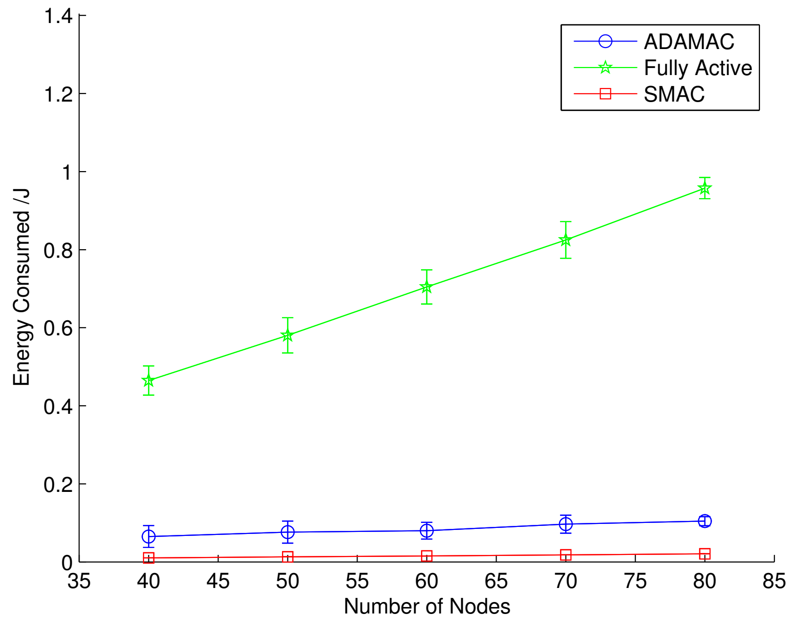


Figure 5.22: The effect of node density on energy consumption using a source and destination pair at a fixed located at $\theta = 0.0280$

The effect of node density on energy consumption where the locations of source and base station are fixed is shown in Figure 5.22. As the number of nodes increases, energy consumption using all protocols increases. Figure 5.22 also shows that energy consumption using ADAMAC in a fixed network is higher than energy consumed in a random network (see Figure 5.20). This increase in energy consumption is due to the increased end-to-end delay in a fixed network caused by the elongated distance between the source and destination. This leads to a network spending more time and energy for event reporting. In spite of this increased energy consumption, ADAMAC still has significantly higher energy savings in comparison to FA.

5.6.4 The effect of packet loss on delay performance

This section investigates the effect of packet loss on end-to-end delay and energy consumption. The packet loss ratio is indicative of the efficiency of packet transmission from an event node to a base station and is an indicator of the extent of the inefficiencies caused by the Broadcast Storm problem in a data collection protocol. In order to demonstrate ADAMAC's robustness to packet loss, simulations were first done using a 50 node network at an event occurrence rate of $\theta = 0.0280$. These simulations illustrated the effect of packet loss on ADAMAC's performance in terms of energy and end-to-end delay. Secondly, simulations were also carried out to compare the energy consumption and end-to-end delay of ADAMAC, SMAC and FA using different packet loss percentages at various event occurrence rates.

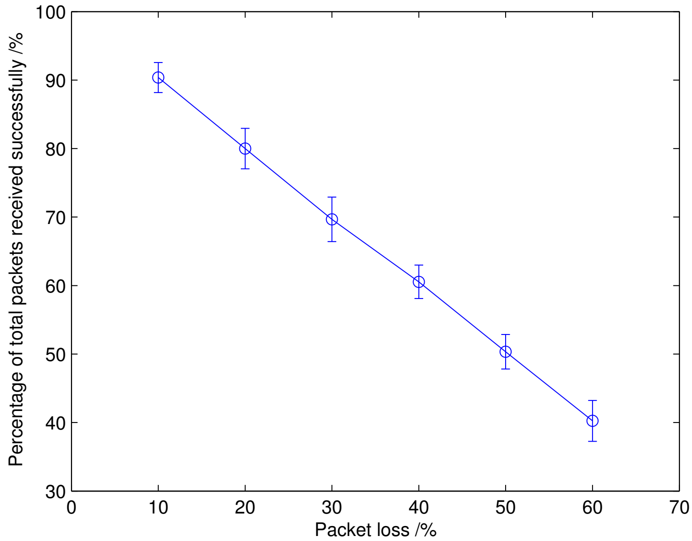
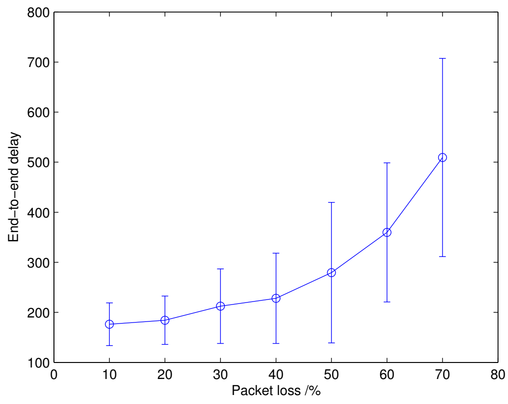


Figure 5.23: Variation of successful broadcasts with packet loss percentages

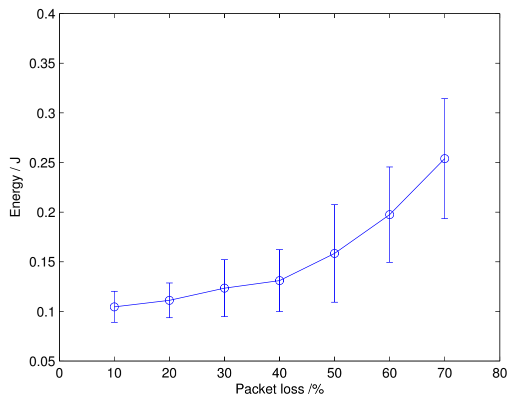
Figure 5.23 shows the effect of packet loss percentages on proportion of total packets which are successfully received in a network using ADAMAC. As expected, the percentage of total packets successfully broadcast to a destination using ADAMAC decreases as the packet loss percentage increases. Therefore a 10% packet loss means that around 90% of all packets are successfully received at a destination whereas a 60%

packet loss means that about 40% of all packets are successfully received. This means that the effect of the Broadcast Storm problem becomes more detrimental as packet loss percentages increase.

Simulations results shown in Figure 5.24 reveal that the effect of packet loss on end-to-end delay and energy consumption using ADAMAC worsens with increasing packet loss percentage. For example Figure 5.24(a) shows that over the same packet loss percentage range, end-to-end delay is increased from below 200 seconds at a packet loss percentage of 10% to over 500 seconds at a packet loss percentage of 70%. Similarly Figure 5.24(b) reveals that energy consumption is more than doubled between 10% and 70% packet loss percentages. The increase in both end-to-end delay and energy consumption is caused by the increased number of rebroadcasts required to report event data successfully to a base station.



(a) The effect of packet loss percentages on end-to-end delay using ADAMAC ($\theta = 0.0280$)



(b) The effect of packet loss percentages on energy consumption using ADAMAC($\theta = 0.0280$)

Figure 5.24: The effect of packet loss on delay and energy consumption

Figure 5.25 shows the number of packets lost using ADAMAC, SMAC and FA in networks with packet loss percentages of 10% and 30% respectively. As expected, increasing the packet loss percentage from 10% to 30% increases the number of packets lost in all three protocols. However, because ADAMAC adjusts the toggling period of a network in order to receive and transmit more data when an event occurs, less data packets are lost in comparison to FA.

