UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

ENERGY-EFFICIENT MAC PROTOCOL FOR WIRELESS SENSOR NETWORKS

by

Christoph Erik Tönsing

Submitted in partial fulfilment of the requirements of the degree

Master of Engineering (Computer Engineering)

in the

Faculty of Engineering, the Built Environment and Information Technology

UNIVERSITY OF PRETORIA

February 2008

ACKNOWLEDGEMENTS

Thank you to my family and friends who gave encouragement and support throughout my studies whenever necessary. Especially to those whose prayers have been answered by the conclusion of this research.

A special word of thanks is also due to my supervisor, Prof. Gerhard Hancke from the Department of Electrical, Electronic and Computer Engineering at the University of Pretoria. His insight and support were essential in developing this research.

Most importantly, all honour and glory be to the Lord God Almighty, Jesus Christ and the Holy Spirit, in whom and through whom all things find their beginning and end.

SUMMARY OF DISSERTATION

ENERGY-EFFICIENT MAC PROTOCOL FOR WIRELESS SENSOR NETWORKS

by

Christoph Erik Tönsing

Supervisor: Prof. G. P. Hancke
Department: Electrical, Electronic and Computer Engineering
University of Pretoria
Degree: Master of Engineering (Computer Engineering)

A Wireless Sensor Network (WSN) is a collection of tiny devices called sensor nodes which are deployed in an area to be monitored. Each node has one or more sensors with which they can measure the characteristics of their surroundings. In a typical WSN, the data gathered by each node is sent wirelessly through the network from one node to the next towards a central base station.

Each node typically has a very limited energy supply. Therefore, in order for WSNs to have acceptable lifetimes, energy efficiency is a design goal that is of utmost importance and must be kept in mind at all levels of a WSN system. The main consumer of energy on a node is the wireless transceiver and therefore, the communications that occur between nodes should be carefully controlled so as not to waste energy. The Medium Access Control (MAC) protocol is directly in charge of managing the transceiver of a node. It determines when the transceiver is on/off and synchronizes the data exchanges among neighbouring nodes so as to prevent collisions etc., enabling useful communications to occur. The MAC protocol thus has a big impact on the overall energy efficiency of a node.

Many WSN MAC protocols have been proposed in the literature but it was found that most were not optimized for the group of WSNs displaying very low volumes of traffic in the network. In low traffic WSNs, a major problem faced in the communications process is clock drift, which causes nodes to become unsynchronized. The MAC protocol must overcome this and other problems while expending as little energy as possible. Many

useful WSN applications show low traffic characteristics and thus a new MAC protocol was developed which is aimed at this category of WSNs.

The new protocol, Dynamic Preamble Sampling MAC (DPS-MAC) builds on the family of preamble sampling protocols which were found to be most suitable for low traffic WSNs. In contrast to the most energy efficient existing preamble sampling protocols, DPS-MAC does not cater for the worst case clock drift that can occur between two nodes. Rather, it dynamically learns the actual clock drift experienced between any two nodes and then adjusts its operation accordingly.

By simulation it was shown that DPS-MAC requires less protocol overhead during the communication process and thus performs more energy efficiently than its predecessors under various network operating conditions. Furthermore, DPS-MAC is less prone to become overloaded or unstable in conditions of high traffic load and high contention levels respectively. These improvements cause the use of DPS-MAC to lead to longer node and network lifetimes, thus making low traffic WSNs more feasible.

Keywords: wireless sensor networks, medium access control, energy efficient, preamble sampling, clock drift, synchronization.

OPSOMMING VAN VERHANDELING

ENERGIE-EFFEKTIEWE MEDIUM-TOEGANG-BEHEER-PROTOKOL VIR DRAADLOSE SENSORNETWERKE

deur

Christoph Erik Tönsing

Toesighouer: Prof. G. P. Hancke

Departement: Elektriese, Elektroniese en Rekenaar-Ingenieurswese
Universiteit van Pretoria

Graad : Meester van Ingenieurswese (Rekenaar-Ingenieurswese)

’n Draadlose Sensornetwerk (DSN) is ’n versameling van klein toestelle, genoem sensor-nodes, wat versprei word in ’n gebied wat gemonitor moet word. Elke node het een of meer sensors waarmee hierdie nodes die eienskappe van die omgewing meet. In ’n tipiese DSN word die inligting wat elke node versamel draadloos deur die netwerk gestuur, van een node na die volgende tot by ’n sentrale basis-stasie.

Elke node het tipies ’n baie beperkte energiebron. Om te verseker dat DSNe ’n aanvaarbare leeftyd het is energie-doeltreffendheid dus ’n baie belangrike ontwerpdoelwit wat by alle vlakke van ’n DSN-ontwerp in aanmerking geneem moet word. Die grootste verbruiker van energie in ’n node is die sender/ontvanger en dus moet die kommunikasie wat tussen nodes plaasvind goed beheer word om energie te bewaar. Die Medium-Toegang-Beheer-(MTB) protokol is direk verantwoordelik vir die beheer van die sender/ontvanger. Dit bepaal wanneer die sender/ontvanger afgeskakel is en sinchroniseer die data-uitruilings tussen nodes sodat botsings, ens. vermy word en sinvolle kommunikasie kan plaasvind. Die MTB-protokol het dus ’n groot invloed op die totale energie-doeltreffendheid van ’n node.

Baie DSN-MTB-protokolle is reeds in die literatuur voorgestel, maar die meeste daarvan is nie vir DSNe met lae netwerk-verkeer geoptimeer nie. In lae-verkeer DSNe is klokdryf ’n groot probleem in die kommunikasie-proses omdat dit kan veroorsaak dat nodes sinchronisme verloor. Die MTB-protokol moet hierdie en ander probleme oorkom en

terselfdertyd so min energie as moontlik verbruik. Baie van die nuttige DSN-toepassings ervaar lae verkeer en daarom is ‘n nuwe MTB-protokol spesifiek vir so geval ontwikkel.

Die nuwe protokol, genoem ‘Dynamic Preamble Sampling Medium Access Control’ (‘DPS-MAC’), bou voort op die groep van “preamble sampling” protokolle wat tot dusver mees geskik vir lae-verkeer DSNe was. Die mees energie-effektiewe “preamble sampling“ protokolle tot dusver maak voorsiening vir die maksimum tydsein-afwyking tussen twee nodes (as gevolg van klokdryf). In teenstelling hiermee leer DPS-MAC dinamies die werklike tydsein-afwyking tussen enige twee nodes en pas dan sy werking daarby aan.

Simulasies het getoon dat die DPS-MAC protokol minder boodskapuitruilings benodig tydens die kommunikasie-proses vir verskillende toestande. DPS-MAC is dus meer energie-effektief as sy voorgangers. Verder is daar gevind dat DPS-MAC minder maklik oorlaai of onstabiel word in situasies van hoë verkeer-volumes of hoë node-digtheid. Hierdie verbeteringe lei tot langer node- en netwerk-leeftye en bied dus ‘n meer effektiewe oplossing vir lae-verkeer DSNe.

Sleutelwoorde: draadlose sensornetwerke, medium-toegang-beheer, energie-effektief, tydsein-afwyking, sinchronisasie.

LIST OF ABBREVIATIONS

ACK – Acknowledgement packet

AC-MAC – Adaptive Coordinated MAC

ADC – Analog to Digital Converter

AI-LMAC – Adaptive Information-centric and Lightweight MAC

AIMD – Additive Increase Multiplicative Decrease

BER – Bit Error Rate

BMA MAC – Bit Map Assisted MAC

B-MAC – Berkeley MAC

BTMA – Busy Tone Multiple Access

CAP – Contention Access Period

CCA – Clear Channel Assessment

CC-MAC – Correlation-based Collaborative MAC

CDMA – Code Division Multiple Access

CR – Communication Request

CRC – Cyclic Redundancy Check

CS – Carrier Sense

CSMA – Carrier Sense Multiple Access

CSMA/CA – CSMA with Collision Avoidance

CSMAC – CDMA Sensor MAC

CSMA-MPS – CSMA with Minimum Preamble Sampling

CSMA-PS – CSMA with Preamble Sampling

CTS – Clear To Send

DCF – Distributed Coordinated Function

DE-MAC – Distributed Energy-aware MAC

DLL – Data Link Layer

DPS-MAC – Dynamic Preamble Sampling MAC

DSMAC – Dynamic Sensor MAC

EBB – Exponential Binary Back-off

EKF – Extended Kalman Filter

E-MAC – Event MAC

ER-MAC – Energy and Rate-based MAC

FDMA – Frequency Division Multiple Access

FIFO – First In First Out

FLAMA – Flow-Aware Medium Access

FSK – Frequency Shift Keying

GTS – Guaranteed Time Slot

IEEE – Institute of Electrical and Electronic Engineers

INS – Iterative Node Selection

ISM – Industrial, Scientific and Medical

LAN – Local Area Network

LEACH – Low-Energy Adaptive Clustering Hierarchy

LMAC – Lightweight MAC

LOS – Line Of Sight

LR-WPAN – Low-Rate Wireless Personal Area Network

MAC – Medium Access Control

MACA – Multiple Access with Collision Avoidance

MANET – Mobile Ad-hoc Network

MF – Mobility Framework

MIT – Massachusetts Institute of Technology

MMAC – Mobility-adaptive collision free MAC

MS-MAC – Mobility-aware Sensor MAC

N-ACK – Negative Acknowledgement packet

NAV – Network Allocation Vector

N-MAC – Network MAC

OOK – On-Off Keying

OSI – Open Systems Interconnect

PAMAS – Power-Aware Multiple Access with Signalling

PDA – Personal Digital Assistant

PHY – Physical layer

PLE – Path Loss Exponent

PMAW – Power and Mobility-Aware Wireless protocol

ppm – parts per million

PRNG – Pseudo Random Number Generator

QoS – Quality of Service

RAM – Random Access Memory

RF – Radio Frequency

ROM – Read-Only Memory

RSSI – Received Signal Strength Indicator

RTS – Request To Send

RX - Receive

S-MAC – Sensor MAC

SMACS – Self-Organizing MAC for Sensor networks

SNR – Signal to Noise Ratio

SOP – Start Of Packet

STEM – Sparse Topology and Energy Management

SYNC – Synchronization packet

TC – Traffic Control

TDMA – Time Division Multiple Access

TDMA-W – TDMA Wakeup

TEEM – Traffic aware, Energy Efficient MAC

T-MAC – Timeout MAC

TRACE – Time Reservation using Adaptive Control for Energy efficiency

TRAMA – Traffic-Adaptive Medium Access

TX – Transmit

UCLA – University of California, Los Angeles

WLAN – Wireless Local Area Network

WPAN – Wireless Personal Area Network

WSN – Wireless Sensor Network

WU – Wake-Up

TABLE OF CONTENTS

PAGE

Chapter 1
RESEARCH OVERVIEW

1.1 Introduction

The Internet revolution in the nineties started an ever-growing trend to search for new and better ways to connect everything, anytime, anywhere [1]. The success of the mobile phone industry, for example, rests on this trend which can be called the quest for ubiquitous computing, or ambient intelligence, where one's entire surrounding is connected and "digitized". Wireless Sensor Network (WSN) technology forms part of this trend and has enjoyed growing interest in the recent past.

A WSN is an autonomous collection of many small devices called sensor nodes which are distributed over an area. Each node is capable of performing some type of environment sensing (e.g. temperature, humidity, vibration sensing etc.). Furthermore, nodes are able to process the sensed data and then wirelessly send the data towards a base station node which collects the data from all the nodes in the network. WSN applications are diverse in nature, ranging from home automation and environmental monitoring to battlefield surveillance and many more. In a number of WSN applications, sensor nodes are not required to communicate with each other frequently, since the tasks they perform only require infrequent sensor readings to be delivered to the base station by each node. Such WSNs will naturally display low volumes of traffic, with a node taking a sensor reading once every few minutes, hours or even less frequently. A typical example would be an environmental monitoring WSN which maps the annual temperature distribution of a certain geographical region. On the other hand, some WSNs could require much more frequent communications among sensor nodes. A WSN monitoring a piece of factory

machinery for example could be required to have real-time characteristics with multiple sensor readings taken and forwarded to the base station per minute.

It is envisioned that sensor nodes could be deployed in large numbers in the area where they will operate. For this reason, and because of the possibly hostile, dangerous or simply hard-to-reach terrain in which a WSN will be active, such a network should be dynamically deployable and furthermore, sensor nodes must be able to function for long periods of time without any outside assistance. The main issue affecting this goal is energy efficiency. If a node can survive for a long period of time on a very limited energy supply, then it can satisfy its purpose. If not, then the WSN concept will not be able to deliver on its promises. Other challenges for sensor nodes such as small size, low cost etc. may or may not be crucial depending on the specific application. However, without energy efficiency, most of the other challenges can never be reached. A node cannot be made small if it needs a large battery or other source of power such as a solar panel. Also, a WSN cannot be made low cost if the maintenance of the network requires frequent battery replacements.

The challenge of making WSNs energy efficient is one that needs to be dealt with at every level of the network and every step of the design process. Raghunathan et al. [2] gives an excellent overview of energy efficiency in WSNs. An important observation made by the authors is that by far the main consumer of energy on a sensor node, with current state-of-the-art technology, is the radio. This means that every aspect of the communication among nodes must be particularly carefully designed to reduce power wastage by the radio. When one considers energy consumption of a wireless radio system, the Medium Access Control (MAC) protocol is one of the prevailing factors. Of course the modulation scheme, the actual hardware and other physical layer (PHY) issues are also crucial in an energy-efficient design but once this layer has been optimized, the radio hardware will still form a significant portion of the node's total energy consumption and thus the active time of the radio needs to be carefully controlled. This is the task performed by the MAC protocol.

Wireless Sensor Network overviews such as [3], as well as many current calls for papers of upcoming conferences indicate that WSN protocols, not least of all the MAC protocols, are far from a settled issue. The WSN field is still young and the amount of work already done only points to the vast interest in this possibly revolutionary application area.

1.2 Scope

The research in this dissertation deals with the design of a MAC protocol that demonstrates improved energy efficiency compared to existing MAC protocol solutions for WSNs. The new MAC protocol is designed to be simple and generic enough to be eligible for implementation on various hardware bases. Furthermore, the MAC protocol is scalable to different network sizes and is optimized for low traffic WSNs where network lifetime, and thus also sensor node lifetime, is required to be on the order of a number of years.

1.3 Problem Statement

In WSNs, where sensor nodes have limited hardware complexity and a finite supply of energy, MAC protocols encounter certain problems that are more pronounced than in traditional wireless networks such as Wireless Local Area Networks (WLANs) etc. Furthermore, the way in which these problems are dealt with becomes highly restricted, due to the requirement to treat energy resources as an extremely rare commodity. For example, one of the main problems is clock drift, which leads to synchronization loss between nodes. The challenge is for MAC protocols to navigate such problems while causing as little overhead energy consumption as possible.

The specific area of low traffic WSNs has been addressed by a relatively small percentage of research, so that ultra low-power WSN MAC protocols and their ability to deliver long network lifetimes have not been exhaustively dealt with in the literature. Most MAC protocols described in the literature deal with problems such as clock drift at a superficial level, if at all. Thus, there is a need for further research and development in the area of low traffic WSN MAC protocols.

1.4 Objectives

The main objective of this research is to contribute to the existing knowledge base of WSNs by developing a new energy-efficient MAC protocol.

The protocol should exhibit the following characteristics.

- The protocol should display improved energy efficiency compared to existing protocols.
- The protocol should be simple and generic enough to be suitable for use on different hardware platforms.
- The protocol should be scalable to differing network sizes and node densities.

1.5 Research Approach

In order to achieve the objectives mentioned above, it was necessary to take the following steps in the research process.

- To investigate and gain a thorough understanding of WSNs, their application areas, their characteristics and challenges.
- To perform an extensive literature survey of existing WSN MAC protocols so as to
    - attain a thorough understanding of the factors that affect WSN MAC protocol operation,
    - determine the requirements and challenges that WSN MAC protocols face,
    - form an overview of the current *state-of-the-art*, i.e. understand the main features of current WSN MAC protocols, and
    - determine the strengths and weaknesses of current WSN MAC protocols as well as their applicability to different application areas.
- To design a new WSN MAC protocol, building on the best features of current protocols, so as to yield improved energy efficiency.
- To investigate methods of verifying the proposed design.
- To perform verification of the proposed protocol in comparison to existing WSN MAC protocols so as to demonstrate the planned improvement in energy efficiency or give reasons for such improvements not being possible.

1.6 Dissertation Overview

The rest of this dissertation is organized in the following manner.

Chapter 2 investigates the basics of WSNs. A description of WSNs, their application areas, challenges and characteristics is given. Chapter 3 will focus more specifically on WSN MAC protocols. It will identify and discuss requirements and challenges for MAC protocols. Furthermore, the large body of existing literature on WSN MAC protocols will be dealt with in greater detail. On the basis of the existing literature, a new WSN MAC protocol design will then be proposed in Chapter 4. An appropriate verification method for the new design will be discussed in Chapter 5. Results of the verification of the new design in comparison to existing protocols will be presented in Chapter 6. A discussion of these results will also be given. Finally, the research documented in this dissertation will be concluded in Chapter 7.

Chapter 2
WIRELESS SENSOR NETWORKS

The broad and growing field of Wireless Sensor Networks (WSNs) forms the foundation of the research contained in this dissertation. The concept of WSNs will be investigated in some detail in this chapter. Their characteristics, requirements, challenges and application areas will be discussed so as to form an understanding of the research context for the remainder of the dissertation.

2.1 The WSN Concept

With continued advances in the fields of digital electronics and wireless communications, WSNs have become a widely researched concept within the past decade. A major reason for this rapid growth of interest is the vast range of interesting applications which may benefit from WSNs. They include military applications (battlefield surveillance, targeting etc.), environmental monitoring (wildlife tracking, rainfall mapping, forest fire detection, pollution tracking, biodiversity mapping), disaster reaction (structural stability monitoring after earthquakes, survivor detection), agricultural applications (precision farming), commercial and home applications (machinery maintenance monitoring, asset tracking, home automation) [4], and many others, of which some real-world examples will be given later in this chapter. Such applications have become tantalizing possibilities due to the fact that it has become technologically feasible to construct small, inexpensive and low-power sensor nodes which can communicate with each other wirelessly.