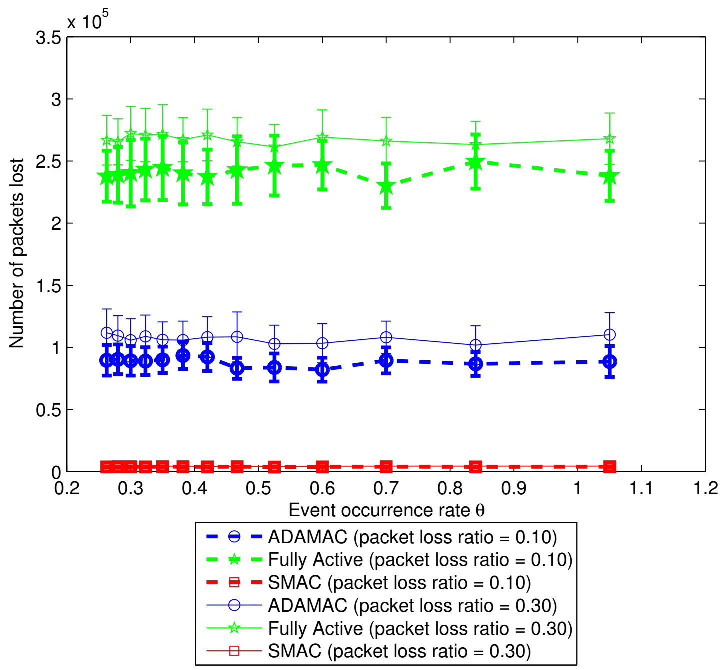


Figure 5.25: Number of packets lost using 10% and 30% packet loss percentages at varying event occurrence rates

Although more packets are lost using ADAMAC than using SMAC, Figure 5.26 shows that end-to-end delay minimisation with ADAMAC is still superior to that of SMAC and only marginally higher than end-to-end delay with FA. Figure 5.26 also shows that as the event occurrence rate increases, end-to-end delay in ADAMAC increases at a slower rate than in SMAC. End-to-end delay between $\theta = 0.2625$ and $\theta = 1.05$ increases by 100 seconds using ADAMAC. This is significantly lower than the increase of 900 seconds obtained with SMAC over the same event occurrence range. Further examination of Figure 5.26 reveals that at packet loss percentages of both 10% and 30%, ADAMAC reduces end-to-end delay by up to 50% in comparison to SMAC.

Indeed this is still the case when end-to-end delay at a packet loss percentage of 30% in ADAMAC is compared with the end-to-end at a packet loss percentage of 10% in SMAC; ADAMAC with 20% more packet loss consistently provides significantly lower end-to-end delay when compared with SMAC. These results confirm that ADAMAC is a more robust data collection tool not only in terms of alleviating the Data Forwarding Interruption problem, but also in terms of minimising the Broadcast Storm problem.

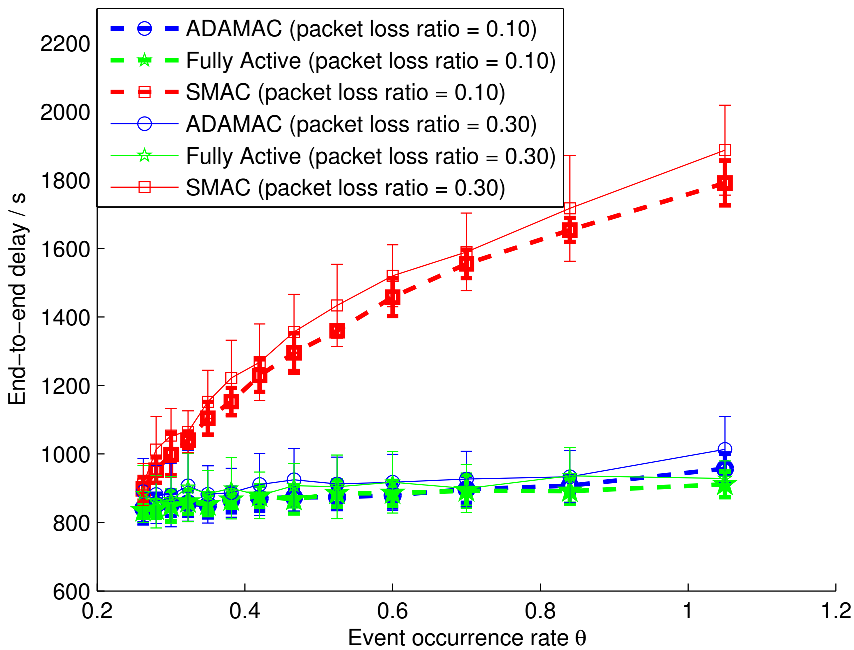


Figure 5.26: End-to-end delay using 10% and 30% packet loss percentages at varying event occurrence rates

As expected, Figure 5.27 reveals that energy consumption increases in all protocols as packet loss increases from 10% to 30% because more energy is used for retransmissions. Figure 5.27 also reveals that ADAMAC reduces energy consumed in comparison to FA.

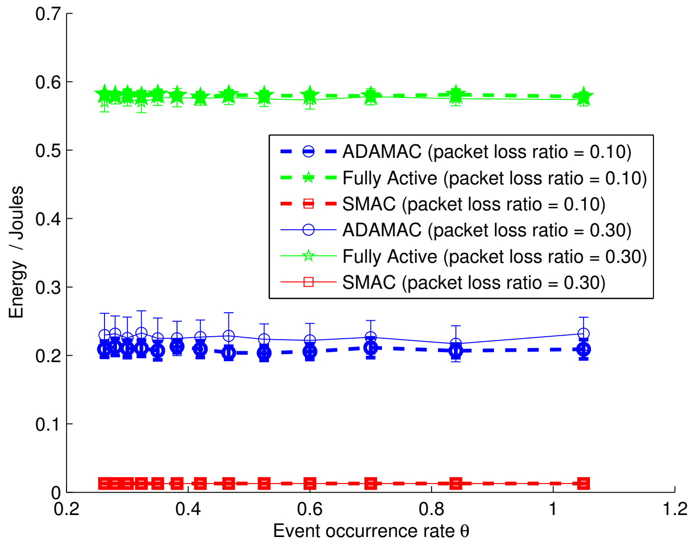


Figure 5.27: Energy consumption using 10% and 30% packet loss percentages at varying event occurrence rates

5.7 Chapter Summary

This chapter proposed an Adaptive Detection-driven Ad hoc Medium Access Control algorithm (ADAMAC) as a solution to the Data Forwarding Interruption problem, a problem affecting periodic scheduling protocols such as SMAC. Using early warning event signalling produced from historical data, ADAMAC adapts the duty cycle of a network prior to the onset of an event. By increasing the communication frequency using a *toggling period adaptation function*, ADAMAC reduces end-to-end delay. Simulation results revealed that end-to-end delay is limited using the proposed ADAMAC in comparison with the SMAC protocol. Furthermore, results also revealed that ADAMAC limits this end-to-end delay to levels obtained in a Fully Active network, a network with optimum end-to-end delay characteristics, while simultaneously using only a small fraction of the energy consumed in a Fully Active network. Reduced delay was obtained under of a variety of network conditions including packet loss and node density. This demonstrates that ADAMAC is not only a scalable solution to the Data Forwarding interruption problem, but is also a robust monitoring tool which can facilitate data collection in wide array of environmental applications.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This thesis addresses the problem of energy limitations affecting sensor-based networks. Protocols were developed to increase the lifetime of a network by decreasing energy consumption; techniques were also developed to decrease delay during the dissemination of data without substantially increasing energy consumption. It was argued that by applying these techniques in monitoring programmes requiring data collection and dissemination, the deployment of Wireless Sensor Networks could be extended into wider application areas than had hitherto been possible.

After reviewing model-based techniques for improving energy efficiency, a new data collection protocol was designed: a Dual Prediction and Probabilistic Scheduler (DPPS). DPPS amalgamates ideas from probability theory and time series prediction using a stochastic scheduler in order to conserve energy whilst maintaining accuracy of results.

Simulations were carried out comparing DPPS with eSENSE, another stochastic scheduler that reduces energy consumption. Simulation results indicated that DPPS offered improved performance over eSENSE in terms of reducing the expected miss ratio (an estimate of the number of undetected events). Whilst offering these tighter missed ratio guarantees, DPPS also offered reductions of up to 35% in sensor usage compared with eSENSE, saving more energy and thus increasing the lifetime of the network. Results also revealed that the mean square error could be explicitly controlled so that it satisfied a user's quality requirements. Preliminary experimentation of DPPS was carried out on a real system using several Microchip PICDEMZ prototype boards in order to demonstrate the feasibility of DPPS as a data collection tool.

Neither DPPS nor eSENSE however consider improving data collection efficiency in a multi-hop sensor network by reducing delay between a source and a base station. Thus in Chapter 5 an Adaptive Detection-driven Ad hoc Medium Access Control (ADAMAC) algorithm was developed. ADAMAC increased data collection efficiency by using event probability, as obtained from algorithms such as DPPS and eSENSE in Chapter 4, to trigger early event warnings. When received, these warnings adapt the duty cycle of a sensor so events are reported faster across a network. Despite these advantages, Chapter 5 also demonstrates that as the event occurrence rate increases, duty cycling protocols such as ADAMAC may *breakdown* causing end-to-end delay to increase rapidly because a message can only be relayed through a finite number of hops before it encounters a node that will fall asleep.

Thus, an analysis of the conditions leading to ADAMAC *breakdown* was also carried out in Chapter 5. Results showed that although *breakdown* cannot always be avoided as the event occurrence rate increases, ADAMAC extends the critical breakdown rate (CBR) beyond that which is obtained in SMAC, an alternative data collection protocol which uses periodic sampling. Furthermore ADAMAC counteracted the effects of the Data Forwarding Interruption problem better than SMAC, approaching the optimal performance obtained using a Fully Active sensor schedule in terms of end-to-end delay.

ADAMAC also improved the Event Detection Time, the duration between an event's occurrence and the event's detection by a source node, by up to 30% compared with SMAC. Additionally, end-to-end delay is reduced by a third in comparison with SMAC because at the onset of an event, the network has a higher duty cycle and consequently a higher communication frequency for transmitting reports.

Simulations in Chapter 5 also demonstrated that the effect of packet loss was to increase both end-to-end delay and energy consumption as a result of necessary retransmissions in ADAMAC and SMAC. However when both systems suffer a 30% packet loss, ADAMAC reduced the end-to-end delay by 80% in comparison to SMAC. Indeed, even when ADAMAC had 20% more packet loss than SMAC, a 70% improvement in the end-to-end delay could be observed over SMAC.

ADAMAC was demonstrated to be scalable when used in dense networks of up to 80 nodes. In such networks, the increased number of nodes can adversely affect the energy consumption. Although an increase of node density led to an increase in energy consumption in all algorithms, this increase was less marked in ADAMAC because a ramp down mechanism was used to adaptively decrease the duty cycle of a node once an event has been successfully reported.

The development of these more energy efficient monitoring protocols will allow WSNs to be deployed in wider application areas especially in networks where energy constraints currently limit the effectiveness of data collection.

6.2 Future Work

Owing to time restrictions it was not possible to implement and deploy sensors programmed with DPPS or ADAMAC in a real life environmental monitoring context. It was only possible to evaluate their efficiency using simulations. Future work would therefore involve the deployment of DPPS and ADAMAC in real life monitoring programmes. During the preliminary experiments conducted with Microchip prototype boards the following practical challenges were highlighted and therefore could be addressed in future work:

1. Packet collision - A robust collision avoidance mechanism could be developed in order to further improve the efficiency of the data collection process by dealing with the problem of packet collision. During experiments, as a means of reducing packet loss, a random time was added to the scheduling interval so that no two nodes transmitted data simultaneously. However, some packet collisions still occurred leading to a partial loss of data.

2. Synchronisation - During the preliminary experiment it was discovered that the clocks onboard each node were not synchronised as a consequence of the fact that prototypes were being used which contained hand-soldered components. This caused varying degrees of inaccuracy in the onboard watch crystals, thus leading to small inconsistencies in results. Future work could be done to eliminate these inaccuracies by assuring the synchronisation of all nodes.

3. Outliers and data corruption - Approximately 1 % of packets received contained corrupted data such that temperature and timing information contained erroneous values. While this problem is not critical, it could be addressed in future work in order to further improve the efficiency of the protocol.

Another area of research which could be investigated in the future is the area of false alarms. False alarms can occur when a faulty node produces a false reading, and therefore a false alert. Such false alerts could perhaps be minimised by co-ordinating nodes with each other in order to detect faulty nodes or by issuing trust levels to each node based on the number of correctly reported alarms.

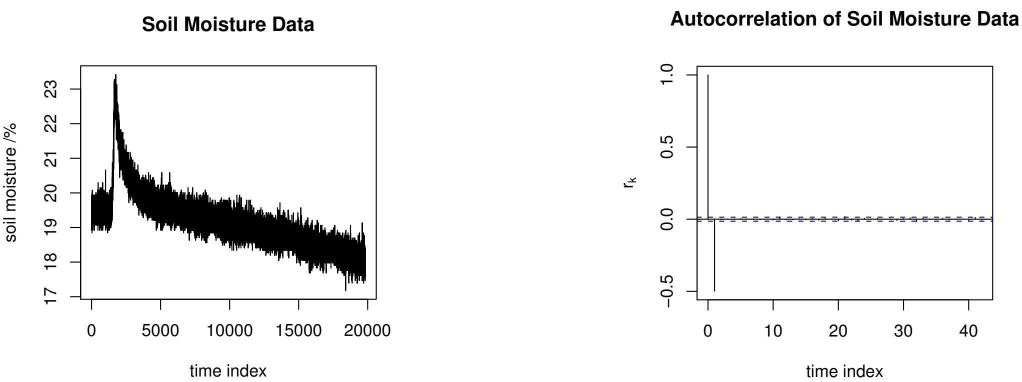
Owing to a lack of spatio-temporal datasets, it was not possible to explore the correlation between readings from neighbouring sensors fully. More datasets, such as those pertaining to tilt angle during an embankment failure or a landslide, if obtained, would therefore be useful. For example, as event occurrence becomes more likely, neighbouring sensors are likely to exhibit similar trends which allow the event occurrence time to be determined more accurately. This in turn allows early warning signals to be issued more accurately thus enhancing ADAMAC's performance. An analysis of spatial data could also significantly improve the miss ratio of DPPS.

Currently one of the assumptions during simulations is that sensors have infinite transport buffers for packet storage. A more realistic scenario would be to assume a finite buffer size for the transport queue. This would mean that some packets are dropped and never relayed to the destination. It could also introduce more complexity in the MAC layer, for example if a packet acknowledgement is needed. Investigating how adapting the transport buffer size affects ADAMAC and DPPS could further enhance their effectiveness during deployment in real life scenarios.

Appendix A

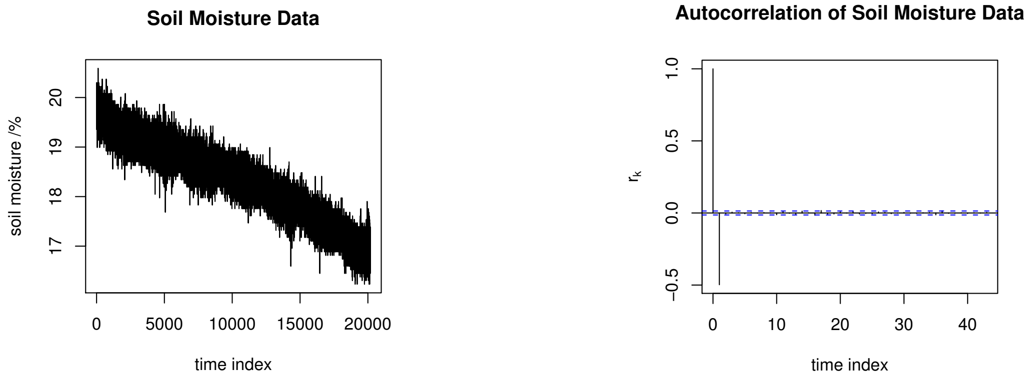
Integrated Moving Average Model

In time series analysis, determination of the order of a Moving Average (MA) process is achieved by the evaluation of the Auto-Correlation Function (ACF) for a given dataset. Theoretically, an MA(q) process is identifiable from its ACF because it cuts off after lag q. In order words after q lags, the values of autocorrelation function, r_k, should become negligibly close to 0. An Integrated Moving Average (IMA(q)) process of order q is akin to an MA(q) process where the data is differenced. Figures A.2-A.3 show the autocorrelation function of various differenced datasets.