In order to give a very general or stereotype definition, it can be stated that a WSN is a large collection of tiny sensing devices which communicate with each other over short-range wireless links. The sensor nodes collect data about their environment via various types of sensors (e.g. acoustic, light, temperature, moisture, pressure sensors etc.). This data is forwarded in a hop-by-hop fashion from one node to another until it reaches a central base station where it can be analyzed or processed in whatever manner is needed. A graphical illustration of this most basic architecture is given in Figure 2.1 a. It is seen that the base station could be connected via a wired link to other networks, e.g. the Internet, to make it possible to remotely access the collected data.

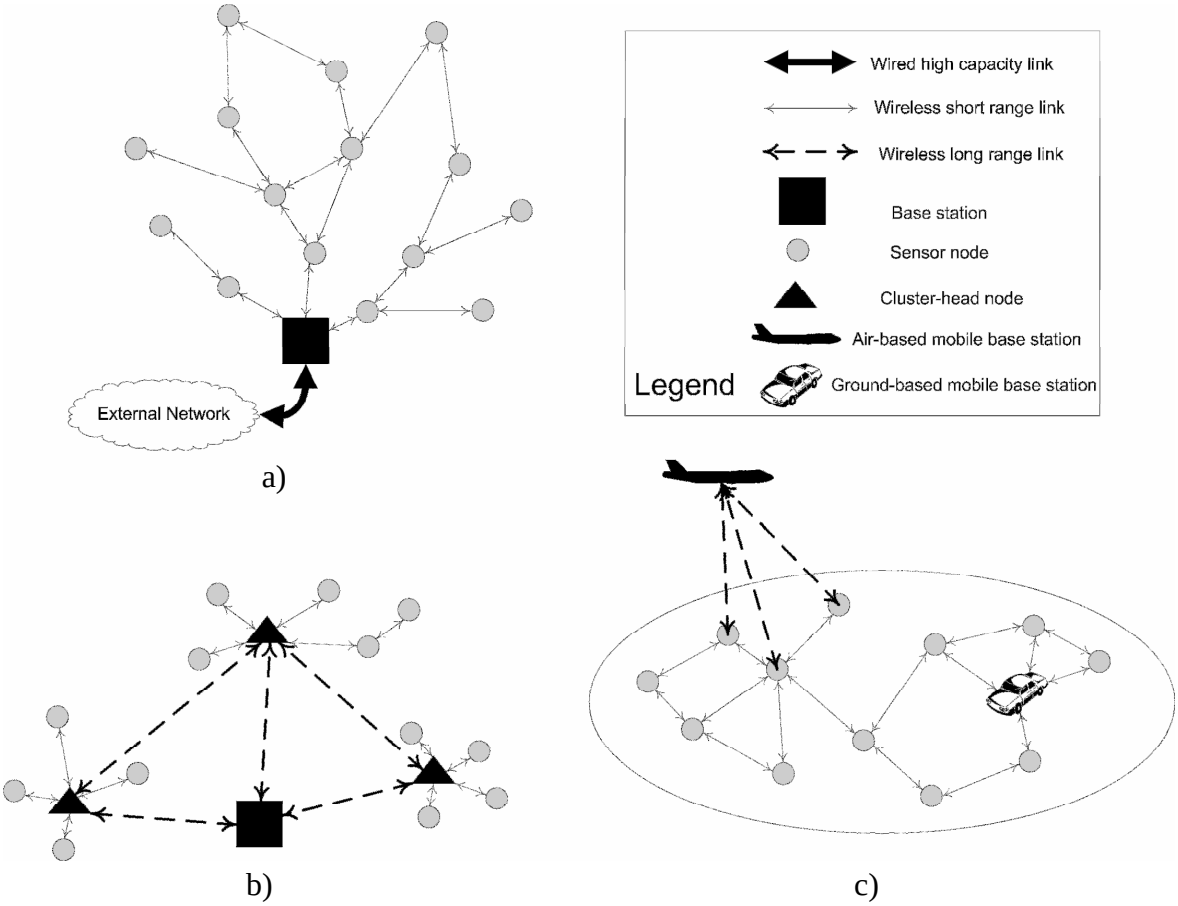

Figure 2.1. Various WSN architectures. a) Flat architecture, b) hierarchical architecture and c) architecture with mobile base stations.

Apart from this, other architectures are also possible, of which two are pictured in Figure 2.1 b and c. Figure 2.1 b shows a hierarchical WSN structure, where certain nodes are (possibly) more sophisticated and act as cluster heads or local gathering/management

stations. Figure 2.1 c on the other hand shows an architecture where mobile base stations (ground or air-based) move through or over a sensor field to collect sensor node data.

2.2 WSN Design Factors

Despite the various types of WSNs that may exist, there are a number of common factors and characteristics that should be kept in mind in any WSN design. They are mentioned below (a number of these were adapted from [5] and [6]).

2.2.1 Network size and scalability

For the purpose of large scale data gathering, network sizes could be on the order of thousands of nodes or more. It cannot be argued that all WSNs will have such proportions, but indeed, it could be a possibility. This means that WSNs should demonstrate scalability at all system levels, including the physical hardware as well as any protocols or algorithms that operate in the network.

2.2.2 Operating environment and deployment

The environments in which WSNs can be applied are as diverse as the applications themselves. However, a common feature is that such environments could well be dangerous, hard to reach or remote geographic locations that are isolated from human presence. This means that WSNs should function without any human assistance after deployment. Furthermore, the deployment can take place in several ways. It can be random (e.g. nodes dropped from aeroplane) or structured (nodes placed in strategic positions “by hand”). It is desirable in either case, that once deployed, the network configures itself automatically.

2.2.3 Network lifetime

Most applications for WSNs require that the network remain active for long periods of time (on the order of several months to several years). This is due to the possibly remote location of the WSN and on the other hand the shear high cost that regular maintenance of a large distributed network would incur. Since the failure of a small number of nodes could cause network partitioning and thus failure of the entire network, individual sensor nodes can be required to have the same lifespan as the network itself.

2.2.4 Expected data traffic

The expected data traffic in a WSN can vary widely depending on application. The variation occurs both in terms of traffic volume and traffic patterns. In the most general case however, WSNs display low volumes of traffic, where the term “low volume” is used in relation to other network types such as wired Local Area Networks (LANs). Furthermore, unlike many other network types which display peer-to-peer traffic patterns, WSNs often exhibit a many-to-one, source-to-sink, data gathering tree traffic pattern.

2.2.5 Sensor node size, cost and resources

The required size of individual sensor nodes may again vary for different applications; however, in general, a small form factor is desirable, especially in certain applications such as military surveillance or environment monitoring. In [5], node sizes are grouped as *shoebox*, *matchbox*, *grain* and *dust*. The latter two sizes remain elusive so far, but technological advances continue to pursue this goal of decreasing sizes.

Regarding sensor node cost, there are two main reasons for aiming at an eventual very low cost per node. Firstly, WSNs can consist of large numbers of nodes. Secondly, such nodes can be treated as throw-away devices. In other words, they are likely not to be gathered and reused once their energy supply has been depleted. In order to make a WSN feasible, the nodes cannot be expensive.

In terms of a sensor node’s resources, it is clear that since a node is required to be small, cheap and highly energy efficient, such nodes do not have the luxury of an abundance in resources. This means that nodes have little processing capability, limited memory (RAM and ROM) and most importantly, a highly limited resource of energy, usually in the form of a battery with which it must survive for as long a time as possible.

2.2.6 Mobility

Mobility of nodes in a WSN is a factor that is highly application dependant. In many of the more traditional application areas of WSNs, such as environmental monitoring, static networks have been assumed at this stage. However, other applications such as asset tracking clearly require support for dynamic network architectures. It seems from the literature that dynamic WSNs with mobile nodes and/or base stations are still at a less

mature stage than static networks, since they have greater challenges and obstacles to overcome. This may change in the future.

2.2.7 Heterogeneity

It has generally been assumed that WSNs consist mainly of homogeneous sensor nodes. This lies in contrast to Mobile Ad-hoc Networks (MANETs), where devices of very different kinds can form small, wireless, mobile networks, e.g. laptops, Personal Digital Assistants (PDAs), cell-phones etc. Whether the homogeneity assumption will hold in the future remains to be seen. For now, most research has been based on such an assumption.

2.3 Sensor Nodes

Having gained some insight into WSNs in general, the actual sensor node can now be discussed further.

2.3.1 Sensor node architecture

As typical sensor node consists of a few basic building blocks to be able to perform its functions. A simple block diagram of a generic node is shown in Figure 2.2 below.

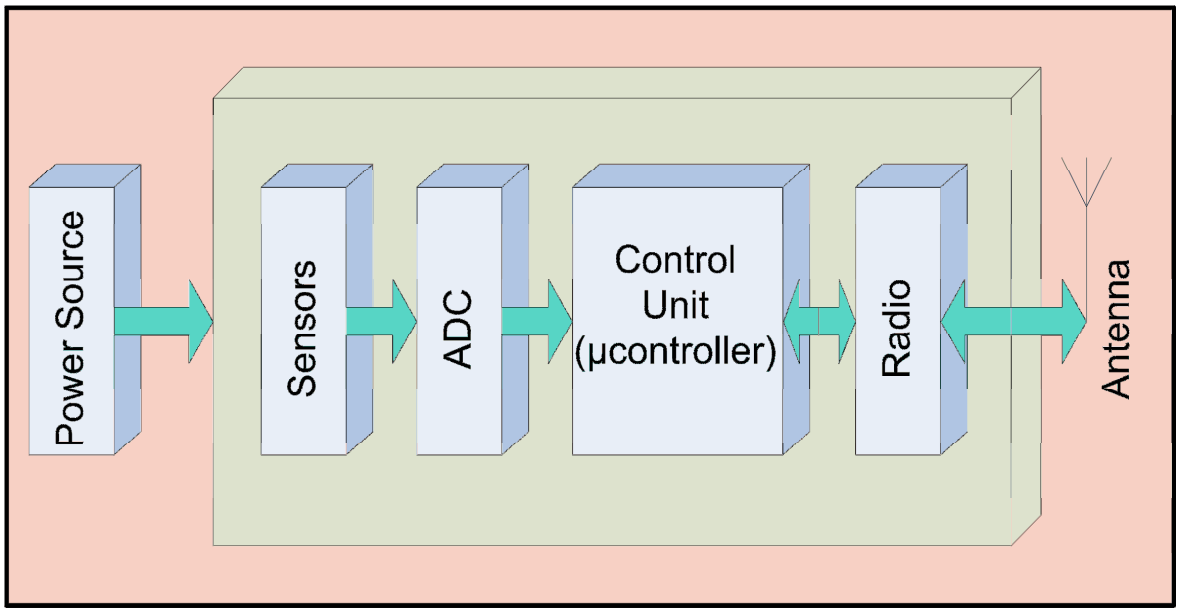

Figure 2.2. Block diagram of a basic, generic sensor node.

From the above figure it is seen that the *Power Source* powers the entire node. It mostly appears in the form of a battery. The block labelled *Sensors* represents any number of different sensor types. The analog output from the sensors is converted to digital format by an Analog to Digital Converter (*ADC*) so as to be usable by the *Control Unit*, which mostly consists of a simple microcontroller. Finally, the *Radio* is used to communicate with other nodes or the base station. Since the control unit and the radio are especially important in terms of the topic of MAC protocols, they will be discussed further.

2.3.2 The control unit

The control unit is responsible for executing all the algorithms and protocols involved in the sensing task as well as the communication with other nodes. For the sake of this research, the communication process is particularly interesting. In essence, the basis for the development of any communications scheme is the OSI reference model as shown below in Figure 2.3.

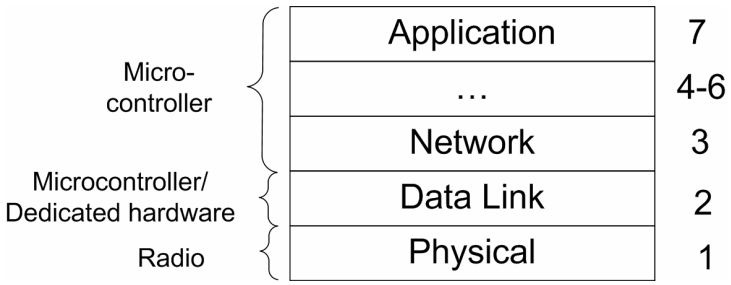

Figure 2.3. The responsibility of the control unit in the communications process in terms of the OSI model.

As can be seen from the above figure, the sensor node's control unit usually performs the upper layer functions of the communications process. The Data Link function may be performed by the control unit but is often performed by dedicated hardware, especially the most timing critical parts of the Data Link functionality. The Physical layer functionality is performed by the radio. In summary, the Data Link layer is responsible for forming and maintaining direct links between neighbouring nodes and to reliably transfer data across these links. Traditionally, the MAC protocol has only been one part of the Data Link layer, however, in WSNs, the Data Link layer mostly consists only of the MAC protocol. The Network layer performs routing and the Application layer contains the particular sensing

and data processing application which the sensor node is performing and which is unique for each WSN. Layers 4-6 have been left out because in the WSN context they are mostly irrelevant. Examples of microcontrollers used in actual sensor nodes so far are the 8-bit AVR microcontrollers from Atmel used on some of the Mica Motes from Crossbow [7] as well as the 16-bit MSP430 microcontroller from Texas Instruments used on the TinyMote which was developed at the University of Vienna [1].

2.3.3 The radio interface

The radio interface of a sensor node links a sensor node to any other nodes which are within its communication range by making use of wireless Radio Frequency (RF) data transmission. Due to the high cost of licensing special RF frequency bands for particular uses, WSNs operate in the license-free Industrial, Scientific and Medical (ISM) frequency bands. Some popular ISM bands occur at 433 MHz, 868 MHz, 915 MHz and 2.4 GHz. The advantage of lower frequency bands is that radio signals sent in these bands have longer transmission ranges at the same sending power due to less signal losses. On the other hand, at higher frequencies, radios can have smaller antennas and higher bandwidths. Thus, even though the higher bandwidth radios require more power to achieve the same range, they also remain active for much shorter periods of time to transmit the same number of bits as compared to low bandwidth radios. This can yield energy savings. Another important radio feature is the time it takes to switch the radio between different states. Short turn-on and send-receive switching times are crucial for energy efficiency.

Two particular radio interfaces were found to be especially interesting. They are the CC2400 [8] and the CC2500 [9] radios from Chipcon. Some of their properties are shown in Table 2.1 below. Also shown in the table are the properties of two other popular radios, the RFM TR1001 and the RFM TR1100. Their summarized properties were obtained from [1]. These are shown for comparison purposes.

	TR1001	TR1100	CC2400	CC2500
Frequency Band	868.35 MHz	916.5 MHz	2.4 GHz	2.4 GHz
Operating Voltage	3 V	3 V	1.8 V	3 V
Current Consumption Sleep Transmit (max power) Receive	 0.7 µA 12 mA 3.8 mA	 0.7 µA 12 mA 8 mA	 1.5 µA 19 mA 24 mA	 0.4 µA 21.2 mA 19.6 mA
Max. Bit rate	115.2 kbps	1 Mbps	1 Mbps	500 kbps
Receiver Sensitivity	-97 dBm	-87 dBm	-87 dBm	-81 dBm
Turn-on Time	RX:51.8,TX:16µs	RX:518,TX:16µs	1.27 ms	1.11 ms
TX/RX Switching Time	TX→RX: 51.8µs RX→TX: 12µs	TX→RX: 518µs RX→TX: 12µs	Both: 40 µs	TX→RX: 21.5 µs RX→TX: 9.6 µs
RSSI Support	No	No	Yes	Yes

Table 2.1. Comparison of four popular RF radios used in sensor nodes.

From the above table it can be seen that although the TR1001 has low power consumption and fast turn-on and switching times, it has a much lower bit rate and does not provide a Received Signal Strength Indicator (RSSI). The TR1100 on the other hand has a high bit rate, but its TX/RX switching time is long, it does not provide an RSSI either and furthermore, use of the frequency band in which it operates is not allowed in Europe. Not shown in the table are the additional features that the Chipcon radios provide, such as Cyclic Redundancy Check (CRC) generator, data buffering, packet handling hardware and data encoding hardware, all of which are absent in the RFM radios and many others. Such considerations make the CC2400 and CC2500 attractive in terms of performance and additional features at this stage of WSN development.

2.4 Example Systems

To round off a high level view of the growing field of Wireless Sensor Networks, a number of current real-world applications employing WSNs are mentioned below, together with some major WSN research projects that are being conducted around the world.

2.4.1 Real applications

In [5], a list of actual WSN implementations is given. A few samples are listed below.

- Bird breeding behaviour monitoring on Great Duck Island, Maine, United States; sensors are hidden in nesting burrows and take various measurements such as temperature, humidity, presence of birds etc.
- ZebraNet at Mpala Research Center in Kenya; sensors are attached to various wild animals to observe their behaviour patterns; data is collected via mobile base stations.
- Glacier monitoring in Brisdalsbreen, Norway; sensor nodes which detect movement are deployed in drill holes in the glacier ice; base station collects data and determines supra-glacial displacements.
- Ocean bed structure monitoring at a wind farm off the coast of England; sensors are dropped to the ocean floor to measure impact of the wind farm on the ocean bed structure; each node is connected to a buoy at the ocean surface which contains radio and GPS equipment.
- Ocean water monitoring, ARGO project; free drifting sensors are deployed in ocean globally; sensors monitor state of the upper ocean; every 10 days, they submerge to a depth of 2000m; when they reach the surface again, they transmit their data to a satellite.
- Vineyard monitoring, Oregon, United States; sensors across a large vineyard monitor factors that influence plant growth to achieve precision harvesting.
- Cold chain monitoring using the commercial Securifood system; sensors travel with the products that need to be kept cold and log temperature data; once shipment arrives at warehouse, the data is collected by relay units which form a multi-hop ad-hoc network. The data is passed to an access box, which sends the data to a central server via the Internet.
- Rescue of avalanche victims; skiers and hikers wear sensor nodes which can measure heart rate and respiratory activity; in the event of an avalanche, the rescue team can use a PDA to receive sensor node data to locate buried people and automate the prioritization of victims.

2.4.2 Research projects

Some of the leading WSN research projects are as follows.

- The PicoRadio project at the Berkeley Wireless Research center together with the SmartDust project also at Berkeley.
- The μAMPS project at the Massachusetts Institute of Technology (MIT).
- The WiseNet project at the Swiss center for electronics and micro-technology (CSEM).
- The WINS project at the University of California, Los Angeles (UCLA).
- The PAWiS project at the Vienna University of Technology.
- The EYES project in Europe.

2.5 Chapter Summary

In this chapter, the Wireless Sensor Network concept has been investigated so as to gain an overview of the current state of research pertaining to this growing field. Some of the design factors influencing WSNs were discussed. Further, the general functioning and architecture of sensor nodes was investigated and finally some examples of real applications and research projects involving WSNs were given.