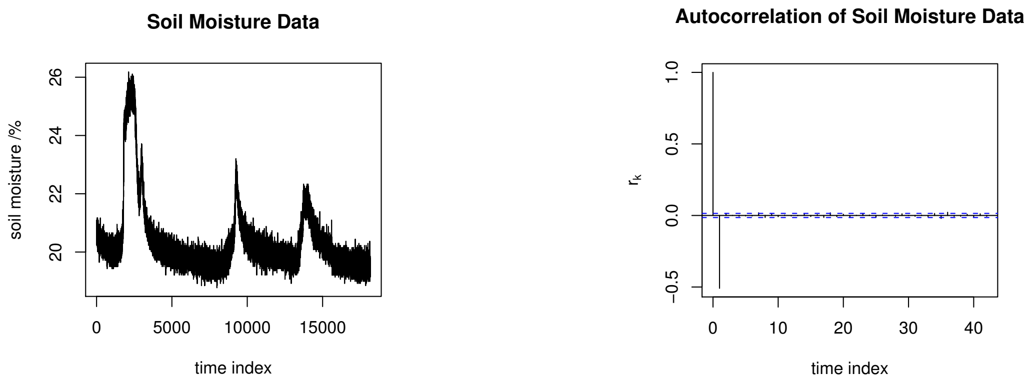
(a) Time domain plot (b) Autocorrelation function

Figure A.1: Autocorrelation function of soil moisture (dataset 1)



(a) Time domain plot (b) Autocorrelation function

Figure A.2: Autocorrelation function of soil moisture (dataset 2)

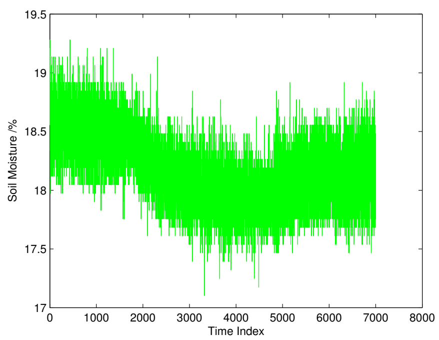


(a) Time domain plot (b) Autocorrelation function

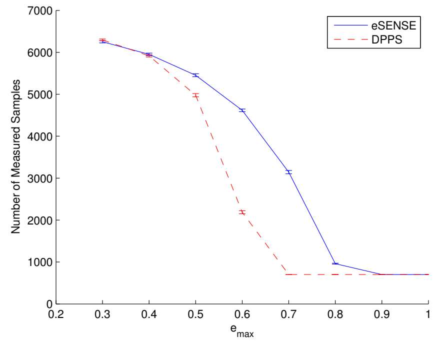
Figure A.3: Autocorrelation function of soil moisture (dataset 3)

Appendix B

Supplementary Datasets

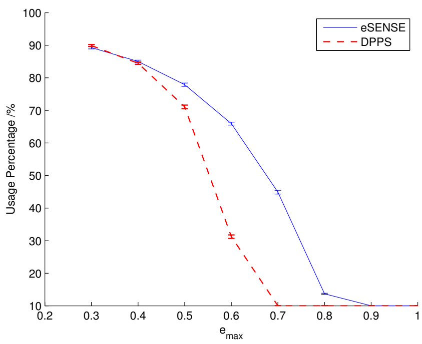


(a) Sampled soil moisture data

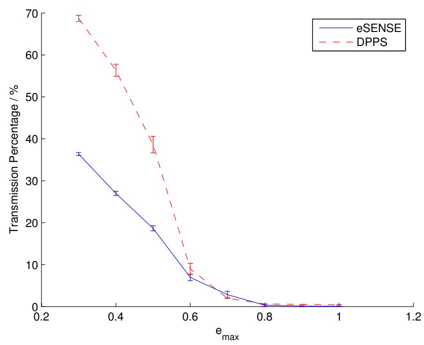


(b) Number of measurements using DPPS and eSENSE

Figure B.1: Data collection using DPPS and eSENSE protocols ($F_N = 5\%$)

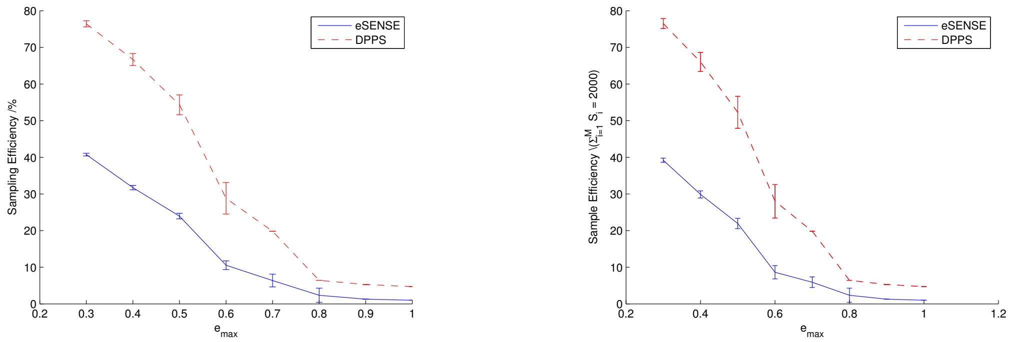


(a) Usage Percentage of DPPS compared to eSENSE



(b) Transmission percentage of DPPS compared to eSENSE

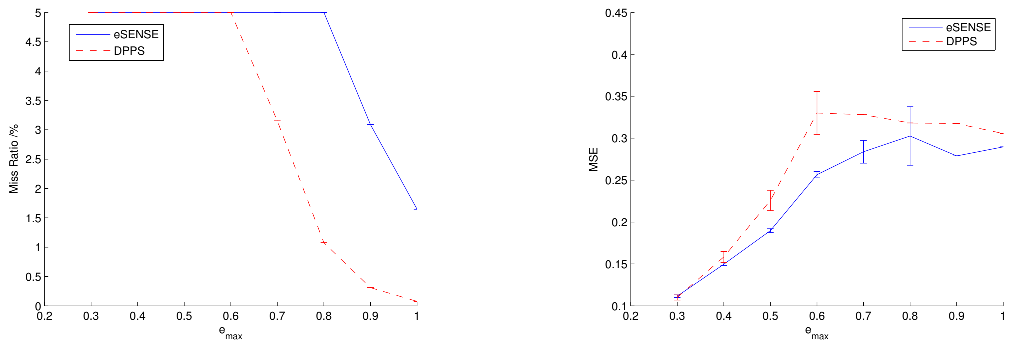
Figure B.2: Usage and transmission percentages of DPPS and eSENSE ($F_N = 5\%$)



(a) Sampling efficiency of DPPS compared to eSENSE calculated over time

(b) Sampling efficiency of DPPS compared to eSENSE calculated after 2000 measurements

Figure B.3: Sampling efficiency of DPPS and eSENSE when $F_N = 5\%$



(a) Expected miss ratio of DPPS and eSENSE

(b) Mean square error of DPPS and eSENSE

Figure B.4: Miss ratio and mean square error of DPPS and eSENSE ($F_N = 5\%$)

Appendix C

Transition time and the Number of Active Cycles in ADAMAC

The transition time for a h hop network where all the nodes have the same toggling frequency f_i in ADAMAC is given by:

$$t_r = \frac{h-1}{f_i}$$

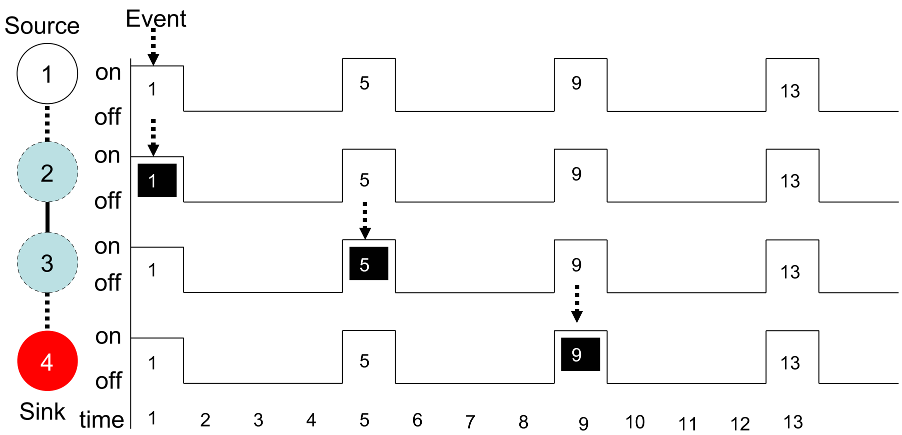


Figure C.1: Number of sensor wake-up cycles in a network with periodic duty cycle

This is demonstrated using Figure C.1 where all nodes have the same toggling frequency of $f_2 = \frac{1}{2^2}$. In a broadcast medium access control protocol, when an event is detected say at $t = 1$, it is immediately reported to neighbouring nodes in order to minimise the delay. In this case at time $t = 1$ when an event occurs at node 1, it is forwarded to node 2. Recall that owing to the broadcast storm problem, all intermediate relay nodes between the event source and the destination must wait until the next active cycle before any received messages can be forwarded. Therefore node 2 forwards the reported event occurrence onto node 3 at time $t = 5$ seconds and node 3 repeats the forwarding process at $t = 9$. Hence the total delay required to propagate the occurrence of the event after it is detected is $\left(\frac{h-1}{f_2} = \frac{2}{1/2^2}\right)$ 8 seconds.

When a new policy ϕ is introduced at time t_0 to replace a default toggling frequency f_j with a new toggling frequency f_i, the total number of sensor wake-up cycles, Ω, in time T is given by:

$$\Omega = \gamma + \left(\sum_{i=1}^{i=h-1} (h-i)\frac{f_i}{f_j} + i \right) + (h+1)\left\lfloor (T - \tau + t_0)\, f_i \right\rfloor$$

$\gamma = (h-1)\frac{f_i}{f_j} + 1$ and $\tau = \frac{(h-1)}{f_j}$.

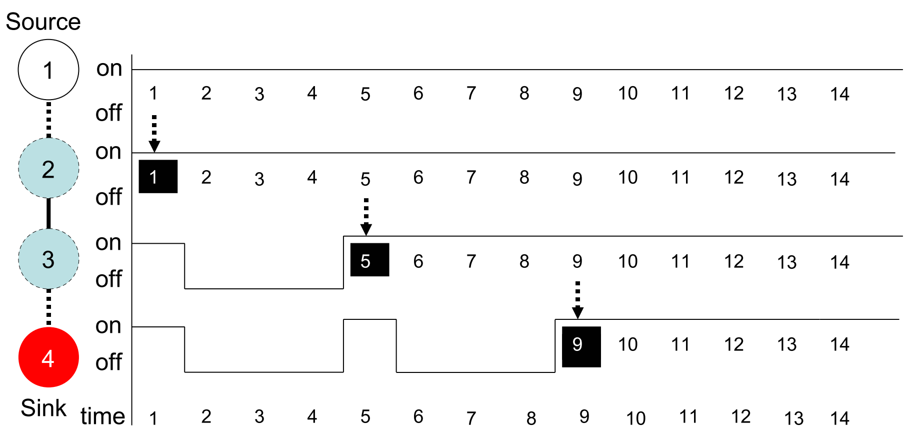


Figure C.2: Sensor wake-up cycles with a new duty cycle policy

To demonstrate this, consider again the 3 hop network in Figure C.2 where each node has a default toggling frequency of $f_2 = \frac{1}{2^2}$. When a new toggling frequency of $f_0 = 1/2^0$ is introduced at time $t = 1$, the network has a total of 47 active wake-up cycles in 14 seconds. Referring to Figure C.2, this new duty cycle disseminates across the network by time $t = 9$ seconds. During this time, there are a total of 25 sensor wake-up cycles throughout the network: ($\gamma = 2 \times 2^2/2^0 + 1$) 9 wake-up cycles for node 1, $(2 \times 2^2/2^0 + 1)$ 9 wake-up cycles for node 2, $(1 \times 2^2/2^0 + 2)$ 6 wake-up cycles for node 3 and $(0 \times 2^2/2^0 + 3)$ 3 wake-up cycle for node 4. Between 9 and 14 seconds, the network has a total of $\left(4 \times \lfloor (14 - 9) \times \frac{1}{2^0} \rfloor\right)$ 20 wake-up cycles. Putting this together gives a total of $(9 + 9 + 6 + 3 + 20)$ 47 wake-up cycles in 14 seconds as confirmed in Figure C.2

Bibliography

[ABDH08] Tal Anker, Danny Bickson, Danny Dolev, and Bracha Hod. Efficient clustering for improving network performance in wireless sensor networks. In *EWSN*, pages 221–236, 2008.

[ABL+94] B Alberts, D Bray, J Lewis, M Raff, K Roberts, and JD Watson. *Molecular Biology of the Cell*. Garland Publishing, 3rd edition, 1994.

[AKP08] Dylan G. Allegretti, Garrett T. Kenyon, and William C. Priedhorsky. Cellular automata for distributed sensor networks. *International Journal of High Performance Computing Applications*, 22(2):167–176, 2008.

[Alc06] Blake Alcott. *The Jevons Paradox and the myth of Resource Efficiency Improvements*, chapter 2, page 7. Earthscan, 2006.

[Ash62] WR Ashby. *Principle of Self-Organizing System, in Principles of Self-Organization*. Pegamon Press, 1962.

[ASYS02] I.F. Akyildiz, W. Su, and E. Cayirci Y. Sankarasubramaniam. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, 2002.

[BA52] By Doris Behrens-Abouseif. *Islamic Architecture in Cairo*, chapter Chapter 4, pages 50–51. Drill, 1952.

[Bag05] A. Baggio. Wireless sensor networks in precision agriculture. In *Workshop on Real World Wireless Sensor Networks*, 2005.

[BJ70] George E. P. Box and Gwilym M. Jenkins. *Time Series Analysis Forecasting and Control*. Holden-Day, fifth edition, 1970.

[BJ84] George E. P. Box and Gwilym M. Jenkins. *Introduction to Mathematical Statistics*. John Wiley & Sons, fifth edition, 1984.

[BK92] George Box and Tim Kramer. Statistical process monitoring and feedback adjustment-a discussion. *Technometrics*, 34(3):251–267, 1992.

[BM08] M.S. Britton and M.S. Maddison. Towards the reality of intelligent infrastructure with wireless meshed sensors. In *Railway Condition Monitoring, 2008 4th IET International Conference on*, pages 1–5, June 2008.

[Bre94] Hans J. Breermann. Self-organization in evolution, immune systems, economics, neural nets, and brains, in On Self-Organization. *Springer Series in Synergetics*, 61:5–34, 1994.

[BS06] Matthew Britton and Lionel Sacks. The secoas project-development of a self organising wireless sensor network for environmental monitoring. In *2nd International Workshop on Sensor and Adhoc Networks*, 2006.

[BSB07] Yann-Ael Le Borgne, Silvia Santini, and Gianluca Bontempi. Adaptive model selection for time series prediction in wireless sensor networks. *Journal of Signal Processing*, 87(12):3010–3020, 2007.

[BTB04a] R. Beckwith, D. Teibel, and P. Bowen. Unwired wine: Sensor networks in vineyards. In *Workshop on Real World Wireless Sensor Networks*, 2004.

[BTB04b] Richard Beckwith, Daniel Teibel, and Pat Bowen. Report from the field: Results from an agricultural wireless sensor network. In *29th Annual IEEE International Conference on Local Computer Networks*. IEEE Communications Society, 2004.

[BTC05] Seema Bandyopadhyay, Qingjiang Tian, and Edward J. Coyle. Spatio-temporal sampling rates and energy efficiency in wireless sensor networks. *IEEE/ACM Transactions on Networking*, 13(6):1339–1352, 2005.

[CAHS05] Qing Cao, Tarek Abdelzaher, Tian He, and John Stankovic. Towards optimal sleep scheduling in sensor networks for rare-event detection. In *IPSN '05: ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2005.

[Cal07] Alberto Calcagno. Dams and development: Relevant practices for improved decision-making. Technical Report 1, Division of Environmental Policy Implementation (DEPI), United Nations Environmental Programme, 2007.

[CC82] Gaylon S. Campbell and Melvin D. Campbell. Irrigation scheduling using soil moisture measurements: Theory and practice. *Advances in Irrigation*, 1:25–42, 1982.

[cc207] Cc2420 2.4 GHz ieee 802.15.4 / ZigBee RF Transceiver (rev b). http://focus.ti.com/docs/prod/folders/print/cc2420.html, March 2007.