The aim of the discussions in this chapter has been to introduce the requirements and challenges which should be addressed and kept in mind at all levels of a WSN system. This includes of course the communication protocols of which the MAC protocol is the center of attention in this dissertation.

Chapter 3
MAC PROTOCOLS

Having gained a clearer view of the research context in Chapter 2, in this chapter, MAC protocols are analyzed in more detail. At first, a general description of a MAC protocol's task is given so as to clarify its role in a communications system. Afterwards, some early wireless MAC protocols are discussed to form a point of reference for wireless MAC designs. The focus then shifts to protocols specifically aimed at WSNs. The specific challenges for MAC protocols in WSNs are discussed, followed by an investigation into the many WSN MAC protocols that have already been proposed in the literature.

3.1 Task of the MAC Protocol

In order to begin a discussion of WSN MAC protocols, it is imperative to have an understanding of the task a MAC protocol is supposed to perform.

The MAC protocol forms part of the Data Link Layer (DLL) of the OSI reference model. This is layer two of the OSI model (see Figure 2.3). As its name suggests, the task of a MAC protocol is to coordinate the access to a shared medium among multiple users. Users can be computers, mobile phones, sensor nodes as in the WSN case, and so forth. The medium can be optical fibre, twisted copper pairs, the air as in the WSN case, and so forth. Thus, the MAC protocol determines for users the points in time when they are allowed to transmit data to a specific other user (unicast) or to a collection of users (multicast or broadcast) [10].

When the existence of only a single communication channel in the medium is assumed, then only one pair of nodes is able to communicate over the medium at any point in time. If two or more nodes try to transmit data at the same time, a collision will occur and the data from both nodes will be lost i.e. will not be successfully received by the intended receiver(s). It is also possible to have multiple available channels in the medium, say *N* channels. This can be achieved for example by Frequency Division Multiple Access (FDMA) with *N* frequencies or Code Division Multiple Access (CDMA) with *N* different codes. In that case, coordination of channel access will still be necessary since

1) each receiver can usually only receive data on one channel at a time and thus only one sender is allowed to send to a particular node at a time,
2) the number of nodes will most likely be more than *N* such that *N* is not large enough to supply a separate channel for each pair of nodes and
3) the channels must still be orderly assigned to nodes on a static or dynamic basis.

The above task of the MAC protocol can appear to be rather simple, but as will be shown in the rest of the chapter, this is not necessarily the case.

3.2 Early Wireless MAC Protocols

This section describes some of the early work related to conventional wireless MAC protocols as opposed to WSN MAC protocols. The purpose of this is to show a loosely chronological progression from basic wireless MAC protocols to more and more energy efficient designs. It is believed that this can lead to a better understanding of the thought processes and developments that were involved in the evolution of WSN MAC protocols. Such an understanding can point to underlying trends and possible future developments.

3.2.1 ALOHA and basic CSMA

The evolution of wireless MAC protocols started as early as 1970 with the ALOHA protocol [11]. In ALOHA, any node wanting to send a packet simply sends the packet immediately, without regard for any other nodes or possible ongoing transmissions. Obviously, collisions occur easily in this way. A variant of ALOHA is slotted ALOHA (S-ALOHA) in which time is divided into successive slots large enough for the transmission of a maximum size packet. A node can only start transmission at the beginning of a time

slot and thus the collision probability is decreased by a small margin [10]. When these protocols became inefficient due to heavier traffic loads, the Carrier Sense Multiple Access (CSMA) class of protocols was devised [12] in 1975. In the basic CSMA scheme, the node wanting to send a packet performs a sensing operation to determine the state of the medium. If idle, the packet is sent immediately. If busy, a back-off algorithm is executed to wait for a certain time before sensing the medium again.

3.2.2 Busy Tone Multiple Access

Also in the year 1975, Tobagi and Kleinrock noted the CSMA procedure to be prone to the *hidden-terminal* problem [13]. This problem is shortly explained at the hand of Figure 3.1 below.

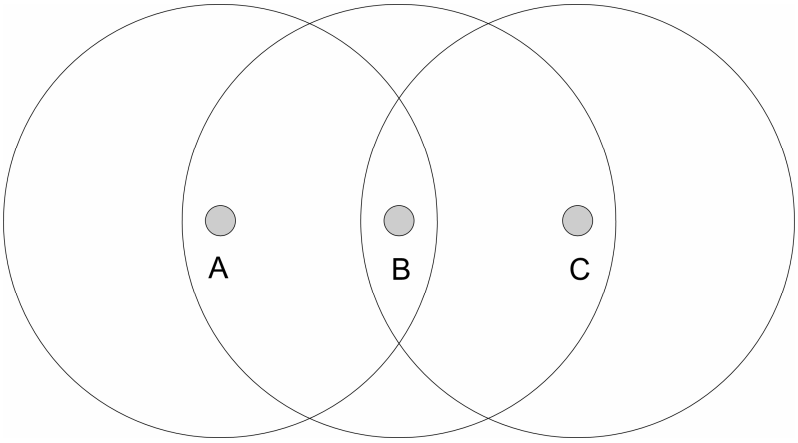

Figure 3.1. A hidden-terminal problem scenario. The large circles represent the transmission/interference ranges of the nodes.

The hidden-terminal problem occurs when *A* is busy transmitting data to *B*. If *C* decides to start a transmission as well, it performs a carrier sense operation. Since *C* is not within range of *A*, the medium is sensed idle. Thus *C* starts to send data and causes a collision at *B*, which wastes precious energy resources. In other words, *A* is hidden from *C* and vice versa.

The authors continued by proposing a modification to the basic CSMA protocol to relieve the hidden-terminal problem. The proposed protocol was termed Busy-Tone Multiple-Access (BTMA). BTMA involves the sending of a busy signal on a separate channel throughout the transmission of data on the data channel. The busy tone is sent by the receiving terminal during reception, which is *B* in this case. The busy tone that *B* sends

while it is receiving data from *A*, lets *C* know that a communication is in progress and thus *C* refrains from sending data. In this way the hidden-terminal problem is relieved.

3.2.3 MACA

A number of years after this initial work, the Multiple Access with Collision Avoidance (MACA) protocol was proposed by Karn [14] in 1990. MACA also addresses the hidden-terminal problem. The protocol is based on CSMA/CA which is CSMA with Collision Avoidance. CSMA/CA uses an initial handshake, the RTS/CTS handshake. A node desiring to send data first sends a Request To Send (RTS) packet to the receiver and the receiver answers with a Clear To Send (CTS) packet if reception is possible i.e. the medium currently has no other users. If the receiver determines that reception is not possible at this stage, it simply does not send a CTS response. Neighbouring nodes overhearing the RTS/CTS handshake are required to refrain from transmission for a certain period of time, usually the maximum packet duration time.

What MACA does is to remove the Carrier Sense (CS) part from CSMA/CA and leaves only the MA/CA part. In other words, no carrier sensing is done. The reasoning for this, as Karn explains, is that the hidden-terminal problem arises due to false sensing of the medium and further, good sensing circuitry is hard to build. Instead of CS, MACA includes in the initial RTS packet the amount of data that the sender wishes to transmit (i.e. the time that the channel will be busy), and this same information is echoed in the CTS response. In other words, The RTS and CTS packets each contain a header field called the Network Allocation Vector (NAV) which indicates for how much longer the medium will be busy. Neighbouring nodes overhearing the RTS/CTS handshake are thus required to refrain from transmission for the period of time indicated by the NAV. This is illustrated in Figure 3.2 below.

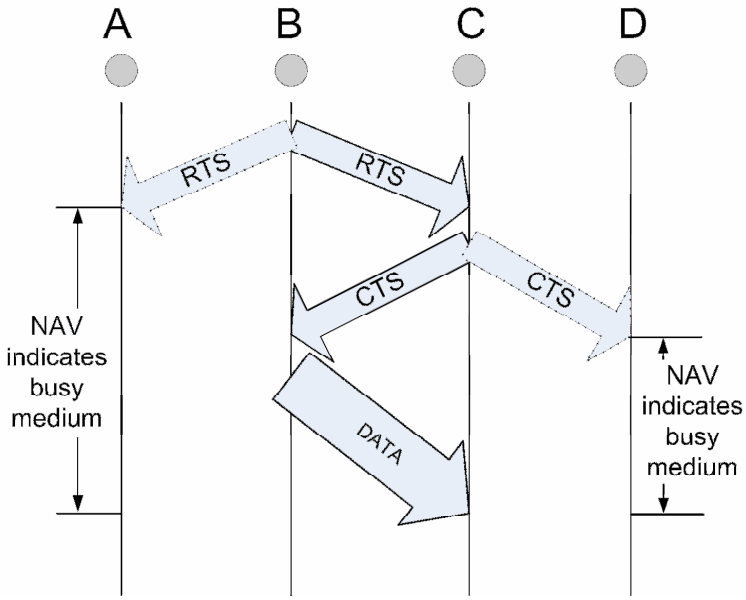

Figure 3.2. Operation of the MACA protocol. (NAV – Network Allocation Vector).

3.2.4 MACAW and IEEE 802.11

Next in the development came the MACAW protocol by Bharghavan et al. [15] in the year 1994. This protocol is based closely on MACA but adds an acknowledgement packet (ACK) which the receiver sends back to the sender after the data has been successfully received. It also makes some changes to the packet size and header contents of MACA and uses a much refined back-off algorithm which is used to determine the waiting time after an unsuccessful transmission attempt. The authors of the protocol show that the changes significantly increased the performance of the MACA protocol. The MACAW protocol was the basis for the development of the widely deployed Distributed Coordinated Function (DCF) protocol in the Institute of Electrical and Electronic Engineers (IEEE) 802.11 standard [10], [16] which was formalized in the year of 1997.

3.2.5 PAMAS

Up to this point in time, no attention had been given to the energy consumption of the MAC protocols. The first step in this direction was the development of the Power Aware Multi-Access with Signalling (PAMAS) protocol proposed by Singh and Raghavendra [17] in the year 1998. PAMAS evolved from MACAW. The unique feature with PAMAS is that when a node overhears an RTS/CTS exchange between two other nodes, it conserves energy by powering itself off for the duration of the data exchange since it need not be

involved in the exchange. The authors of the protocol show that this can yield significant power savings without affecting latency and throughput.

3.2.6 PMAW

In the year 2000, the authors of the Power and Mobility-Aware Wireless (PMAW) protocol [18] pointed out that PAMAS unfortunately lacks the ACK packet used in MACAW. This leaves PAMAS with no guarantee of data delivery. PMAW adds an ACK to the data exchange. Furthermore, the authors of the PMAW protocol show cases in which the PAMAS protocol fails under both static and mobile situations, i.e. when nodes are static or mobile. It is argued that in ad-hoc networks, mobility is the norm rather than the exception and so PMAW aims to include mobility-awareness. Better performance and lower energy consumption than PAMAS are claimed.

3.3 Design Factors for WSN MAC Protocols

Conventional MAC protocols, including the wireless MAC protocols discussed above, have usually been designed with the following main design factors in mind.

- Fairness
- Latency
- Bandwidth utilization
- Throughput

With the emergence of WSNs, the above design factors became much less important. The primary design factor for WSNs has become the need to conserve energy, which is a requirement that has not received much attention for most other applications. For example, the DCF protocol of the IEEE 802.11 WLAN standard [16], as discussed earlier, requires endpoints to permanently listen to the channel so that they will be ready to receive data destined for them at any time. This is referred to as *idle listening*. As is shown in [2], and can be seen from the comparison of RF radios in Table 2.1, idle listening consumes energy on the same order as actively receiving or transmitting data and thus much energy is wasted. This is not acceptable in WSNs.

Ye et al. [19] have identified 4 major sources of energy wastage with respect to MAC protocols as follows.

- *Collisions*: collisions cause wasted reception energy at the destination node, wasted sending energy at the source node and furthermore, retransmission needs to take place after a collision, which leads to the risk of a further collision.
- *Overhearing*: this occurs when a node unnecessarily receives a packet destined for another node; receiving consumes energy and thus this energy is wasted.
- *Control packet overhead*: sending and receiving of protocol control packets consumes energy and also reduces the amount of actual data that can be sent.
- *Idle listening*: this is when a node listens for possible incoming data but such data does not arrive.

Apart from the overriding requirement for energy efficiency of any WSN MAC protocol, a somewhat related design factor is the need for the protocol to have low complexity. This is due to the fact that WSN sensor nodes are simple devices with highly limited resources, and any WSN MAC protocol should thus show low algorithmic complexity as well as low memory requirements.

Lastly, the WSN design factors that were discussed in section 2.2, such as scalability etc. should always be kept in mind when dealing with protocols at any layer of the OSI reference model.

3.4 Existing Literature on WSN MAC Protocols

From the surveyed literature, it is clear that WSN MAC protocols can be broadly categorized in different ways, depending on what aspects of the protocols are considered. Some authors classify the protocols as *distributed* vs. *centralized*. In distributed protocols, every node in the network runs the same algorithms. In centralized protocols, some nodes act as local coordinators or clusterheads to manage the protocol functionality of the surrounding nodes. However, the categorization that is encountered most often is that of *schedule-based* vs. *contention-based* protocols [20]. Some authors, such as [10], call the contention-based protocols *random access* protocols and divide the schedule-based protocols into *fixed assignment* protocols and *demand assignment* protocols. However, the

schedule-based vs. *contention-based* description is simpler and more commonly accepted and so it is used in this dissertation.

It should be noted that some authors add another class to the above contention and schedule-based categorization. This 3rd class is sometimes referred to as *low duty cycle* protocols and other times as *hybrid* protocols. In this dissertation, the approach is taken that such hybrid protocols are grouped into whichever of the two main classes they share the most properties with.

The literature is divided as to which of the two protocol classes is best suited for WSNs since each have their advantages and disadvantages. On the other hand, WSNs can be used in many different application domains. The characteristics of WSN MAC protocols therefore depend very much on the intended application [21], [3], [10] with differences in expected traffic patterns, required bandwidth, size of the WSN, periodicity/randomness of generated traffic, mobility etc. This observation is extremely important and should be kept in mind throughout the rest of this dissertation. It has to be assumed that future WSN deployments will be of wide varieties and different applications will require different hardware and protocols.

In the rest of this section, the existing literature on WSN MAC protocols will be discussed and analyzed. A completely exhaustive analysis could not be performed since there is simply too large a collection of texts. However, it has been attempted to evaluate the most important and relevant literature.

3.4.1 Contention-based WSN MAC protocols

Contention-based MAC protocols are those in which nodes compete with each other for access to the channel. This leads naturally to the possibility of collisions. The early wireless MAC protocols that were discussed in section 3.2 fall into this category. However, these initial protocols were not designed to be appropriate for WSNs as they make little or no attempt to save energy. Approximately at the turn of the century, contention-based MAC protocols specifically intended for WSNs started to appear. They are discussed in the following.

3.4.1.1. PicoRadio MAC

The PicoRadio project of the Berkeley Wireless Research Center is described by Zhong et al. [22]. The PicoRadio MAC is a theoretical concept. It involves the extension of the CSMA technique to a multi channel system such as FDMA or CDMA. The proposal made is that a node will randomly select a channel (frequency/code) to transmit its data. The paper does not explain satisfactorily how the receiver will know on which channel to listen. Furthermore, the protocol proposes that the main radio remain sleeping most of the time and only be awoken for actual reception of data by an ultra low-power wakeup radio consuming less than 1μW running at full duty cycle. To the best of the author's knowledge, such ultra low-power wakeup radios physically do not exist yet. If they did, then the idle listening problem mentioned before would be vastly reduced. Nevertheless, the proposed protocol points to an interesting research area in radio hardware design, which is beyond the scope of this dissertation.

3.4.1.2. Berkeley MAC

The Berkeley MAC protocol proposed by Woo and Culler [23] is designed mainly for fair bandwidth allocation. This is achieved by an additive increase multiplicative decrease automatic rate control mechanism which not only adjusts the sending rate of each node but also tries to phase periodic sending of data which causes collisions. In other words if nodes periodically send data and two or more nodes have similar sending times, then the algorithm phase shifts the periodic transmissions of both nodes so that they will not collide repeatedly. Furthermore, the protocol reduces the hidden-node problem without the use of RTS/CTS exchanges. This is done by letting a node sleep for an extended time after it has forwarded a packet so that the next node will have enough time to forward the packet further. In other words, a node waits till it believes that its packet has been forwarded two hops away before starting transmission again. The authors of the protocol conclude that the protocol achieves good fairness and energy efficiency. It must be questioned however what is meant by energy efficiency, since this protocol does not display any significant energy saving mechanisms. In fact, apart from the energy savings due to the absence of the RTS/CTS overhead and the sleeping after packet forwarding, the protocol seems rather similar to the early CSMA protocols.

3.4.1.3. S-MAC

A major step in the development of WSN MAC protocols was the proposal of Sensor MAC (S-MAC) by Ye et al. [19]. Basically this protocol was inspired by PAMAS but took much more radical steps to reduce energy consumption. Latency and fairness were considered much less important goals. In essence, S-MAC tries to reduce the enormous energy wastage incurred by idle listening. This is accomplished by letting each node sleep for a large percentage of time and only become active for short listen periods. This is known as a duty-cycled approach. The basic operation of S-MAC is shown in Figure 3.3.

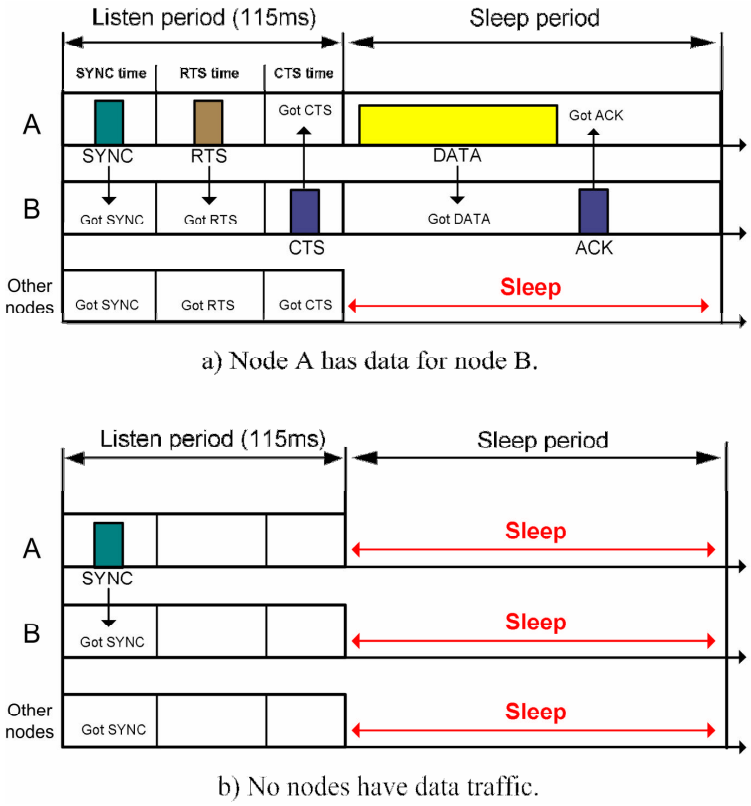

Figure 3.3. Operation of the S-MAC protocol. Taken from [24] (p. 2976, Fig. 1).