[CES04] David Culler, Deborah Estrin, and Mani Srivastava. Guest Editor's Introduction: Overview of Sensor Networks. *IEEE Computer*, 37(8):41–49, 2004.

[CLBA+07] A. Cano, E. Lopez-Baeza, J. L. Anon, C. Reig, and C. Millan-Scheding. Wireless sensor network for soil moisture applications. In *SENSORCOMM '07: Proceedings of the 2007 International Conference on Sensor Technologies and Applications*, pages 508–513, Washington, DC, USA, 2007. IEEE Computer Society.

[CPD08] Simon Carlsen, Stig Petersen, and Paula Doyle. Using wireless sensor networks to enable increased oil recovery. In *ETFA 09: 13th*

IEEE International Conference on Emerging Technologies and Factory Automation, pages 1039–1048. IEEE, 2008.

[CPR03] Jim Chou, Dragan Petrovic, and Kannan Ramchandran. A distributed and adaptive signal processing approach to reducing energy consumption in sensor networks. In *INFOCOM '03: Proceedings of IEEE INFOCOM*, 2003.

[DAL+10] Bing Dong, Burton Andrews, Khee Poh Lam, Michael Hynck, Rui Zhang, Yun-Shang Chiou, and Diego Benitez. An information technology enabled sustainability test-bed (itest) for occupancy detection through an environmental sensing network. *Energy and Buildings*, In Press, Corrected Proof:–, 2010.

[DFB+07] Thanh Dang, Sergey Frolov, Nirupama Bulusu, Wu chi Feng, and Antonio Baptista. Near optimal sensor selection in the COolumbia RIvER(CORIE) observation network for data assimilation using genetic algorithms. In *Distributed Computing in Sensor Systems*, pages 253–266. Springer Berlin/Heidelberg, 2007.

[DGM+04] Amol Deshpande, Carlos Guestrin, Samuel R. Madden, Joseph M. Hellerstein, and Wei Hong. Model-driven data acquisition in sensor networks. In *VLDB '04: Proceedings of the Thirtieth international conference on Very large data bases*, pages 588–599. VLDB Endowment, 2004.

[DGM05] Amol Deshpande, Carlos Guestrin, and Samuel R. Madden. Resource-aware wireless sensor-actuator networks. *Bulletin of the IEEE Technical Committee on Data Engineering*, 28(1):40–47, 2005.

[DKR04] Antonio Deligiannakis, Yannis Kotidis, and Nick Roussopoulos. Compressing historical information in sensor networks. In *ACM*

SIGMOD '04: ACM Special Interest Group on Management of Data. ACM, 2004.

[Dre06] Falko Dressler. Self-organization in ad hoc networks: Overview and classification. Technical report, Department of Computer Science, University of Erlangen, 2006.

[Emb04] Ember 2420 2.4 GHz ieee 802.15.4 / zigbee RF transceiver. http://www.ember.com/pdf/EM2420datasheet.pdf, 2004.

[ES79] H Eigen and P Schuster. *The Hypercycle: A Principle of Natural Self-Organization*. Springer, 1979.

[ETAA04] F Emekci, S E Tuna, D Agrawal, and A E Abbadi. Binocular: A system monitoring framework. In *In Proceeedings of the 1st International Workshop on Data Management for Sensor Networks (DMSN)*, pages 5–9. ACM Press, New York, 2004.

[EY09] Chibuzor Edordu and Yang Yang. Dual prediction and probabilistic scheduling for efficient event detection. In *Wireless ViTAE '09: IEEE International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology*, 2009.

[FHAM95] PL Fuhr, DR Huston, TP Ambrose, and EF Mowat. An internet observatory:remote monitoring of instrumented civil structures using the information superhighway. *Smart Material and Structures*, 4:14–19, 1995.

[flo08] Wireless flood detection provides early warning for underserved countries. External Research Digital Inclusion Program, Microsoft Research, 2008.

[For65] John Formby. *An Introduction to the Mathematical Formulation of Self-Organizing Systems*. E. & F.N. Spon, 1965.

[FW02] Paul G Flikkema and Brent W West. Wireless sensor networks:from the laboratory to the field. In *In Proceedings of National Conference for Digital Government Research*, volume 129, pages 1–4, 2002.

[GEH03] Deepak Ganesan, Deborah Estrin, and John Heidemann. Dimensions: why do we need a new data handling architecture for sensor networks? *SIGCOMM Computer Communications Review*, 33(1):143–148, 2003.

[GLY07] Bŭgra Gedik, Ling Liu, and Philip S. Yu. Asap: An adaptive sampling approach to data collection in sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 18(12):1766–1783, 2007.

[GM04] Chao Gui and Prasant Mohapatra. Power conservation and quality of surveillance in target tracking sensor networks. In *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 129–143, New York, NY, USA, 2004. ACM.

[Goi08] Michael Gois. Flatmesh firmware design. Unpublished Report for Senceive Ltd, 2008.

[GS97] Igor Grabec and Wolfgang Sachse. *Synergetics of Measurement, Prediction and Control*. Springer, 1997.

[Hae03] M Haenggi. Energy-balancing strategies for wireless sensor networks. In *International Symposium on Circuits and Systems*, volume 4, page 25, 2003.

[HB06] J. Henao and C. Baanante. Agricultural Production and Soil Nutrients Mining in Africa: Implications for Resource Conservation and Policy

Development. Technical report, International Centre For Soil Fertility and Agricultural Development, 2006.

[HCB00] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *HICSS '00: Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8*, page 8020, Washington, DC, USA, 2000. IEEE Computer Society.

[Hei00] Wendy Beth Heinzelman. *Application-Specific Protocol Architecture for Wireless Networks*. PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 2000.

[HHM+09] Anton Hergenroder, Jens Horneber, Detlev Meier, Patrick Armbruster, and Martina Zitterbart. Distributed energy measurements in wireless sensor networks. In *SenSys '09: Proceedings of the 7th international conference on Embedded networked sensor systems*, pages 299–300. ACM, 2009.

[HHW97] Joseph M. Hellerstein, Peter J. Haas, and Helen J. Wang. Online aggregation. In *ACM Sigmod 97: International Conference on Management of Data*, pages 171–182, 1997.

[HMV04] Qi Han, Sharad Mehrotra, and Nalini Venkatasubramanian. Energy efficient data collection in distributed sensor environments. In *ICDCS '04: Proceedings of the International Conference on Distributed Computing Systems*, 2004.

[HMV07] Qi Han, Sharad Mehrotra, and Nalini Venkatasubramanian. Application-aware integration of data collection and power management in wireless sensor networks. *Journal of Parallel and Distributed Computing*, 67:992–1006, 2007.

[hon08] Honeywell sensor datasheet. http://www.honeywell.com/sensing, 2008.

[HP05] Peter Hebden and Adrian R. Pearce. Bloom filters for data aggregation and discovery: a hierarchical clustering approach. In *ISSNIP '05: Intelligent Sensors Sensor Networks and Information Processing*, pages 175–180, 2005.

[IEGH02] Chalermek Intanagonwiwat, Deborah Estrin, Ramesh Govindan, and John Heidemann. Impact of network density on data aggregation in wireless sensor networks. In *ICDCS '02: Proceedings of the International Conference on Distributed Computing Systems*, 2002.

[JC04] Ankur Jain and Edward Y. Chang. Adaptive sampling for sensor networks. In *DMSN '04: Proceedings of the 1st International workshop on Data management for sensor networks*, pages 10–16, New York, NY, USA, 2004. ACM.

[JCW04] Ankur Jain, Edward Y. Chang, and Yuan-Fang Wang. Adaptive stream resource management using kalman filters. In *ACM SIGMOD '04: ACM Special Interest Group on Management of Data*, 2004.

[Jeo09] Wootae Jeong. *Springer Handbook of Automation*, chapter 20, page 333. Springer Berlin Heidelberg, 2009.

[JRC08] Bo Jiang, Binoy Ravindran, and Hyeonjoong Cho. Energy efficient sleep scheduling in sensor networks for multiple target tracking. In *DCOSS '08: Proceedings of the 4th IEEE international conference on Distributed Computing in Sensor Systems*, pages 498–509, Berlin, Heidelberg, 2008. Springer-Verlag.

[JWT01] CA Janeway, M Walport, and P Travers. *Immunobiology: The Immune System in Health and Disease*. Garland Publishing, 5th edition, 2001.

[KE01] J Kennedy and RC Eberhart. *Swarm Intelligence*. Morgan Kaufmann, 2001.

[KFV11] Raghavendra V. Kulkarni, Anna Forster, and Ganesh Kumar Venayagamoorthy. Computational intelligence in wireless sensor networks: A survey. *To appear in IEEE Communications Surveys and Tutorials*, 13(1):1–29, 2011.

[Kin94] By Henry C. King. *The History of the Telescope*, chapter Chapter 3, page 34. Dover Publications, 1994.

[Lad07] Manish Lad. Challenges of resource constrained network embedded systems. Presentation at Wisig '05, 2007.

[Law08] Felicity Lawrence. Revealed: the massive scale of uk's water consumption. The Guardian, 2008.

[LCS05] Haiyang Liu, Abhishek Chandra, and Jaideep Srivastava. dsense: Data-driven stochastic energy management for wireless sensor platforms. Technical Report TR 05-018, Department of Computer Science, University of Minnesota, 2005.

[LCS06] Haiyang Liu, Abhishek Chandra, and Jaideep Srivastava. eSENSE: Energy Efficient Stochastic Sensing Framework for Wireless Sensor Platforms. In *IPSN '06: ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2006.

[Lim06] RIS International Limited. Canadian consumer battery baseline study, 2006.

[LK00] AM Law and WD Kelton. *Simulation, Modeling and Analysis*. McGraw-Hill, 3rd edition, 2000.

[LKR07] Gang Lu, Bhaskar Krishnamachari, and Cauligi S. Raghavendra. An adaptive energy-efficient and low-latency mac for tree-based data gathering in sensor networks: Research articles. *Wireless Communications and Mobile Computing*, 7(7):863–875, 2007.

[LM03] I. Lazaridis and S. Mehrotra. Capturing sensor-generated time series with quality guarantees. In *ICDC '03: IEEE Proceedings of the International Conference on Data Engineering*, 2003.

[LR02] S. Lindsey and C.S. Raghavendra. Pegasis: Power-efficient gathering in sensor information systems. *Aerospace Conference Proceedings, 2002. IEEE*, 3:1125–1130, 2002.

[LS03] Jacob R. Lorch and Alan Jay Smith. Operating system modifications for task-based speed and voltage scheduling. In *Mobisys '03: International Conference on Mobile Systems, Applications, and Services*, 2003.

[LSG04] Xiaotao Liu, Prashant Shenoy, and Weibo Gong. A time series-based approach for power management in mobile processors and disks. In *NOSSDAV '04: ACM Proceedings of the Network and Operating System Support for Digital Audio*, 2004.

[LWF03] Xiang-Yang Li, Peng-Jun Wan, and O. Frieder. Coverage in wireless ad hoc sensor networks. *Computers, IEEE Transactions on*, 52(6):753 – 763, june 2003.

[LWP07] Chong Liu, Kui Wu, and Jian Pei. An energy-efficient data collection framework for wireless sensor networks by exploting spatiotemporal correlation. *IEEE Transactions on Parallel and Distributed Systems*, 18(7):1010–1023, 2007.

[LWT05] Chong Liu, Kui Wu, and Min Tsao. Energy efficient information collection with the arima model in wireless sensor networks. *Global Telecommunications Conference, IEEE GLOBECOM'05*, 5:2470–2474, 2005.

[MC02] Rex Min and Anantha Ch. Top five myths about the energy consumption of wireless communication. *ACM Sigmobile Mobile Communication and Communications Review*, 7:65–67, 2002.

[McG04] J. McGlade. Self-organizing networks: Historical perspectives on the resilience of societal systems. Technical report, Economic and Social Research Council Report, 2004.

[MFH02] Samuel Madden, Michael J. Franklin, and Joseph M. Hellerstein. Tag: a tiny aggregation service for ad-hoc sensor networks. In *OSDI '02: Symposium on Operating Systems Design and Implementation*, 2002.

[Mid00] Gerard V. Middleton. *Data Analysis in The Earth Sciences Using Matlab*. Prentice Hall, Inc., New Jersey, 2000.

[MOH04] Kirk Martinez, Royan Ong, and Jane Hart. Glacsweb: A sensor network for hostile environments. In *EWSN '04: Proceedings of the First European Workshop on Wireless Sensor Networks*, pages 81–86, 2004.

[MPS+02] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson. Wireless sensor networks for habitat monitoring. In *WSNA '02: First ACM International Workshop on Wireless Sensor Networks and Applications*, Altlanta, USA, 2002. ACM Press.

[MS90] Renato E. Mirollo and Steven H. Strogatz. Synchronization of pulse-coupled biological oscillators. *SIAM Journal on Applied Mathematics*, 50(6):1645–1662, 1990.

[MSG05] Matthew J. Miller, Cigdem Sengul, and Indranil Gupta. Exploring the energy-latency trade-off for broadcasts in energy-saving sensor networks. In *ICDCS '05: Proceedings of the 25th IEEE International Conference on Distributed Computing Systems*, pages 17–26, Washington, DC, USA, 2005. IEEE Computer Society.

[NK08] Vinod Namboodiri and Abtin Keshavarzian. Alert: An adaptive low-latency event-driven mac protocol for wireless sensor networks. In *IPSN '08: Proceedings of the 7th international conference on Information processing in sensor networks*, pages 159–170, Washington, DC, USA, 2008. IEEE Computer Society.

[NTCS99] Sze-Yao Ni, Yu-Chee Tseng, Yuh-Shyan Chen, and Jang-Ping Sheu. The broadcast storm problem in a mobile ad hoc network. In *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 151–162, New York, NY, USA, 1999. ACM.

[OLW01] Chris Olston, Boon Thau Loo, and Jennifer Widom. Adaptive precision setting for cached approximate values. In *ACM SIGMOD '01: ACM Special Interest Group on Management of Data*, 2001.

[Ora91] Benedict Okechukwu Oramah. *An Evaluation of Economic Impact of Irrigation Projects in the Lower Anambra River Basin on The Farming Community*. PhD thesis, Department of Agricultural Economics, Obafemi Owolowo University,Ile-Ife, Nigeria, 1991.

[OW02] Chris Olston and Jennifer Widom. Best-effort cache synchronization with source cooperation. In *In SIGMOD*, pages 73–84, 2002.

[PAC05] R. Poornachandran, H. Ahmad, and H. Cam. Energy-efficient task scheduling for wireless sensor nodes with multiple sensing units. In *IPCCC '05: International Conference on Performance, Computing, and Communications*, 2005.

[PB05] Christian Prehofer and Christian Bettstetter. Self-organization in communication networks: Principles and design paradigms. *IEEE Communications Magazine*, 43(7):78–85, 2005.

[PHP+97] D. Pimentel, J. Houser, E. Preiss, O. White, H. Fang, L. Mesnick, T. Barsky, S. Tariche, J. Schreck, and S. Alpert. Water resources: Agriculture, the environment, and society. *Bioscience*, 47(2):97–106, 1997.

[Pit08] Michael Pitt. The pitt review: Learning lesson from the 2007 floods. Technical report, Cabinet Office, Whitehall, 2008.

[PK00] Gregory J. Pottie and William J. Kaiser. Embedding the internet: Wireless intergrated network sensors. *Communications of the ACM*, 43(5):51–58, 2000.

[PRP+06] Jacques Panchard, Seshagiri Rao, T.V. Prabhakar, H.S. Jamadagni, and Jean-Pierre Hubaux. COMMON-Sense Net: Improved Water Management for Resource-Poor Farmers via Sensor Networks. In *International Conference on Communication and Information Technologies and Development (ICTD2006)*, 2006.

[RAdS+00] J.M. Rabaey, M.J. Ammer, J.L. da Silva, D. Patel, and S. Roundy. Picoradio supports ad hoc ultra low power wireless networking. *IEEE Computer*, 33(7):42–48, 2000.

[Rap02] T. S. Rappaport. *Wireless communications principles and practices*. Prentice-Hall, 2002.