A complete cycle of listen and sleep interval is termed a *frame*. The listen period duration is fixed according to physical- and MAC layer parameters. Ye et al. have chosen a listen interval of 115ms but for different underlying hardware, this changes (i.e. for a low bit rate radio, the listen interval has to be longer than for a high bit rate radio).

The *duty cycle* is defined as the ratio of listen time to frame time. Thus a duty cycle of 10% would mean that during a frame time, a node only listens for 10% of the time and sleeps

for 90% of the time. This means that in order to achieve low power consumption, the duty cycle should be low. The choice of duty cycle depends on the requirements of the application since the lower the duty cycle, the lower the power consumption but also, the less traffic can be successfully delivered. Typical values for duty cycle that have been observed in the literature lie between 1% and 10%.

The listen interval is divided into three parts as shown in Figure 3.3 above. The first part is the SYNC interval. The purpose of the SYNC interval is for neighbours to exchange SYNC packets. The SYNC packets are required for two reasons. Most importantly they inform a node's neighbours of its sleeping pattern so that they will be able to send data to the node at the right time. S-MAC is designed so that neighbouring nodes form virtual clusters within which sleep schedules are loosely synchronized so that all nodes in the cluster can be reached with a single SYNC packet. Nodes that lie between two virtual clusters need to adhere to both schedules. In this way, the protocol is a kind of hybrid between a contention-based and a schedule-based protocol. The second reason for the SYNC packets is for nodes to maintain clock synchronization among neighbours so as to overcome clock drift. Clock drift occurs because the quartz crystals on a node are not perfect and thus different nodes' clocks can have small frequency differences which accumulate over time causing nodes to become unsynchronized. This makes it necessary for nodes to exchange SYNC packets at regular intervals to keep them synchronized. The synchronization is not strict and thus the overhead of tight time synchronization is avoided.

After the SYNC interval, the rest of the listen period is for actual data exchange to be initiated. Immediately after the SYNC interval is the RTS interval, where nodes contend for the medium in order to send RTS packets if the have data to send. Any packet sending follows the standard CSMA procedure, i.e. a carrier sense is done, if the medium is idle, the packet is sent, else back-off is performed. The node that successfully sends its RTS packet then waits for a CTS response from the destination. If this occurs, then data exchange can take place while all other nodes in the virtual cluster go to sleep. The data reception is then acknowledged with an ACK packet.

It is clear that the per-hop latency of S-MAC will be approximately equal to the sleep period, which can add up to a significant amount over multiple hops. In a later paper by the same authors [20], Ye et al. add the adaptive listening scheme to S-MAC. This scheme

approximately halves the per-hop latency as follows. Suppose that node *A* wants to send data to node *B* and that node *C* overhears the RTS/CTS exchange of *A* and *B*. Node *C* will then schedule an extra listen period for the time when the transmission between *A* and *B* will end (*C* knows this time from the NAV duration field in the RTS or CTS packets) even though this might be outside *C*'s actual listen period. This is done because *C* might be the next hop for the packet sent to *B*. In this manner, the latency is halved.

S-MAC and its subsequent adaptation by its authors has been one of the most significant contributions to the WSN MAC protocol field. It has been quoted by a vast body of subsequent literature and many times forms a basis of comparison for new proposed protocols.

3.4.1.4. Adaptations and derivatives of S-MAC

Due to the significant impact that S-MAC has had on the research area of WSN MAC protocols, it has been widely studied and analyzed. A number of authors have subsequently proposed adaptations to the original protocol or proposed protocols closely related to S-MAC. These are discussed here.

3.4.1.4.1. T-MAC

The Timeout-MAC (T-MAC) protocol was proposed by van Dam and Langendoen [25] and has subsequently also become a popular basis of comparison for many new protocols. T-MAC introduces an adaptive duty cycle (ratio of active time to total period). This is achieved by dynamically ending the active period when no activation event has occurred for a certain amount of time. An activation event can be

1) the firing of a periodic frame timer,
2) the reception of any data on the radio,
3) the sensing of communication on the radio, e.g. during a collision where no valid data is received,
4) the end-of-transmission of a node's own data packet or acknowledgement, or
5) the knowledge, through overhearing prior RTS and CTS packets, that a data exchange of a neighbour has ended.

This has the result that messages buffered during the sleep period are sent in a burst at the beginning of a frame. This means that T-MAC can adapt to traffic volume and type. Under

homogeneous load, S-MAC and T-MAC perform at a similar level, but under variable load, the authors of the protocol claim that T-MAC outperforms S-MAC by a factor of up to five. This can be very valuable in terms of reducing idle listening and thus saving energy.

3.4.1.4.2. AC-MAC

The Adaptive Coordinated MAC (AC-MAC) was proposed by Ai et al. [26]. This protocol, similarly to T-MAC, applies an adaptive duty cycle scheme to S-MAC. In this case the number of packets queued at the MAC layer is used as a measure of traffic. Based on this measure, the duty cycle is adapted so that different traffic patterns can be efficiently supported. Energy consumption is claimed similar to S-MAC but latency and throughput are closer to those obtained by 802.11, especially under heavier loads. This protocol is thus more applicable to circumstances where many packets flow through a network and stricter latency requirements are needed than can be achieved with S-MAC.

3.4.1.4.3. TEEM

The Traffic Aware, Energy Efficient MAC (TEEM) protocol proposed by Suh and Ko [24] also reduces idle listening. Firstly, compared to S-MAC, the actual listening time is shortened from 115ms to 83ms. Then, the SYNC and RTS slots are combined into one transmission. The protocol is best explained in Figure 3.4 below which should be compared to Figure 3.3 to visualize the differences between S-MAC and TEEM.

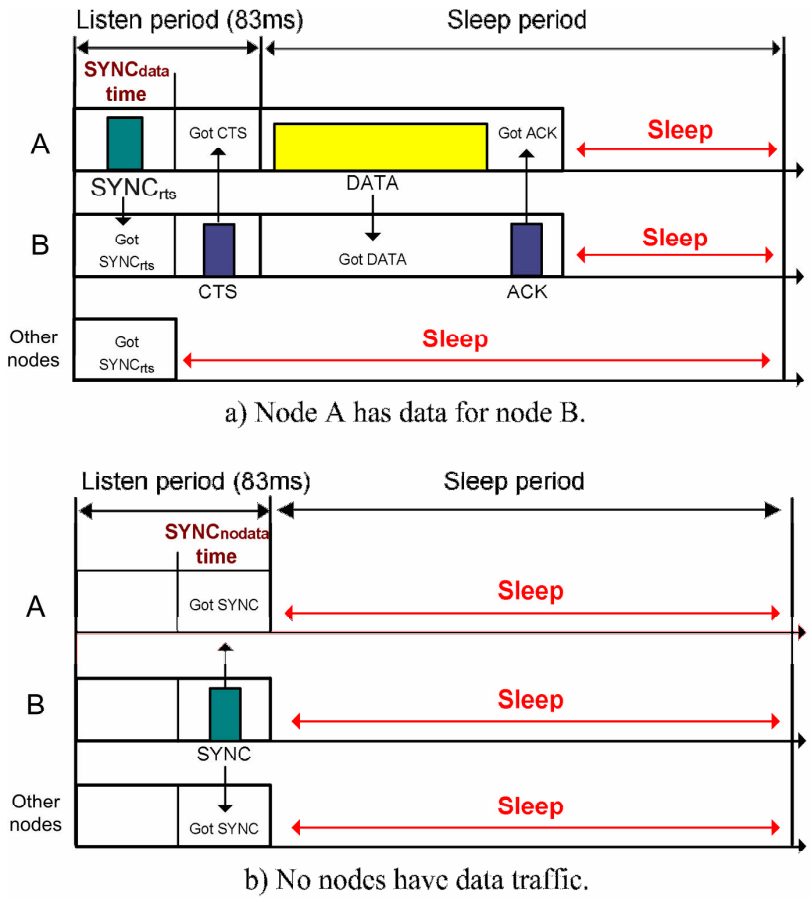

Figure 3.4. Operation of the TEEM protocol. Taken from [24] (p. 2977, Fig. 3).

The authors of the protocol go on to show that TEEM has more sleep time, less control packets and lower energy consumption than S-MAC.

3.4.1.4.4. DMAC

A somewhat different approach to improving S-MAC is taken by Lu et al. [27] in their development of DMAC. This protocol is specifically designed for WSNs where the major traffic pattern consists of data collected from several source nodes to a sink through a unidirectional tree. The idea is that in both S-MAC and T-MAC, data forwarded to the sink will be halted temporarily at virtual cluster boundaries due to synchronization differences. In other words, at some node, the data will have to wait for the listen interval of the next hop. Depending on the size of the WSN, these delays could add up to a significant amount since a virtual cluster is by definition a small collection of neighbouring nodes and thus many virtual clusters may need to be traversed, with a corresponding increase in delay probability. What DMAC attempts to do is to organize the active/sleep schedules in a

staggered manner from the edge of the network towards the sink so that once a packet is sent from a node, it will continually be forwarded since the active period of the next hop for the packet will start as soon as the packet has arrived at the current node. The DMAC authors show that for tree-based applications that require real-time data delivery, DMAC outperforms S-MAC both in terms of low latency and energy savings. It is easy to see that if data would need to travel in the reverse direction, the delay would be near worst case, since the active period of the next hop for the packet would have been over just a short while before the packet arrived at the current node. Obviously, this protocol is suitable for the case where such reverse traffic is not needed. However the protocol overhead necessary to organize the schedules in the proposed way could be quite significant.

3.4.1.4.5. MS-MAC

The Mobility-aware MAC protocol for Sensor networks (MS-MAC) was proposed by Pham and Jha [28]. In essence it attempts to extend S-MAC to the case where a sensor network consists not only of static nodes but also of mobile nodes. It is pointed out that the resynchronization interval of standard S-MAC is fixed at 2 minutes. In other words, a number of SYNC packets are sent by each node every 2 minutes. If a new node joins the network, it will thus have to wait for at least 2 minutes to learn the schedule of its neighbours so that it can participate in meaningful communications. This is much too long to support communication with a node that travels through a network. Thus, whereas protocols such as 802.11 are inherently good at supporting mobility, S-MAC is not.

In MS-MAC, a node discovers mobility from the received signal strength of the periodic SYNC messages received from its neighbours. If there is a change in the signal strength, then a node deduces that either the neighbour or the node itself is moving. The amount of change gives a reflection of the speed of movement. Information about the maximum speed of any mobile neighbours is included by a node in its SYNC packets. With the help of this extra information, the recipients of the SYNC packets can form an *active zone* around any mobile node as it moves from one cluster into the next. Within this active zone, nodes perform the resynchronization more often. On the one hand, this causes higher power consumption, due to more overhead, but on the other hand, a mobile node can set up connections to neighbours in a new cluster it is moving into much quicker.

One problem with this approach is that mobility information is solely derived from received signal strength. It is known that the path loss experienced by a signal on route from sender to receiver is not a static variable [29]. In fact, this path loss varies continuously, and in a random manner, because of factors such as signal reflection, diffraction and scattering, different antenna gains in different directions etc. Since in MS-MAC, such changes in signal losses could have a significant effect on the resynchronization interval and thus on the power consumption, the surrounding environment in which this protocol is applied must be of such a nature that random path loss fluctuations are minimal. This may not always be possible.

3.4.1.4.6. DSMAC

The Dynamic Sensor MAC (DSMAC) protocol was proposed by Lin et al. [30]. The authors note that S-MAC was designed with energy efficiency as the main goal and latency was traded off for this design goal. They state that there are several applications, such as those in military and medical areas, which are sensitive to delay and thus propose a new protocol to deal with this issue.

As in T-MAC and AC-MAC, DSMAC applies the idea of a dynamic duty cycle to S-MAC. In other words, the frame time is changed on-the-fly. The basis for changing the frame time is the average latency measured by the node. The authors define the one-hop latency as the difference in time between when a packet gets into the queue and when it is successfully sent out. This value is included in the packet header of the outgoing packet and thus the receiver can also retrieve it. At the end of a synchronization period the receiver calculates an average latency from all the received values in the different packets that were received in that past synchronization period. Furthermore, a sensor also keeps track of its own energy consumption. If, at the end of the synchronization period a node notices an intolerable latency value, and if its energy consumption is below a threshold, it decides to double its duty cycle by decreasing the sleep interval accordingly. The updated duty cycle is sent in the SYNC packet of the new synchronization interval and the upstream neighbours will be able to send data to the node more often (thus reducing latency) because the node will be awake more often. The duty cycle is only doubled or halved within certain boundaries, so as to keep the fundamental period the same. In other

words, upstream neighbours will still be able to reach the node with the updated duty cycle in their own normal awake interval.

In summary, this is a novel scheme which could find application in networks where throughput and latency are important.

3.4.1.4.7. MAC-CROSS

MAC-CROSS was proposed by Suh et al. [31]. The authors noted that in the adaptive listen S-MAC protocol, which was proposed by the original developers of the S-MAC protocol as described earlier, there is the possibility of significant energy wastage caused by the extra listen interval. Recall that in adaptive S-MAC, any node overhearing an RTS/CTS exchange not destined to itself will wake up at the end of the neighbouring data transfer in case it is the next hop for the packet. This is done even though it is outside of the normal listen interval. However, as pointed out by Suh et al., only one of the possibly many neighbours overhearing the RTS/CTS exchange will actually be the next hop and thus all the other affected nodes will wake up for no reason.

What MAC-CROSS does is to use valuable information from the routing layer to determine which nodes should wake up and which shouldn't. Such sharing of information across different protocol layers is referred to as cross-layering. This is the reason for the name MAC-CROSS. The way in which the protocol operates is demonstrated at the hand of Figure 3.5 below.

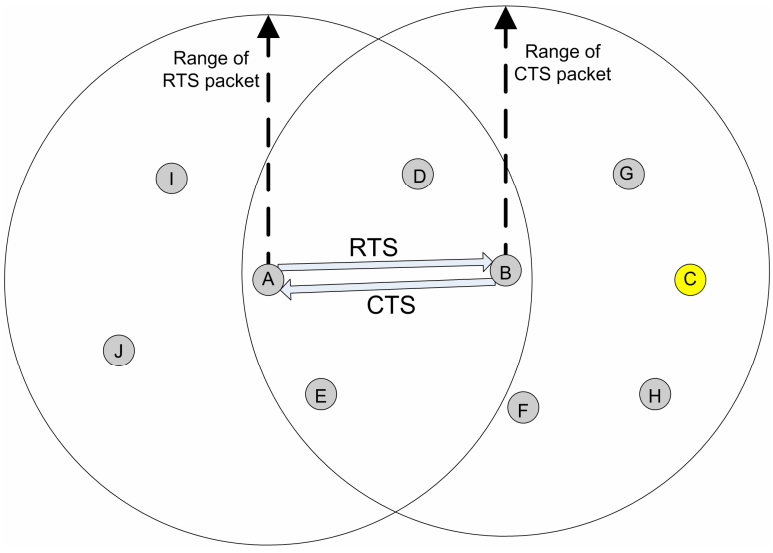

Figure 3.5. MAC-CROSS next hop selection.

When *A* wants to send data to *B*, it starts with an RTS packet. Once the MAC protocol at *B* has received this packet, it consults the routing tables from the network layer and thus determines which node is the next hop for the packet to be received from *A*. Say *B* determines that *C* is the next hop, it includes this information in the CTS response sent to *A* but also received by all other nodes in its reception range, which are *C*, *D*, *E*, *F*, *G* and *H*. Node *C* notes that it is the next hop and thus wakes up at the end of the data exchange between *A* and *B*, as indicated by the NAV in the CTS packet. *B* can then immediately forward the packet to *C*. All other nodes listed above see that they are not the next hop and thus do not schedule an extra listen interval at the end of the data exchange. By doing this, they conserve a considerable amount of energy. The next hop value in the CTS packet can also be a list of nodes such as if a broadcast packet needs to be sent.

3.4.1.5. STEM

The Sparse Topology and Energy Management (STEM) protocol proposed by Schurgers et al. [32], [33] is in fact not a complete MAC protocol but it suggests a solution to the idle listening problem. The assumption made is that most of the time, a WSN will only be monitoring the environment waiting for an event to occur. This is called the monitoring state. Once an event takes place, the nodes start to send data and the WSN moves to the transfer state. STEM concentrates on the monitoring state. Once in the transfer state, any other MAC protocol can be used to accomplish data delivery.

STEM attempts to use as little power as possible in the monitoring state. This is achieved by letting the nodes sleep for long periods and only wake up for very short intervals to listen for incoming data. A sender has to transmit a train of beacon packets towards the intended receiver until it hits the receiver's listen period. These beacons contain the MAC address of both the sender and receiver. Once the receiver notices the beacon stream, it replies with an acknowledgement packet so that the sender can stop sending beacons. The receiver then continues to listen and receives the actual data. There is thus a trade-off between energy efficiency and wakeup latency. The longer the sleep period, the less energy is consumed and the higher the transmission latency. Furthermore, monitoring and transfer states can coexist in neighbouring nodes and thus in order to avoid the wakeup beacons from interfering with other ongoing data transmissions, two separate radios are used, one for data and the other for wakeup packets.

3.4.1.6. CSMA-PS

CSMA with Preamble Sampling (CSMA-PS) was proposed by El-Hoiydi [34], [35]. It is similar to STEM and was also developed at about the same time. Unlike STEM, it is a complete MAC protocol, and is aimed to operate on nodes continually, not just in a monitoring state. The main differences between CSMA-PS and STEM are that CSMA-PS can operate on a single radio and a single channel or optionally additional data channels. Furthermore, the preamble is sent for the entire length of the sleep period to make sure that the receiver's listen slot is intersected. Thus unlike STEM, it does not provide for an ACK packet to end the preamble early so as to save energy.

3.4.1.7. WiseMAC

The same author that proposed CSMA-PS was later involved in the development of the WiseMAC protocol as part of the WiseNet project [36], [37]. This protocol is very similar to CSMA-PS but makes some adjustments so as to decrease the time that the preamble is sent. As in CSMA-PS, every node has a periodic listen slot. The length of the listen slot is as small as possible, the minimum length needed to detect whether data is being received or not. The time between listening slots is the listening interval T_w, which is fixed for every node of the network. The authors of WiseMAC use T_w values in the range of 100ms-100s for their simulations. Upon network initialization, a node does not have knowledge of any of its neighbours' listening schedules. Thus, if it wants to send a packet to one of its neighbours, it must precede the packet with a preamble packet of duration $T_{preamble} = T_w$ to ensure that the receiver's listen slot is "hit". As in the CSMA protocol, a Carrier Sense (CS) operation is done before sending to ensure that the medium is idle. At the receiver, when the periodic listen slot starts, the incoming preamble is detected. The node then continues listening until the subsequent data packet has been received. At this point, the receiver responds with an ACK. This is shown in Figure 3.6 a below.

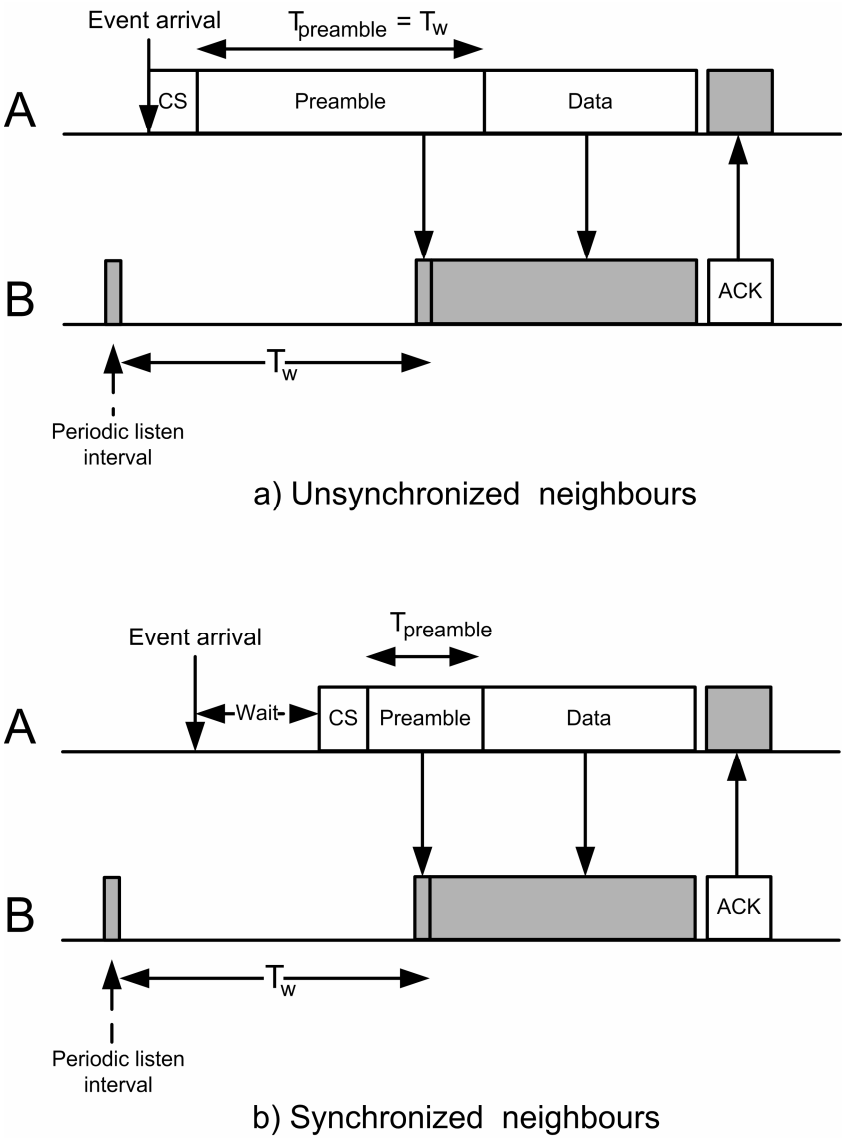

Figure 3.6. Operation of the WiseMAC protocol in the unsynchronized and synchronized state (CS – Carrier Sense).

Every ACK packet includes a header field which indicates the receiver's listen schedule. Thus at the point when a sender node receives an ACK packet from one of its neighbours, it then has an estimate of when that neighbour's next listen slot or any subsequent listen slot starts. The next time the node wants to send a packet to the same neighbour, the length of the preamble can be shortened. It is shown that if both nodes' oscillators have a maximum frequency tolerance of $\pm\Theta$ and the two nodes had their last communication L seconds ago, then the preamble need only be of length $T_{preamble} = \min(4\Theta L, T_w)$. The reason for this is that in the L seconds since the last communication, each of the nodes' clocks could have drifted from the absolute time by as much as $\pm\Theta L$ seconds. In the worst

case, if node A's clock has drifted by +ΘL and node B's clock has drifted by -ΘL, then the synchronization error between node A and B is +2ΘL. On the other hand if node A's clock has drifted by -ΘL and node B's clock has drifted by +ΘL, then the synchronization error between node A and B is -2ΘL. Thus to take both of these possibilities into account, the minimum preamble length should be 4ΘL. However, if L is large and 4ΘL is greater than T_w, then obviously the preamble length need only be T_w. The sender starts sending 2ΘL seconds before the estimated start of the receiver's listen slot and ends preamble sending 2ΘL seconds after the estimated listen slot as shown in Figure 3.6 b. Thus the receiver is guaranteed to receive the preamble. Once the sender receives the ACK packet from the destination, the estimated listen schedule is again updated and L is reset to zero.

WiseMAC makes use of a table at each node which contains the listen schedule of its neighbours. The algorithm is completely isolated at each node and yet loose synchronization is maintained among nodes. With the knowledge of a neighbour's listen schedule, the sender can make the preamble much shorter than in CSMA-PS. The result is a network that reduces energy consumption more and more as the synchronization between nodes becomes more accurate.

3.4.1.8. CSMA-MPS

The CSMA with Minimum Preamble Sampling (CSMA-MPS) protocol was proposed by Mahlknecht and Böck [38] shortly after the WiseMAC protocol, of which it is essentially an improvement. In fact, it uses the best attributes from STEM and WiseMAC and combines these to further reduce energy consumption. From STEM, it borrows the approach of during the preamble, alternating between sending short preamble packets and listening for a response. As soon as the destination hears a preamble packet it immediately responds with a short preamble ACK to enable the sender to stop sending preamble packets and continue with the actual data. This ensures the sender need not send unnecessary preamble packets when the destination is already awake and listening. From WiseMAC, it uses the estimate of the receiver's listen schedule to start the preamble sending just at the right time, taking into account the maximum frequency tolerance Θ. It is shown that the preamble time can be significantly reduced in comparison to WiseMAC.

In another paper by the same authors [39], the very interesting observation is made that even though high-speed radios consume more energy, they are also active for shorter

periods of time. By simulations, it is shown that high-speed short bursts of transmission actually consume less energy than low speed transmission. The authors notice that none of the actual sensor node hardware implementations from the different research institutions have made use of this finding and most use lower-speed radio transmissions.

3.4.1.9. B-MAC

The Berkeley MAC (B-MAC) protocol proposed by Polastre et al. [40] is not to be confused with the Berkeley MAC protocol mentioned in section 3.4.1.2, which was developed a number of years prior to B-MAC. B-MAC is similar to CSMA-PS. Nodes also have a periodic listening interval. During the short listen slots, the channel is checked for activity. If an incoming preamble is detected, the node carries on listening, if not, the node returns to sleep. The time between listen slots, T_w, used in B-MAC ranges from 10ms-1.6s. As with CSMA-PS, preambles are sent for a time $T_{preamble} > T_w$. Strangely enough, the authors refer to the WiseMAC protocol and state that it meets most of their needs "except that it has no mechanism to reconfigure based on changing demands from services using the protocol". The reason for this being strange is that the B-MAC protocol does not make use of WiseMAC's novel method of reducing the preamble length even though Polastre et al. clearly drew some other ideas from WiseMAC.

The B-MAC protocol is implemented in TinyOS on the Mica2 Mote from Crossbow as a small core of functionality. The implementation provides a software interface for upper layers to dynamically adjust certain properties of the MAC protocol. These properties include the length of T_w, enabling/disabling carrier sense (called Clear Channel Assessment) and enabling/disabling ACK packets. One of the features of the protocol is the Clear Channel Assessment (CCA) scheme used. Basically, incoming signal strength samples are taken when the channel is assumed to be free such as directly after a transmission. These samples are stored in a FIFO queue which is used as a low-pass filter to build up an accurate estimate of the noise floor. Then, when a carrier sense operation must be performed, the transceiver takes a number of successive channel samples, instead of just one as is the common method. These samples are compared with the noise floor and when none of them lie significantly below the noise floor, the channel is deemed busy. This method is claimed to provide improved results i.e. fewer false sensing operations.

Apart from this, even though B-MAC provides an interface for changing the listen interval T_w during normal network operation, the protocol does not provide a means for such changes to be effectively communicated to neighbouring nodes. It is in fact rather simple to locally adjust the T_w value, however, how to coordinate such a change among neighbours is not so trivial. B-MAC leaves this responsibility to higher layers. This can be viewed as a cross-layered approach, since upper layers would not usually be tasked with such responsibilities.

3.4.1.10. SpeckMAC

The SpeckMAC protocol proposed by Wong and Arvind [41] in fact consists of two different improvements to the basic operation of B-MAC, namely SpeckMAC-Back-off (SpeckMAC-B) and SpeckMAC-Data (SpeckMAC-D). These two schemes make adjustments to B-MAC as illustrated in Figure 3.7 below.

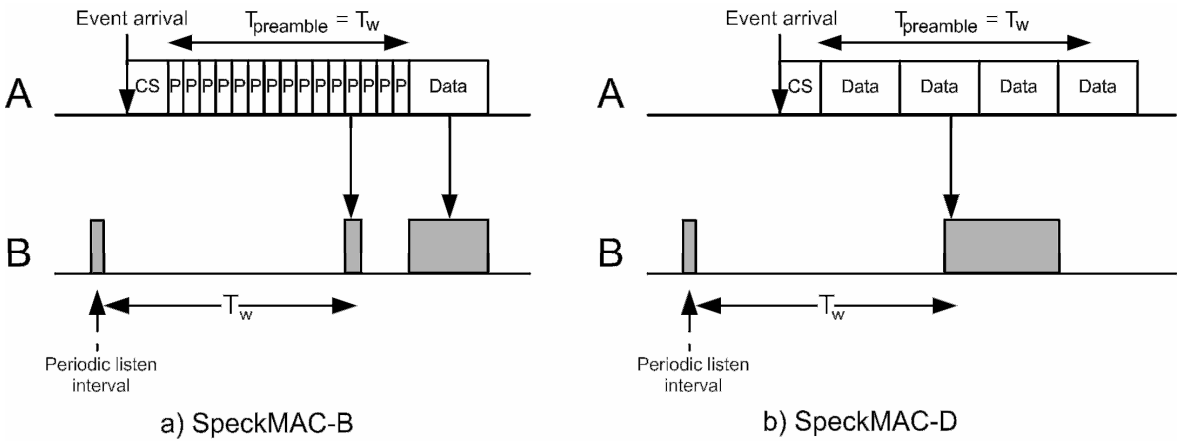

Figure 3.7. The operation of the two SpeckMAC protocols.

In SpeckMAC-B, instead of sending one long packet to wake up the destination, the preamble consists of a train of short packets. Each of these Wake-Up (WU) packets contains timing information about when the actual data packet will be sent. Thus, as soon as the destination has successfully received a WU packet, it can extract this timing information and return to sleep mode, waking up just in time to receive the data packet.

In SpeckMAC-D, the sending node simply sends repetitions of the actual data packet as a preamble. When the receiver detects reception at the start of its listen slot, it carries on

listening until it has received a whole data frame. At this point, the node goes back to sleep.

In essence, both SpeckMAC-B and SpeckMAC-D relieve some of the energy costs of the *receiver* as compared to B-MAC. The energy spent on preamble *sending* remains the same and is worse than for both WiseMAC and CSMA-MPS. Also notable is the absence of any ACK packets and thus, reliability could be a concern. Especially in SpeckMAC-D, the sending of an ACK packet is in fact not possible with the proposed scheme, since the sender could continue sending the preamble for an arbitrary time after the receiver has already received the data packet. The receiver thus does not know the point in time at which to send an ACK packet, except if it continues to listen to the data train until the last packet, in which case, there would be no gain in energy efficiency over B-MAC.

3.4.1.11. CSMA/CA2

In [42], the application of a MAC protocol called CSMA/CA2 to the WSN domain is discussed. The specific WSN scenario is a WSN that is event driven, where the detection of a certain event can trigger a burst of data from surrounding nodes, leading to a high probability of collision. Standard 802.11 uses Exponential Binary Back-off (EBB) to control the rate at which individual users send data. If a data transmission was successful, the rate is increased. If a transmission failed, the rate is decreased. It is explained why the use of an Additive Increase, Multiplicative Decrease (AIMD) function is more effective at reducing collisions and thus saving energy than the EBB function.

It is quite clear from the discussion that the protocol does not include any functionality that allows nodes to sleep. It is thus questionable how this protocol could be applied to a WSN node with severely limited energy resources. The protocol seems more apt for situations in which high and bursty traffic rates are expected and energy resources are less constrained than in typical WSNs.

3.4.1.12. Modified CSMA/CA

In [43], another CSMA-based MAC protocol is discussed. Here, an optimal frame size predictor is incorporated into the standard CSMA/CA algorithm. A frame size predictor is software that predicts what packet length should be used to transmit data. In this case, the frame size predictor consists of an Extended Kalman Filter (EKF), which uses channel

quality measures to predict the optimal frame size. The idea is to send longer packets when the channel quality is good (pump lots of data through the channel) and to send shorter packets when the channel quality is degraded to reduce losses caused by bit errors.

Although this is a novel idea, firstly, the EKF calculations are rather complex. It might be the case that they cannot be implemented efficiently on limited hardware resources. Secondly, and even more critically, the protocol again does not give any indication of any sleeping patterns. In fact, the radio is always on in the description of the protocol. This is not acceptable for the vast majority of WSN types.

3.4.1.13.nanoMAC

The nanoMAC protocol [44] is another CSMA/CA-based protocol. When referring to CSMA protocols, the term *persistence* is often encountered. In summary, there exist 3 types of CSMA protocols, categorized by their back-off behaviours. These are *1-persistent, p-persistent* and *non-persistent*. In 1-persistent CSMA, when there is data to send, the protocol first does a carrier sense. If the medium is idle, the data is sent. If not, the protocol keeps listening and sends when the medium becomes free (i.e. with probability 1). In the non-persistent case, the protocol will also do a carrier sense and send the data if the medium is free, but if it is busy, it will back-off for a random interval and then again perform a carrier sense operation. In the meantime, the radio can be switched off. Lastly, the p-persistent case is usually used for slotted systems, where a node can start transmissions only at the beginning of a new slot. What the protocol does is if the medium is found idle, it will transmit with probability p and with probability $q = 1-p$ it will defer until the next slot. If the next slot is also idle, the same procedure will be repeated.

In essence, nanoMAC adapts the p-persistent case to an un-slotted system. With probability p the protocol will behave in a non-persistent way and with probability $1-p$ the protocol will refrain from sending even before a carrier sense is performed and chose a new time for carrier sensing. It is shown that nanoMAC consumes less energy than standard CSMA.

As with the previous two protocols, this protocol is not really applicable to highly energy constrained WSNs.

3.4.1.14.CC-MAC

The spatial Correlation-based Collaborative MAC (CC-MAC) protocol [45] makes use of the assumption that in a dense WSN, when a new event is detected or a certain phenomenon is sensed, then not all sensors registering this event or phenomenon need send a sensor update to the sink station. Only a subset of sensor readings needs to be delivered to the base station to enable it to build a sufficiently accurate view of what is taking place. In other words, sensor readings from nodes in the same geographical region will likely be highly correlated and thus, if all these readings were to travel across the network, a lot of redundant information would find its way to the base station and thus a lot of energy would be wasted. CC-MAC attempts to prevent such a situation.

The protocol starts out with the Iterative Node Selection (INS) algorithm which is performed at the sink. This algorithm takes as input the desired maximum field estimation distortion D_{max}, as well as certain statistics of the sensor network deployment, such as the spatial density of the sensor nodes (nodes/km^2). The output of the algorithm is the required correlation radius r_{corr}, which is the radius within which the detection of an event is highly correlated. The correlation radius is distributed to all sensor nodes in the network as part of network initialization. During normal operation of the network, the CC-MAC protocol then runs on the nodes. It in fact consists of two separate protocols, namely Event-MAC (E-MAC) and Network-MAC (N-MAC). E-MAC elects for each event a *representative* node which is responsible for collaborating the accurate sensing of the detected event. When a number of nodes detect a new event, only the representative node sends a report on the event to the sink node. The other nodes determine whether they are located within the correlation region of the representative node or not. If they are, they refrain from reporting on the detected event. If they are not within the correlation radius, they attempt to become a representative node in their own area and send data to the sink. N-MAC manages the forwarding of route-through traffic towards the sink.

The described protocol makes some assumptions that may not always be valid. The statistics of the network deployment may not always be known as required by the INS algorithm, and furthermore, statistics such as node density may vary widely over different regions of the network. Furthermore, E-MAC needs a way to ascertain whether a node's geographic position falls within the correlation radius or not. This may not be a trivial task.

The CC-MAC protocol may well be suitable for certain specific applications where the statistics of correlated sensor readings can be accurately predefined.

3.4.1.15. Discussion of contention-based protocols

A number of interesting contention-based MAC protocols have been investigated in this section. In general, the contention-based MAC protocols for WSNs have been simple and flexible. They make energy savings possible by having extended sleep periods, where the node is inactive. Contention-based protocols also seem suitable for low-traffic scenarios, since in such situations, a node's duty cycle can be made very low leading to increased energy savings.

It was noted in [46] that for practical low-power radio implementations, receive power consumption is often higher than transmit power consumption due to the fact that in receive mode, a greater number of signal processing circuits need to be active. This trend was already observed for the CC2400 radio (see Table 2.1), which draws a current of 19mA in transmit mode and 24mA in receive mode. The conclusion reached in [46] is that "It's more power-efficient to blindly transmit than to blindly receive - for the same amount of time". The preamble sampling protocols which have been investigated in this section exploit this finding and seem especially suitable for low-traffic scenarios.