[Ric94] Michael Richter. Self-organization, articial intelligence and connectionism, in On Self-Organization. *Springer Series in Synergetics*, 61:80–91, 1994.

[Rou03] S. Roundy. *Energy Scavenging in Wireless Sensor Networks*. Kluwer Academic Publishers, 2003.

[RSF+04] S. Roundy, D. Steingart, L. Frechette, P. Wright, and J. Rabaey. Power sources for wireless sensor networks. In *EWSN '04: Proceedings of* *the First European Workshop on Wireless Sensor Networks*, pages 1–17. LNCS, Springer, 2004.

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

[RV06] Ramesh Rajagopalan and Pramod K. Varshney. Data aggregation techniques in sensor networks: A survey. *IEEE Communications Surveys and Tutorials*, 8(4):48–63, 2006.

[SBF+07] Adam Silberstein, Rebecca Braynard, Gregory Filpus, Gavino Puggioni, Alan Gelfand, Kamesh Munagala, and Jun Yang. Data-driven processing in sensor networks. In *CIDR '07: Third Biennal Conference on Innovative Data Systems Research*, 2007.

[SBLC03] Mohamed A. Sharaf, Jonathan Beaver, Alexandros Labrinidis, and Panos K. Chrysanthis. Tina: A scheme for temporal coherency-aware in network aggregation. In *MobiDE'03: 3rd ACM International Workshop on Data Engineering for Wireless and Mobile Access*, pages 69–76, 2003.

[SC01] Amit Sinha and Anantha Chandrakasan. Dynamic power management in wireless sensor networks. *IEEE Des. Test*, 18(2):62–74, 2001.

[SCB96] Mani Srivastava, Anantha P. Chandrakasan, and Robert W. Brodersen. Predictive system shutdown and other architectural techniques for energy efficient programmable computation. *IEEE Transactions on Very Large Scale Integration Systems*, 4(1):42–55, 1996.

[SCV+06] Pavan Sikka, Peter Corke, Philip Valencia, Christopher Crossman, Dave Swain, and Greg Bishop-Hurley. Wireless adhoc sensor and actuator networks on the farm. In *IPSN '06: ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2006.

[sen07] SensorScope Project. http://sensorscope.epfl.ch/index.php/Downloads, October 2007.

[SHX+09] Wen-Zhan Song, Renjie Huang, Mingsen Xu, Andy Ma, Behrooz Shirazi, and Richard LaHusen. Air-dropped sensor network for real-time high-fidelity volcano monitoring. In *MobiSys '09: Proceedings of the 7th international conference on Mobile systems, applications, and services*, pages 305–318, New York, NY, USA, 2009. ACM.

[SNMT07] Ivan Stoianov, Lama Nachman, Sam Madden, and Timur Tokmouline. Pipenet:a wireless sensor network for pipeline monitoring. In *IPSN '07: ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2007.

[SR06] Silvia Santini and Kay Romer. An adaptive strategy for quality based data reduction in wireless sensor networks. In *Proceedings of the 3rd International Conference on Networked Sensing Systems, INSS '06*, Chicago, IL, USA, Jun 2006. TRF.

[SYTCS99] Sze-Yao, Yu-Chee Tseng, Yuh-Shyan Chen, and Jang-Ping Sheu. The broadcast storm problem in a mobile ad hoc network. In *MobiCom '99: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 151–162, New York, NY, USA, 1999. ACM Press.

[TB08] National Transpotation and Safety Board. Pipeline accident report: Natural gas distribution line break and subsequent explosion and fire plum borough, pennsylvania march 5, 2008. Report, 2008.

[TB09] Moses Makooma Tenywa and Mateete Bekunda. Managing soils in sub-saharan africa: Challenges and opportunities for soil and water conservation. *Journal of Soil and Water Conservation*, 64(1):44–48, January 2009.

[TCP09] Alex Talevski, Simon Carlsen, and Stig Petersen. Research challenges in applying intelligent wireless sensors in the oil, gas and resources industries. In *INDIN 09: 7th IEEE International Conference on Industrial Informatics*, pages 464–469. IEEE, 2009.

[TGL05] J. Thelen, D. Goense, and K. Langendoen. Radio wave propagation in potato fields, 2005.

[TM06] Daniela Tulone and Samuel Madden. Paq: Time series forecasting for approximate query answering in sensor networks. In Kay Rmer, Holger Karl, and Friedemann Mattern, editors, *EWSN '06: Proceedings of the Third European Conference on Wireless Sensor Networks*, volume 3868 of *Lecture Notes in Computer Science*, pages 21–37. Springer, 2006.

[TUML07] Athanasia Tsertou, Rochan Upadhyay, Stephen McLaughlin, and David I Laurenson. Towards a tailored sensor network for fire emergency monitoring in large buildings. In *Proceedings of the 1st IEEE International Conference in Wireless Rural and Emergency Communications (WRECOM07)*. IEEE Communications Society, 2007.

[Tv02] AS Tanenbaum and M van Steen. *Distributed Systems: Principles and Paradigms*. Prentice-Hall, 2002.

[VA06] Mehmet C. Vuran and Ozgur B. Akan. Spatio-temporal characteristics of point and field sources in wireless sensor networks. In *ICC '06: International Conference on Communications*, 2006.

[VAA04] Mehmet C. Vuran, O. B. Akan, and Ian F. Akyildiz. Spatio-temporal correlation: theory and applications for wireless sensor networks. *Comput. Netw.*, 45(3):245–259, 2004.

[vDL03] Tijs van Dam and Koen Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *SenSys '03: Proceedings of* *the 1st international conference on Embedded networked sensor systems*, pages 171–180, New York, NY, USA, 2003. ACM.

[Wai07] Andrew Wain. Personal communication regarding the use of data loggers in hydrological surveys, 2007.

[Wan03] S. Y. Wang. Reducing the energy consumption caused by flooding messages in mobile ad hoc networks. *Computer Networks*, 42(1):101 – 118, 2003. Contains papers of the Theme Issue 'Small and Home Networks'.

[WD08] Lidan Wang and Amol Deshpande. Predictive modeling-based data collection in wireless sensor networks. In *EWSN*, pages 34–51, 2008.

[Wei72] Charles Weiss. Satellites and international resource development. *Finance and Development*, 9(2):9–15, 1972.

[WH06] Quanhong Wang and Hossam Hassanein. *Sensor Network Protocols, A Comparative Study of Energy-Efficient Protocols for Wireless Sensor Networks*, chapter 5, pages 5–1. CRC Press, 2006.

[WZW06] Ning Wang, Naigian Zhang, and Machua Wang. Wireless sensors in agriculture and food industry-recent development and future perspective. *Computers and Electronics in Agriculture*, 50(1):1–14, January 2006.

[YG03] Y. Yao and J. Gehrke. Query processing in sensor networks. In *In Proceedings of IEEE Pervasive Computing 2003*, 2003.

[YHE02] W Ye, J Heidemann, and D Estrin. An energy efficient mac protocol for wireless sensor networks. In *INFOCOM '02: Proceedings of IEEE INFOCOM*, 2002.

[YS05] SunHee Yoon and Cyrus Shahabi. Exploiting spatial correlation towards an energy efficient clustered aggregation technique(cag). In *ICC '05:*

Proceedings of the International Conference on Communications, pages 3307–3313, 2005.

[YWZ06] Yang Yang, Huihai Wu, and Weihua Zhuang. Mester: minimum energy spanning tree for efficient routing in wireless sensor networks. In *QShine '06: Proceedings of the 3rd international conference on Quality of service in heterogeneous wired/wireless networks*, page 17, New York, NY, USA, 2006. ACM.

[ZG04] Feng Zhao and Leonidas J Guibas. *Wireless Sensor Networks: An Information Processing Approach*. Morgan Kaufmann, 2004.

[Zha03] Feng Zhao. Collaborative signal and information processing: an information directed approach. *Proceeding of IEEE*, 91(8):1199–1209, 2003.

[Zim80] H. Zimmermann. Osi reference model-the iso model of architecture for open systems interconnection. *IEEE Transactions on Communications*, 28(4):42–48, 1980.

[ZLN07] Y Zhu, Y Liu, and LM Ni. Low-power distributed event detection in wireless sensor networks. In *INFOCOM '07: Proceedings of IEEE INFOCOM*, 2007.