Again, it should be kept in mind that different protocols are suitable for different application scenarios. For example, the MS-MAC protocol is possibly suitable for the case of mobile nodes, S-MAC with high duty cycle could be used in high-traffic cases, T-MAC could be suitable for variable traffic load and the CSMA/CA adaptations could best be applied to MANETs as opposed to WSNs.

3.4.2 Schedule-based WSN MAC protocols

Schedule-based protocols are also referred to as reservation-based protocols. A typical example is a TDMA system where participating nodes are assigned a repetitive time slot in which they can communicate. Such systems are implicitly immune to collisions introduced by contention. This is because a certain time slot is always reserved for a particular node and thus no other node can communicate in this time slot. It is clear however, that this basic scheme has not only advantages but also some downsides. Reserving a time slot for a certain node means that when that node does not have any data to receive or send in its

slot, this time is wasted since no other nodes can use it even though they might have data to communicate. Another downside of these systems is that they generally require nodes to form genuine clusters with tight time synchronization. Furthermore, inter-cluster communication is not an easy task. Neither is adding nodes to a cluster in a network. Scalability is thus compromised in schedule-based systems compared to CSMA systems. Time synchronization also incurs quite some signalling overhead and as noted in [10], cheap sensors will most likely contain cheap oscillators, which will result in significant clock drift and more overhead. Nevertheless, proposals for schedule-based protocols in WSNs do exist which attempt to leverage the advantages of the basic scheme and reduce the disadvantages as much as possible. Some of these are mentioned next.

3.4.2.1. LEACH

The Low-Energy Adaptive Clustering Hierarchy (LEACH) protocol was proposed by Heinzelman et al. [47]. It is in fact a protocol suite, which contains a TDMA-based MAC protocol, together with topology management (clustering) and a simple routing protocol. Nodes are partitioned into clusters. Each cluster needs a clusterhead which is responsible for managing the TDMA schedule. The protocol provides for periodic rotation of the clusterhead responsibility among the nodes so that energy consumption remains fairly distributed in the network. This is because the clusterhead must communicate directly with the sink node which is usually further away and so higher transmission costs result on top of the additional computation cost resulting from schedule management.

The assumption that each clusterhead can communicate directly with the sink node is not valid for all WSN types and thus, a specific application area may be targeted.

3.4.2.2. SMACS

The Self-organizing MAC for Sensor networks (SMACS) protocol was proposed by Sohrabi and Pottie [48] and Sohrabi et al. [49]. It is part of a suite of algorithms which is based on a TDMA system. SMACS is a distributed protocol which lets nodes discover their neighbours and set up TDMA schedules for communicating with them without the need for a local or global master node. The TDMA-like schedule takes the form of a super-frame which consists of a number of slots which represent time slots for individual nodes. The content of the super-frame can change from time to time such that limited mobility of nodes is supported. The setup of this super-frame is the responsibility of the SMACS

protocol. A critical decision is the length of the super-frame. If it is too short, then the nodes that have not been assigned a slot will simply not take part in the network. If it is too long, empty slots will be wasted.

It is unfortunate that the authors state in [49] that: “contention-based channel access schemes are clearly not suitable for sensor networks, due to their requirement for radio transceivers to monitor the channel at all times”. This gives the indication that the authors did not do much research into the trends of contention-based protocols that were discussed earlier.

Altogether, the research papers are more concerned with the procedure of setting up a TDMA-like ad-hoc network quickly and efficiently, rather than to develop an actual WSN MAC protocol.

3.4.2.3. TRAMA

The Traffic-Adaptive Medium Access (TRAMA) protocol was proposed by Rajendran et al. [50]. It is pointed out that the SMACS protocol suffers from the problem that time slots are wasted if a node does not have any data to send. TRAMA avoids this situation.

The protocol divides time into a scheduled-access period and a random-access period. During the scheduled-access periods, actual TDMA data exchange takes place, whereas during the random-access periods, management takes place. This includes the broadcasting of neighbourhood information. By capturing these neighbourhood packets from their neighbours, nodes can learn about their two-hop neighbourhood which is necessary to avoid time slot conflicts with two-hop neighbours. Furthermore nodes also broadcast their schedule information which is a list of receivers of packets currently in the node’s queue. Using this data, nodes execute a rather complex distributed scheduling algorithm which assigns nodes to each time slot and tells all the other nodes that are not sending or receiving to sleep during that time. The conclusion reached by Rajendran et al. is that in comparison to 10% duty cycle S-MAC, TRAMA has higher delays, comparable energy consumption but high delivery guarantees. Once again, the use of either protocol depends on the intended application.

3.4.2.4. PACT

The PACT protocol was proposed by Pei and Chien [51]. The protocol is described as having the following attributes.

- The duty cycle is adapted to traffic so that the radio is off if the network is inactive.
- Passive clustering is used to take advantage of the redundant dense topology and in doing so the lifetime of the entire network is prolonged. At a given time, only a subset of network nodes (i.e. clusterheads and gateways) is active.
- Clusterhead and gateway duties are rotated and are not elected by probabilistic methods as in LEACH. Rather, remaining battery power is used to make this decision.
- The clustering requires no explicit control messages and therefore incurs negligible energy overhead.

In terms of energy savings, the paper compares TRAMA only with 802.11, which is certainly not an energy efficient MAC protocol. It is questionable whether the proposed protocol in fact can compare with representative contention-based protocols in low traffic scenarios, since the protocol seems to be more applicable to high-traffic networks, as is mostly the case with schedule-based MAC protocols.

3.4.2.5. TRACE

The Time Reservation using Adaptive Control for Energy efficiency (TRACE) protocol was proposed by Tavli and Heinzelman [52]. It is quite a sophisticated centralized TDMA protocol with a single controller node. The controller responsibility is again rotated to spread energy consumption among the nodes. The protocol contains features such as data summarization to minimize transmission costs, as well as a backup controller node to improve reliability. Quality of Service (QoS) guarantees are also included. It should be mentioned though that this protocol is only meant for small single hop networks where emphasis is more on throughput and latency than on low power consumption. The protocol could nevertheless be useful in certain WSN situations.

3.4.2.6. BitMAC

The BitMAC protocol was proposed by Ringwald and Römer [53]. The protocol assumes a network that is a spanning tree, with the sink node as the root. The network is organized

into concentric rings around the base station where ring *n* consists of all nodes that are *n* hops away from the sink. A node in ring *n* will have a number of children in ring *n+1*. A node and its children form a small star topology. In each star, TDMA is used to organize communications. Further, neighbouring stars use different frequencies assigned by a graph colouring algorithm to avoid interference. Data can then travel either from parents to children or from children to parents.

Several techniques are presented which are used to set up and maintain the connected star topology and manage data exchanges. The authors of the protocol show that when radios of nodes are bit synchronized and On Off Keying (OOK) modulation is used (radio on = 1, radio off = 0), then when a number of nodes simultaneously transmit a single bit, the logical OR of all transmitted bits is received at the destination. With some manipulation, other operations (AND, MIN and MAX) can also be implemented using simultaneous OOK transmissions. These operations are used in the graph colouring algorithm that determines the frequency at which each node must transmit in a rather complex manner.

Furthermore, when a parent transmits a packet to all its children and the children are supposed to each reply with a single bit (e.g. yes or no), then in most protocols this would require each child to send an entire packet with synchronization preamble (a few bytes of alternating 1's and 0's) as well as a Start Of Packet (SOP) delimiter field. However, since all children were already bit-synchronized by the initial packet received from the parent, the preamble and SOP are actually not necessary and each child can in fact respond one after the other with a single bit to the parent immediately following the parent's packet.

Once in normal operating mode, the protocol proceeds in rounds. At the start of each round, the parent transmits a broadcast beacon to all its children. The beacon includes information about whether this is an upstream or downstream round. If upstream (from parent to child), the beacon is followed by the actual data with a target address. If downstream, the children respond with send requests if they have data to send, else they remain silent. These responses take place in an organized manner, in that the node with the lowest ID sends a single bit, followed by the next lowest ID also a single bit and so forth as described in the previous paragraph. Once the parent knows which nodes want to send, it constructs a schedule, broadcasts it to the children and data transfer then takes place

according to the schedule. If more nodes requested send slots than are possible in one round, the nodes that were not assigned a slot will be given preference in the next round.

It can easily be seen that this protocol is extremely complex compared to simple contention-based protocols such as B-MAC. Furthermore, there is a great deal of overhead communication involved in keeping the TDMA schedule running. The energy consumption of this protocol would compare unfavourably with some of the contention-based protocols in low traffic WSNs.

3.4.2.7. DE-MAC

The Distributed Energy-aware MAC (DE-MAC) protocol was proposed by Kalidindi et al. [54]. The protocol adapts the normal TDMA operation by treating critical nodes (nodes with low remaining energy resources) differently than other nodes. If a node's energy value falls beneath a certain threshold, it can initiate a local leader election phase which is completely integrated into the normal TDMA overhead and thus does not cause additional energy consumption. During the leader election phase, the initiator node sends its energy level to its neighbours. An election then takes place and the node with the lowest remaining energy is elected the leader. The leader is then in charge of TDMA slot assignment and can assign the majority of slots to itself. Since a node must be awake and listening during other nodes' slots to potentially receive data from them, the leader can sleep for longer since the other nodes have fewer slots. This means that the critical nodes will always spend more time sleeping than the other nodes and thus their energy is preserved. As soon as another node's energy falls beneath that of the winner of the previous election phase, it starts the leader election process again.

3.4.2.8. ER-MAC

The Energy and Rate-based MAC (ER-MAC) protocol [55] is very similar to DE-MAC discussed above. It introduces the term *energy-criticality*. As in DE-MAC, a node has to be awake in its neighbours' time slots to receive potential data from them. In its own slot, a node can sleep if it has nothing to send. The changes made to DE-MAC are that a local election is started when a participating node's energy level falls below a threshold *factor* of the previous winner's energy level (the level that the winner had at that stage). Furthermore, the election scheme takes into account not only the remaining energy of each node but also the traffic flow rate to calculate a value of criticality of a node. Thus the

decision of which node will be the leader is based not only on the energy, but on the *energy-criticality* value which includes traffic flow rates.

3.4.2.9. TDMA-W

The TDMA-Wakeup (TDMA-W) protocol was proposed by Chen and Khokhar [56]. It is another adaptation of the basic TDMA scheme. As the authors of the protocol explain, in traditional TDMA, the communications activities of the participating nodes are organized into a frame, which contains a number of time slots. Every node is assigned a time slot. In TDMA-W, each node is assigned two slots, one called the Send slot for transmission, and one called the Awake slot for listening/reception. The organization procedure for setting up the schedule is described by the authors but not repeated here.

During normal operation mode, a node maintains two counters for every neighbour, namely the send counter and the receive counter. If a node wants to send data to another node, it sends a wakeup packet in the wakeup slot of the intended destination. The destination thus knows that it should listen in the send slot of the source node and wakes up at this time to receive the packet. It then sets the receive counter to the initial fixed value. The node will keep on listening to the sender's send slot in subsequent frame periods. Each time it does not receive something from that neighbour during a frame period, the receive counter is decremented by one. If a packet arrives from the sender during a frame period, the counter is reset to the initial fixed value. Once the counter reaches zero, the node stops listening to that neighbour's send slot. On the sender side, once a wakeup packet has been sent to a certain node, the send counter is initialized with the same fixed value as the receive counter. Every time a packet is sent to that same destination during a frame period, the counter is reset to the initial value. Every time no data is sent to the destination during a frame period, the counter is decremented by one. As long as the send counter is non-zero, the node need not send a wakeup packet to the destination if data needs to be sent to it. Thus, as soon as the counters reach zero, both the sender and receiver consider the link inactive.

This protocol could be very effective in networks that carry a fair to high amount of data with static traffic patterns.

3.4.2.10.Z-MAC

The Z-MAC protocol was proposed by Rhee et al. [57]. A hybrid approach is taken in the design of this protocol. Z-MAC attempts to incorporate the best of both CSMA and TDMA-type protocols. Even though the authors state that the baseline of Z-MAC is CSMA with hints taken from TDMA, the protocol includes the concept of a recurring frame with time slots for individual nodes and thus it is grouped under schedule-based protocols.

Every node in the network owns a time slot in a frame. Any nodes that are more than two hops apart can reuse the same time slot in a frame. Unlike standard TDMA however, a node can transmit in any time slot. Before any transmission, a node does a carrier sense as in CSMA and only sends if the medium is idle. The only requirement is that the owner of a slot is given first priority to send data. This is achieved by giving slot owners earlier chances to transmit. If the owner thus has something to send, transmission can be started immediately. If not, then any non-owner can steal the slot by noting the empty slot when carrier sensing is performed shortly after the slot start. The network initialization procedure with slot assignments etc. is also explained.

By experiment, Z-MAC is compared to TDMA and B-MAC. In terms of energy consumption, B-MAC performs better in a low contention situation.

3.4.2.11.EMACS

The EMACS protocol was proposed by van Hoesel et al. [58] as part of the EYES project. In this TDMA scheme a node can be in one of three states at any given time, namely *active* (it participates in all normal network activities including routing), *passive* (it does not perform routing or similar activities but only keeps track of one neighbouring active node) and *dormant* (goes into a sleep state for a certain time, e.g. to recharge battery). Furthermore, every node's slot in the frame is divided into three sections as shown in Figure 3.8 below.

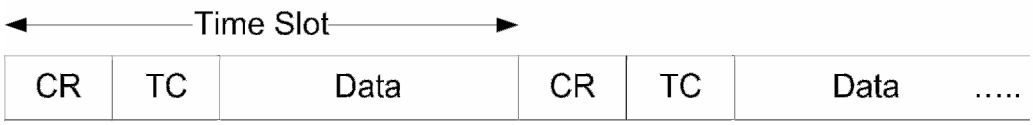

Figure 3.8. Time slot structure of EMACS.

In the Communication Request (CR) section a node accepts sending requests from any passive node. If one is received, the passive node can be granted in the TC section the right to send data in the data section. During the Traffic Control (TC) section, the node always transmits a control packet, which, besides possible acknowledgements of the requests in the CR section, contains all necessary control and synchronization information such as the slot schedule table. These packets are the “heartbeat of the network” as the authors state. New nodes in the network need only listen for such a packet to be able to synchronize to the time slot rate. Whereas passive nodes use the CR section for send requests, active nodes communicate such requests to their neighbours in the TC packet as well. Any data transfers requested in the CR or TC sections then occur immediately in the data section. The local topology information learned from TC packets can be valuable to any routing protocol in building a view of the network. For the choosing of time slots, nodes run a distributed algorithm that uses only local knowledge, but needs quite a lot of message passing. Another algorithm determines which nodes should be active and which should be passive in the local neighbourhood in order to still maintain a fully connected network. It seems that, once decided, these roles are static.

The overhead incurred by this protocol seems large indeed, especially since every active node has to listen during the CR section and always transmit during the TC section in every slot belonging to it.

3.4.2.12. Discussion of schedule-based protocols

Apart from the schedule-based MAC protocols that were discussed above, a number of additional protocols have been studied in this category. They are not all described in detail, due to their similarities to the already mentioned protocols. They include Bit-Map-Assisted MAC (BMA MAC) [59], Lightweight MAC (LMAC) [60], Mobility-adaptive collision-free MAC (MMAC) [61], CDMA Sensor MAC (CSMAC) [62], Adaptive Information-centric and Lightweight MAC (AI-LMAC) [63] and Flow Aware Medium Access (FLAMA) [64].

It is apparent from the literature that in general, schedule-based MAC protocols are suitable for WSNs with medium to high-traffic characteristics as opposed to truly low-traffic characteristics. They are also more suitable for WSNs where energy efficiency as a

design goal is not by far more important than traditional design goals such as throughput, delivery guarantees etc.

3.4.3 Comparison of protocol classes

Having gained an overview of various WSN MAC protocols in each of the two broad protocol classes, a short comparison of their attributes is given in Table 3.1 below.

	Contention-based	**Schedule-based**
Strong points	• Simple • Flexible • Robust • No (strict) synchronization required • Topology changes handled easily • Simple network initialization procedures	• Mostly collision-free • Automatic overhearing avoidance • Not prone to hidden terminal problem
Weak points	• Collisions likely • RTS/CTS causes overhead • RTS/CTS can cause hidden terminal problem	• Complex network initialization procedures • Strict time synchronization required causing added overhead • Inflexible to topology changes • Schedules often built on communication range of nodes instead of on interference range, thus schedules may not be interference free
What happens in low traffic?	• Low delays or • Energy can be saved by low duty cycling	• Low channel utilization and high delays
What happens in high traffic?	• Many collisions degrade channel throughput	• Throughput and bandwidth usage maximized

Table 3.1. Comparison of contention- and schedule-based MAC protocols.

From Table 3.1 it is apparent that schedule-based MAC protocols are more efficient when traffic volumes and thus contention is high, whereas contention-based MAC protocols are more efficient when traffic volumes and thus contention is low.

3.4.4 WSN MAC-related standards

There are three main standards that are applicable to most wireless systems. These are the 802.11 standard, the 802.15.1 standard and the 802.15.4 standard.

The 802.11 standard was already mentioned in section 3.2.4. It is also known as WiFi and applies to Wireless LANs (WLANs) which typically connect computers to each other in a manner similar to Ethernet except with no wired network. It was noted in section 3.2.4 that the MAC protocol of the 802.11 standard does not make energy efficiency an important goal and thus, it is not applicable to WSNs [10].

The 802.15.1 standard, defines the PHY and MAC layers for Wireless Personal Area Networks (WPANs) based on the Bluetooth specification [65]. WPANs can also be called ad-hoc networks and are used to establish connections between devices such as cellphones, PDAs and laptops. In Bluetooth, a WPAN takes the form of a *piconet*, in which one of the devices becomes the master and the others become slaves. A piconet can only accommodate seven slaves. The master, amongst other things, manages the TDMA MAC schedule in the piconet. Bluetooth/802.15.1 has not found application in sensor networks [10] due to factors such as the severe topology limitations and power consumption which is not low enough for WSNs.

The IEEE 802.15.4 standard, closely related to ZigBee, defines a low-cost, low data rate radio interface system including both the PHY and MAC layers specifically aimed at Low-Rate WPANS (LR-WPANS). It is more interesting to the field of WSNs because of the more proactive steps taken in order to save energy compared to Bluetooth/802.15.1.

The 802.15.4 MAC layer is a very complex protocol and will be only partially discussed here [10]. The 802.15.4 MAC protocol defines two types of nodes.

- A Full Function Device (FFD) can take on the roles of a PAN coordinator, a simple coordinator or a device.
- A Reduced Function Device (RFD) can only be a device.

A group of devices are associated with a simple coordinator, forming a small star network. Multiple simple coordinators form a PAN and one of the coordinators is designated as the PAN coordinator. The MAC protocol for communication among the coordinators is different from the one for communication between a simple coordinator and its attached devices. For WSNs, this latter communication process is more interesting and is described below.

In the *beaconed* mode, coordinators are responsible for setting up and maintaining a superframe structure which organizes channel access. The superframe itself is divided into an initial beacon, followed by a Contention Access Period (CAP), a number of Guaranteed Time Slots (GTSs) and then a sleep period. The beacon provides the point of synchronization, by defining the beginning of the superframe. It contains the PAN identifier and some other parameters. During CAP, a slotted CSMA/CA scheme is used for transmission, whereas the GTSs are used for scheduled TDMA access. Data exchange can only take place between the coordinator and a device, i.e. devices cannot communicate with each other directly. In order to become the owner of one of the GTSs, a device can issue a request to the coordinator during the CAP. Once this has been done, a node can freely transmit and receive in its slot. If a device does not desire a GTS, it can request data transfers to/from the coordinator in the CAP. Thus, in beaconed mode, every device must listen for the periodic beacons in order to know the timing of the CAP and GTS sections. Furthermore if a device has a GTS, it must listen during its slot. Otherwise, devices can sleep the rest of the time and wake up only to send or to retrieve data from the coordinator. In the *nonbeaconed* mode, the coordinator does not send any beacons and furthermore, only unslotted CSMA/CA is used, with no GTS period. In this case, a device can follow its own sleep schedule but must regularly check whether there is any data destined to it waiting at the coordinator by sending a data request packet. In essence what this means is that in beaconed mode, the protocol behaves much like a schedule-based protocol, whereas in nonbeaconed mode, it behaves like CSMA/CA.

One characteristic of the IEEE 802.14.5 MAC is that it is optimized for the case where there are energy un-constrained nodes mixed with energy constrained sensors. This may not always be the case. In [37], WiseMAC is compared to 802.14.5 and WiseMAC displays lower energy consumption. It will still need to be seen whether the 802.15.4 standard is useful in certain types of WSNs.

3.4.5 Additional observations

In this section, a short discussion of two additional observations that were made during the literature review is given.

Firstly, it was noted that a number of authors addressed the issue of sensor data correlation in a sensor network. Among these are [45], [66], [67]. It is the opinion of the author that such functions should not necessarily be dealt with at the MAC layer, or only in certain applications. The way in which spatial correlation among sensor nodes can be dealt with is not the same for every type of WSN. In some WSNs, spatial correlation filtering could require complex mathematical formulae. In other WSNs, these formulae might be very simple. In still other WSNs, correlation might be a total non-issue, all data being required to be reliably delivered to the sink. This gives an indication that such a technique should rather be implemented at the application layer. It is understood that the proposals mentioned above try to save valuable time by performing these functions within an integrated MAC layer and thus for their specific applications, the proposals might be extremely well suited. Nevertheless, in this dissertation, sensor data correlation will not be dealt with.

Related to the above observation, it was also interesting to note that the degree of integration of the MAC protocol with the rest of the software on a sensor node can vary widely. In the traditional OSI layered approach, the protocols at each layer form more or less autonomous entities. In WSNs, such clear boundaries between different layers are not always present. Two extreme scenarios are as follows.

- For a simple and tightly compacted application, a MAC protocol such as B-MAC might well control every part of a node's functionality. Since B-MAC has a periodic listen interval, the MAC protocol code might be adapted so that during every x^{th} listen interval, a sensor reading is taken and pushed to the back of the send buffer. When the next hop is awake, the data is sent. In the meantime, the entire node sleeps when the MAC protocol is in SLEEP state. Such a system would typically be one in which a monolithic piece of software, coded in assembler, is programmed onto a node.
- In more complex applications, the application layer could be totally separate from the bottom layers, which in themselves could also be separate entities (e.g. routing

protocol and MAC protocol). In this case, every layer would simply treat the layer beneath it as an incoming and an outgoing link. The application and MAC layer activity periods for example need not be synchronized at all in this case. While the MAC protocol is in the SLEEP state, the application layer could be performing calculations or taking sensor readings etc. An example of a system that has these characteristics is the Mica Mote platform from Crossbow [7]. The nodes in this system run the TinyOS operating system which can be used to build more modular software.

3.5 Focus Area

From the research conducted on WSN MAC protocols in this chapter, it is noted that many of the proposed approaches are focused on WSNs that have more or less high traffic volumes. Since this is a rather vague statement, it is easier to state the corollary. It is noted that only a limited number of the MAC protocol proposals are aimed specifically at WSNs with low traffic rates, where by low traffic rates, packet inter-generation times on the order of minutes up to hours are referred to. In other words, a node will take a sensor reading once every few minutes or hours and then forward this data in a data packet towards a sink node. In the time between such transmissions, the node could theoretically be switched off except for the necessity for it to forward other nodes' packets towards the sink as well.

There are a number of WSN applications that in fact display traffic patterns of the kind described above. As an example, a WSN monitoring the annual temperature variation in a remote region, say a rain forest area in the Amazon, can be considered. Such a WSN would typically not take temperature readings very frequently in a day. A reading might be taken once every hour, or twice a day or even less frequently. In this case, the traffic volume flowing through the WSN would be very low. In fact, the low volume of traffic might be a strict necessity to enable the network to survive for a number of years in order to fulfil its purpose. Some of the real existing WSN applications mentioned in section 2.4.1 would also display such low traffic volumes. These include the glacier monitoring, ocean bed structure monitoring and vineyard monitoring application examples.

The specific WSN application area that will be focused on from here on can therefore be summarized as follows.

- Low traffic.
- Long message inter-arrival times.
- Latency and other traditional network measures not important.
- Long required lifetime (on the order of years)
- Example applications:
    - Environmental monitoring (e.g. building/home monitoring, temperature/ rainfall mapping, pollutant tracking etc.)
    - Precision agriculture (e.g. vineyard monitoring)

The existing MAC protocols that most closely resemble applicability to the described focus area are the contention-based protocols (see section 3.4.3) due to their inherent advantages at low traffic rates. However, even many of the contention-based proposals were not specifically designed for such low traffic rates. Thus, more specifically, the protocols that are found to be useful for the described scenario are the preamble sampling protocols, which were already singled out in the discussion on contention-based protocols as having certain attractive properties (see section 3.4.1.15 for more details). It is the author's opinion that the limited available literature directly applicable to low traffic WSN applications, with their stringent power consumption constraints, warrants further research to advance the MAC protocols in this area.

3.6 Chapter Summary

In this chapter the focus has been on Medium Access Control protocols for Wireless Sensor Networks. A description of the requirements and challenges of WSN MAC protocols was first provided. Subsequently, an overview of the literature on WSN MAC protocols to this date was formed. Existing WSN MAC protocols were grouped into two main classes. The applicability of different protocols as well as the overall protocol classes to various application domains was discussed. This led to the conclusion that insufficient research has been performed on MAC protocols designed specifically for WSNs with low traffic rates. Such low traffic WSNs were then singled out as the focus area for the rest of this dissertation.

The aim of this chapter has been to investigate more closely and set in context the exact research topic that is addressed in this dissertation. A more exact view of the problem statement has thus been developed.

Chapter 4
MAC PROTOCOL DESIGN

The literature review of existing WSN MAC protocols in Chapter 3 revealed that insufficient research has been conducted on protocols specifically aimed at low traffic WSNs. In this chapter, a new MAC protocol for such low traffic WSNs will be proposed. The new protocol aims at increased energy efficiency compared to existing protocols. At first, the most relevant existing protocols forming the design basis will be reviewed. The deficits of the existing designs will then be pointed out, followed by a detailed explanation of the proposed new design.

4.1 Design Basis

4.1.1 Applicable existing literature

After the analysis of existing WSN MAC protocols in the previous chapter, the conclusion was reached that only a few of the proposed MAC protocols so far have aimed specifically at the area of low traffic WSNs with required network lifetimes on the order of a number of years. It was also noted that the preamble sampling protocols most closely resemble applicability to these WSNs. The preamble sampling protocols are STEM, CSMA-PS, WiseMAC, CSMA-MPS, B-MAC and SpeckMAC. The main differences among these protocols are shown in Table 4.1 below.

	STEM	CSMA-PS	WiseMAC	CSMA-MPS	B-MAC	SpeckMAC
Sender keeps track of receiver's schedule?	No	No	Yes	Yes	No	No
Receiver can end preamble sending early?	Yes	No	No	Yes	No	No
Receiver only listens to first valid preamble packet?	Yes	No	No	Yes	No	Yes

Table 4.1. Comparison of preamble sampling protocols.

Each of the questions in the first column of Table 4.1 above refers to a specific energy conserving technique. Firstly, if the sender keeps track of the receivers listen schedule, then the preamble length can be minimized as first proposed in WiseMAC. Secondly, if the receiver can end the preamble sending early, such as first proposed in STEM, then the sender need not continue sending a preamble once the receiver's listen slot has been reached. Lastly, if the receiver only need listen to the first valid preamble packet then the receiver can save receive time by not listening to the rest of the preamble. This should not be confused with the previous question. In other words, the sender might not be able to end the *sending* of the preamble early (as STEM and CSMA-MPS are able to) but still might be able to end its own *listening* to the preamble early. The protocol which can do this is SpeckMAC, which only listens to the first valid preamble frame, goes to sleep again and wakes up just in time for the actual data. This last technique is only applicable to protocols that send a train of short preamble packets as opposed to one long preamble packet.

It is seen from Table 4.1 that only WiseMAC and CSMA-MPS keep track of their neighbours' listening schedules and attempt to start sending preambles only shortly before the estimated start of the listen slot. This technique can save a significant portion of energy and is simple to implement. All that is required is an additional field in a node's neighbour table which indicates the last communication time with each neighbour. From this an estimate of a neighbour's next listen slot can be derived. Furthermore, it is also seen in Table 4.1 that the only protocol which applies all three of the energy saving techniques is CSMA-MPS. This protocol can thus be viewed as the one with the best energy conservation characteristics and it will form the design basis of the proposed new MAC protocol of this dissertation.

4.1.2 Key challenges

All of the preamble sampling protocols reduce the idle listening problem by having a node sleep for most of the time. The nodes only wake up at regular intervals to sample the medium for the minimum amount of time necessary to determine whether there is an incoming packet or not. In an ideal scenario, a sender would know the exact point in time when the intended destination would wake up and send the data at that instant. The receiver would wake up to listen and immediately start receiving the data. In the ideal scenario there would furthermore be no collisions of data packets i.e. two nodes would not send data to the same destination at the same time. The ideal case is shown in Figure 4.1 below. Node *A* and *B* both send data packets to *C* in consecutive listen slots. No preamble is necessary since the timing is exact.

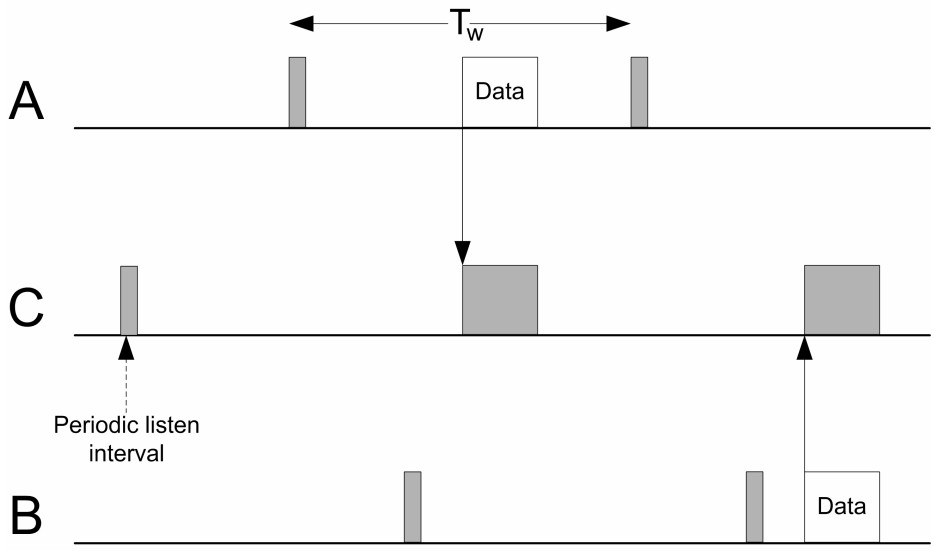

Figure 4.1. Behaviour of a duty cycled MAC protocol in ideal circumstances.

The reason why the ideal scenario cannot be attained is summed up in the term *clock drift*. There is not a single clock that has ever been devised which can maintain a perfect and absolute time signal. This applies especially to cheap sensor nodes which contain cheap quartz crystals as frequency source. As was already explained in section 3.4.1.7, a crystal oscillator frequency tolerance of ±Θ can cause timing errors on the order of ±ΘL, where L is the time since last clock resynchronization. Typical values of Θ for crystal oscillators can be in the region of 10-100ppm (parts per million).

A specification of Θ = ±30ppm for an oscillator means that during an elapsed time of L seconds, the oscillator could have drifted away from absolute time by as much as ±Lx30ppm = ±Lx30x10^{-6} seconds = ±Lx30μs. Say L=1 second, then the oscillator could have drifted by ±30μs in that one second.

The only research on MAC protocols which was found to directly address this issue of clock drift in a quantifiable manner consists of the WiseMAC and CSMA-MPS protocols. It was noted though that clock drift was treated as a single fixed value, Θ, without investigating the properties of this value or what exactly it represents. In the next section, clock drift is investigated in some more detail.

4.2 Clock Drift

4.2.1 Crystal oscillator basics

The control unit of any sensor node can be assumed to have as its clock source a crystal oscillator. A very simplified representation of a crystal oscillator is shown in Figure 4.2 below.

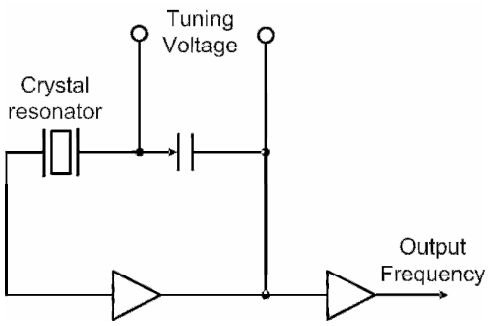

Figure 4.2. Simplified structure of a crystal oscillator.

The crystal resonator is a quartz crystal. Quartz possesses the useful property of being *piezo-electric*. What this means is that when mechanical stress is applied to a quartz crystal, this causes a voltage across the crystal. Conversely, if a voltage is applied across the crystal, this will cause mechanical movement within the crystal. When an AC voltage is applied across the crystal, it will vibrate. Based on such factors as its cut, size and shape, each crystal will have a certain frequency at which it vibrates best, which is the crystal's *resonant* frequency. When used in the oscillator circuit as shown above, the whole oscillator will begin oscillating at the crystal's resonant frequency. One of the reasons why

quartz is used as material for the crystals is simply because it has a very narrow resonant frequency range. This is mostly referred to as quartz having a high quality factor *Q*. This high *Q* value makes the quartz crystal the primary frequency and frequency stability determining element in the oscillator [68], [69], [70].

4.2.2 Frequency accuracy vs. stability

So far in this dissertation, the quality of a clock has been described with a single value Θ and the term *clock drift* has been used to describe the clock errors that can occur between two sensor nodes. However, the quality of a clock in fact consists of two separate concepts, namely *accuracy* and *stability* of the clock. These two terms are best explained with the use of Figure 4.3 below.

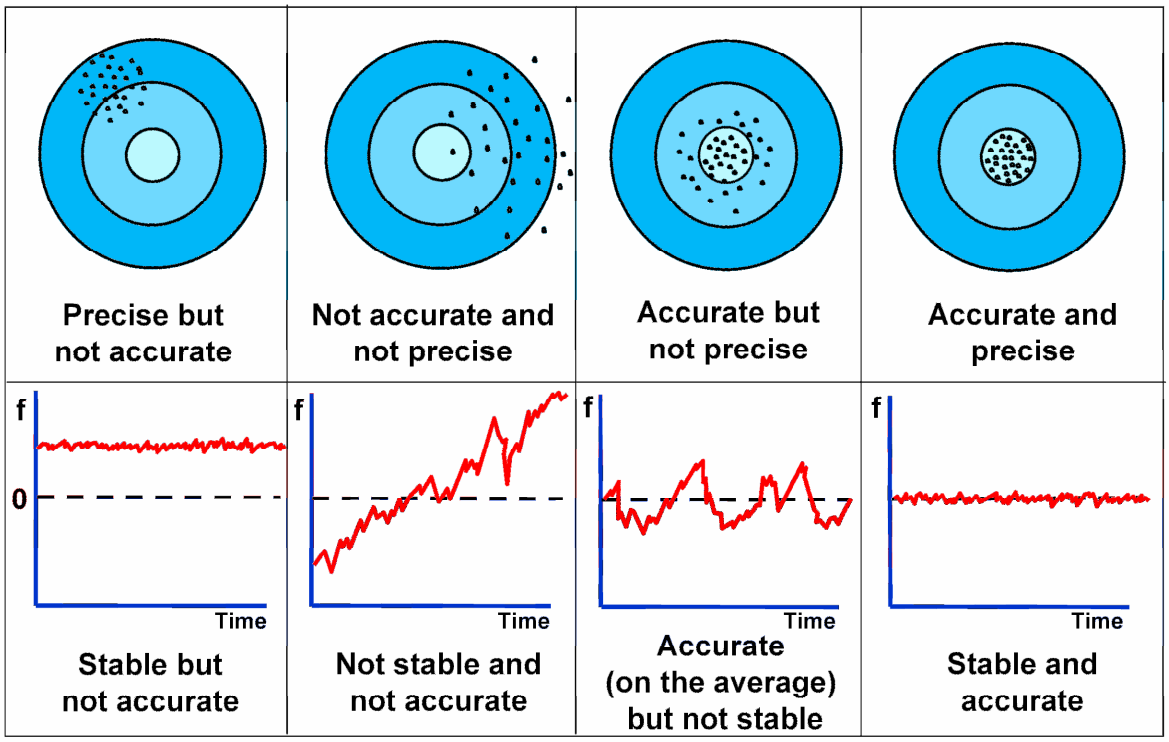

Figure 4.3. An explanation of accuracy vs. stability for clock sources. Taken from [70] (p. 4-2).

The bottom part of Figure 4.3 shows the relation of the terms accuracy and stability to the frequency output of a crystal oscillator.

The degree to which a crystal oscillator is accurate is the degree to which the average frequency at which it oscillates is close to its specified resonant frequency. Thus any

frequency *inaccuracy* can be seen as a fixed average offset of an oscillator's frequency output from the specified value. Say a microcontroller's oscillator is rated at 26MHz, but by actual measurements it is found to have an average frequency of 26.01MHz, then this average offset of 0.01MHz is the degree to which the oscillator is inaccurate. Differences in accuracy are mainly attributed to differences in the exact shapes and sizes of quartz crystals.

On the other hand, the degree to which an oscillator is stable is the degree to which the frequency output of the oscillator varies from its average frequency value. Using the same example as above, if at any two separate points in time the oscillator with an average frequency output of 26.01MHz is found to have an output of first 26.009MHz and then 26.011MHz, then the variability of ±0.001MHz around its average value is the degree to which the oscillator is unstable. Causes of instabilities in oscillators can be grouped as long-term and short-term processes. Long-term processes include component aging and gradual temperature changes. In general, such long-term processes will affect all sensor nodes in a given geographical area in the same way. However, even if such an assumption is not valid, the small gradual changes to clock accuracy which they produce are easily dealt with. After the design of the new MAC protocol has been discussed, it will become apparent why the long-term processes are not critical with regards to sensor node synchronization. At this stage, short-term instabilities will be investigated in some more detail.

4.2.3 Short-term clock instabilities

The main short-term clock instability process is noise in crystals and oscillator circuitry. The inherent unpredictability of short-term instabilities makes them especially critical in developing a method of clock synchronization and thus, quantification of such instabilities is necessary.

Intuitively, the procedure to evaluate a clock's quality would be to repetitively measure the frequency of the clock and then analyze the data. The average value of all the measurements would indicate the accuracy of the clock (fixed offset) and the standard deviation would indicate the stability of the clock (short term fluctuations). There is a problem with this approach though, since clock frequency cannot be measured instantaneously, but needs an averaging interval τ in which to count the number of clock

cycles so as to calculate the frequency for that interval τ. It turns out that the standard deviation is ill-defined for such a set of measurements as it shows dependency on the length of interval τ [71]. Furthermore, in certain situations, the standard deviation value can increase without limit as the number of data points is increased [72]. To solve this problem, different measures have been proposed to characterize clock stability. The specification which has been recommended by the IEEE to characterize clock stability in the time domain is referred to as the *Allan deviation* which is the square root of the Allan variance. It is denoted by $\sigma_y(\tau)$. A detailed explanation of why this measure is needed and what it represents can be found in [72]. Its mathematical definition is given by

$$\sigma_y(\tau)=\sqrt{\frac{1}{2}\langle(y_{k+1}-y_k)^2\rangle}\,, \tag{4.1}$$

where $y=\frac{\Delta f}{f_n}$ are the fractional frequency errors from nominal frequency f_n, which are measured over a time interval τ, and $(y_{k+1} - y_k)$ are the differences between pairs of successive measurements. Lastly <> represents a time average of an infinite number of $(y_{k+1} - y_k)^2$, however, a good estimate can be obtained from 100 or more measurements [70]. A typical plot of the Allan deviation is given in Figure 4.4 below.

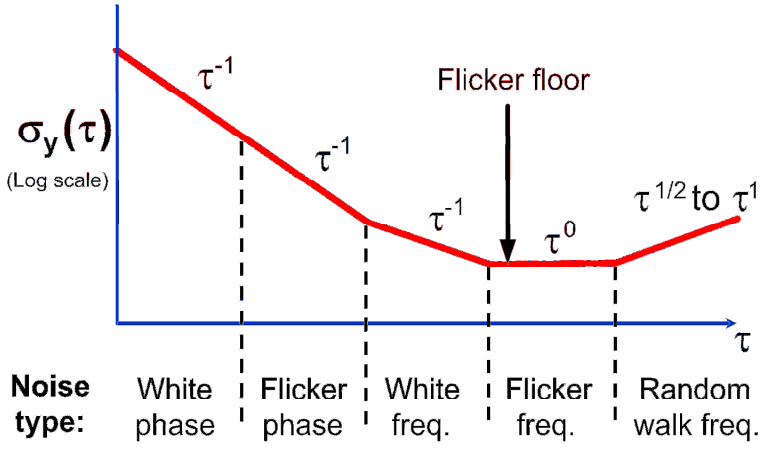

Figure 4.4. A typical Allan deviation plot. Taken from [70] (p. 4-26).

It can be seen that for small τ values, white phase, flicker phase and white frequency noise dominate clock instability. As τ increases, some of these high frequency noise types average out and instability decreases up to a certain point. This point is referred to as the flicker floor. After this point, as τ increases, longer term processes such as random frequency walk and eventually component aging, dominate the frequency instability.

The reason why the Allan deviation is such a useful measure for determining clock stability is because it is straightforward to convert the Allan deviation to the timing errors that can be expected from an oscillator. This is achieved as follows. If a clock is perfectly synchronized and set at a point in time t_0, then τ seconds later, the standard deviation of the expected time error is simply calculated as $\sigma_y(\tau)\times\tau$ [71]. In other words, the two coordinates of a point on the $\sigma_y(\tau)$ plot are multiplied together and this gives the remaining time error, caused by instabilities, that can be expected once the average time error, caused by a fixed frequency offset, has been removed. A plot of $\sigma_y(\tau)$ for various clock sources is shown in Figure 4.5 below.

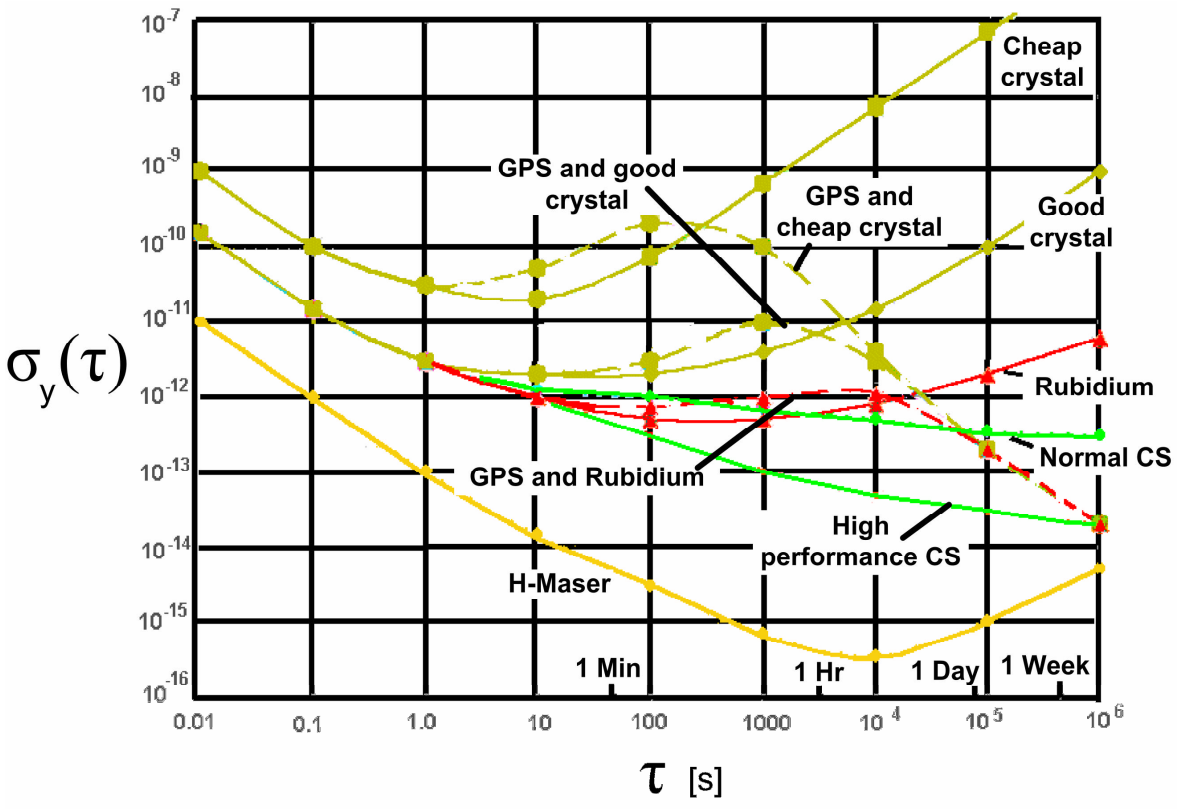

Figure 4.5. Allan deviation plots for various clock sources. Taken from [71] (p. 13, Fig. 10).

The top curve in Figure 4.5 above is for cheap crystal oscillators. This is the information of interest for WSN sensor nodes. To make the information of the above graph more usable, it is converted to the expected time error standard deviation using the method discussed in the previous paragraph. The resulting plot is shown in Figure 4.6.

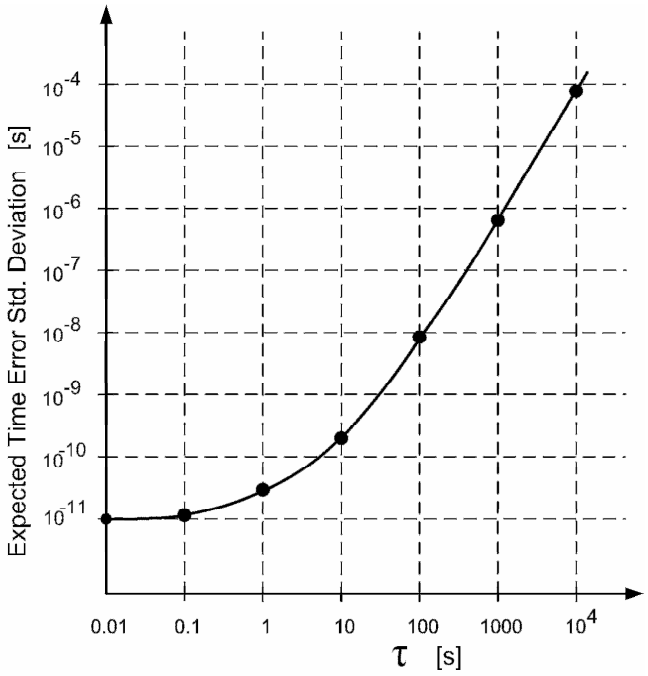

Figure 4.6. Time error standard deviation vs. synchronization interval τ for cheap crystals.

The plot in Figure 4.6 was obtained by simply taking the "cheap crystal" curve from Figure 4.5 and all along the plot multiplying the x- and y-coordinates to get the time error standard deviation for each value of the synchronization interval τ. Having derived at a quantifiable measure of the short term instabilities in a cheap crystal oscillator as shown in Figure 4.6, some important conclusions can be drawn in the next section.

4.2.4 Applicable conclusions

The most important finding that can be deduced from the previous sections is explained with an example. Suppose a microcontroller's oscillator uses a certain quartz crystal which has a frequency tolerance specified at Θ. Suppose the microcontroller implements a clock function that attempts to keep track of time. At time t_0 the microcontroller's time is synchronized perfectly to absolute time. The timing error that can be expected from this clock after τ seconds in fact consists of two main parts.

$$\Delta t_{error} = \underbrace{(\theta_{ave} \times \tau)}_{\substack{stable \\ (frequency\ offset)}} + \underbrace{\delta_{noise}}_{\substack{random \\ (noise)}} \quad (4.2)$$

where Δt_{error} is the time error at the end of time τ, θ_{ave} is the average difference between the crystal's nominal and actual frequency, τ is the time since synchronization and δ_{noise} is a random variable with zero mean and standard deviation = $\sigma_y(\tau)\times\tau$.

To demonstrate the above formula, suppose τ = 100 seconds and θ_{ave} = +20ppm. From Figure 4.6 it is seen that for τ = 100s, $\delta_{noise} \sim \sigma_y(\tau)\times\tau$ = ±9x10^{-9}, therefore

$$\begin{aligned}\Delta t_{error} &= (\theta_{ave}\times L) + \delta_{noise}\\ &= \left(20\times10^{-6}\times100\right) \pm 9\times10^{-9}\\ &= 0.002 \pm 0.000000009\\ &\approx 2\times10^{-3}\ s\end{aligned}$$

From this it is seen that by far the dominating term of the expected time error is that which is caused by the average frequency offset. Since such an average offset can be compensated for, this will be exploited in the design of a new WSN MAC protocol.

4.3 DPS-MAC

In this section, a new MAC protocol for Wireless Sensor Networks is proposed. It is referred to as Dynamic Preamble Sampling MAC (DPS-MAC) and is based on the CSMA-MPS protocol. DPS-MAC can be termed a low duty cycle MAC protocol as it lets nodes remain in a low power sleep state for most of their life. DPS-MAC does not require network-wide synchronization, but to the degree that it keeps track of neighbours' listening schedules, it synchronizes their communications so as to be as frugal as possible with each node's energy reserves. DPS-MAC will be described in detail in this section.

4.3.1 Basic protocol operation

Each node in a network running the DPS-MAC protocol has a periodic wakeup cycle. Every T_w seconds, a node's radio is woken up to sample the medium for possible incoming data. The time it takes to wake up the radio is denoted by T_{wu} and it is dominated by the transceiver's turn on time. Once the radio is operational, the listening slot lasts for T_{pl} seconds, which is as short as possible so as not to waste energy. If during T_{pl} the node determines that the medium is idle and there is thus nothing to receive, it goes back to

sleep and the same procedure takes place again in T_w seconds. The length of T_w is fixed for any particular network, i.e. T_w can be set to any desired value before network deployment, but once deployed, it is fixed and equal for every node in the network.

4.3.1.1. Protocol state 1 – UNSYNCHRONIZED

If a node *A* wants to forward a data packet to one of its neighbours *B* and it does not have an estimate of when that neighbour will start its next listening slot, then DPS-MAC will operate in the UNSYNCHRONIZED state as shown in Figure 4.7 below.

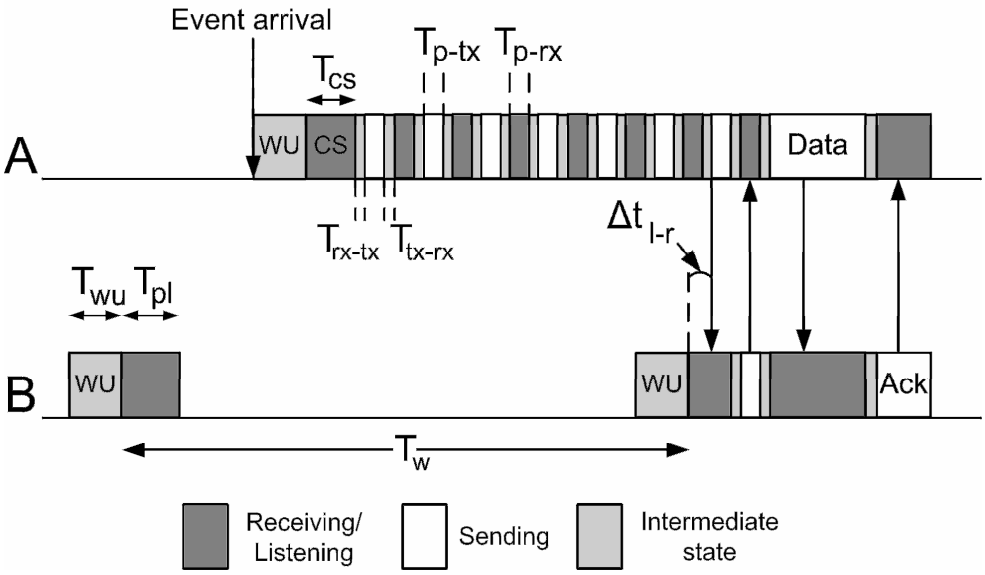

Figure 4.7. DPS-MAC operation in state UNSYNCHRONIZED.

In the UNSYNCHRONIZED state, the sending process is started immediately upon packet/event arrival. First, the radio is started up. After T_{wu} seconds, the radio is ready and a Carrier Sense (CS) operation is started which lasts for T_{cs} seconds. If *A* finds the medium idle during CS, it starts sending preamble packets immediately, and is prepared to send preambles for up to T_w seconds in the worst case (i.e. if *B*'s previous listen slot ended just as the first preamble packet was sent, then preambles will have to be sent for approximately T_w seconds). The preamble in fact consists of alternating Transmit (T) and Receive (R) slots. After every preamble packet, *A* listens for a short time to determine whether *B* has been reached yet. As long as nothing is received during these short listen slots, *A* assumes that *B* is not awake yet and carries on sending. On the receiver's side, when during its periodic listen time *B* starts receiving a preamble packet, it determines whether it is the intended recipient of the packet. If not, it goes back to sleep. If yes, it replies with a short preamble ACK packet to tell *A* that it can stop sending preambles. *B*

then waits for the ensuing data packet. Once it has arrived, a final ACK packet is sent *A* to acknowledge successful receipt of the data packet.

It can be seen in Figure 4.7 that the preamble and preamble ACK packets are kept as short as possible to speed up the neighbour wake-up process as well as to enable the CS and periodic listening times to be as short as possible so as to save valuable energy. The preamble transmit and receive slots last $T_{p\text{-}tx}$ and $T_{p\text{-}rx}$ seconds respectively. Furthermore, the radio's turn-around times between transmit and receive as well as receive and transmit states are $T_{tx\text{-}rx}$ and $T_{rx\text{-}tx}$ respectively. These depend on the radio hardware but should be as short as possible

Furthermore, the value $\Delta t_{l\text{-}r}$ seen in the figure is the difference in time between when *B*'s listening slot starts and when it actually starts receiving a valid preamble packet. This time is measured by node *B* and sent back to *A* in the preamble ACK packet. It enables *A* to know the exact point in time when *B* started its listen slot (i.e. $\Delta t_{l\text{-}r}$ seconds before *A* started sending). At this point in time, *A* has an estimate of *B*'s wakeup schedule and thus the communication of the two nodes moves to the GOT_SLOT_ESTIMATE state.

4.3.1.2. Protocol state 2 – GOT_SLOT_ESTIMATE

In the GOT_SLOT_ESTIMATE state, if *A* again has a data packet to forward to *B*, the following procedure is followed. *A* has an estimate of when node *B* will wake up and furthermore, the maximum frequency tolerance Θ of the nodes' oscillators is also known from the hardware specifications. Thus, as described in section 3.4.1.7 for WiseMAC, the packet is sent just at the right time to take into account the worst case clock drift. This process is shown in Figure 4.8 below.

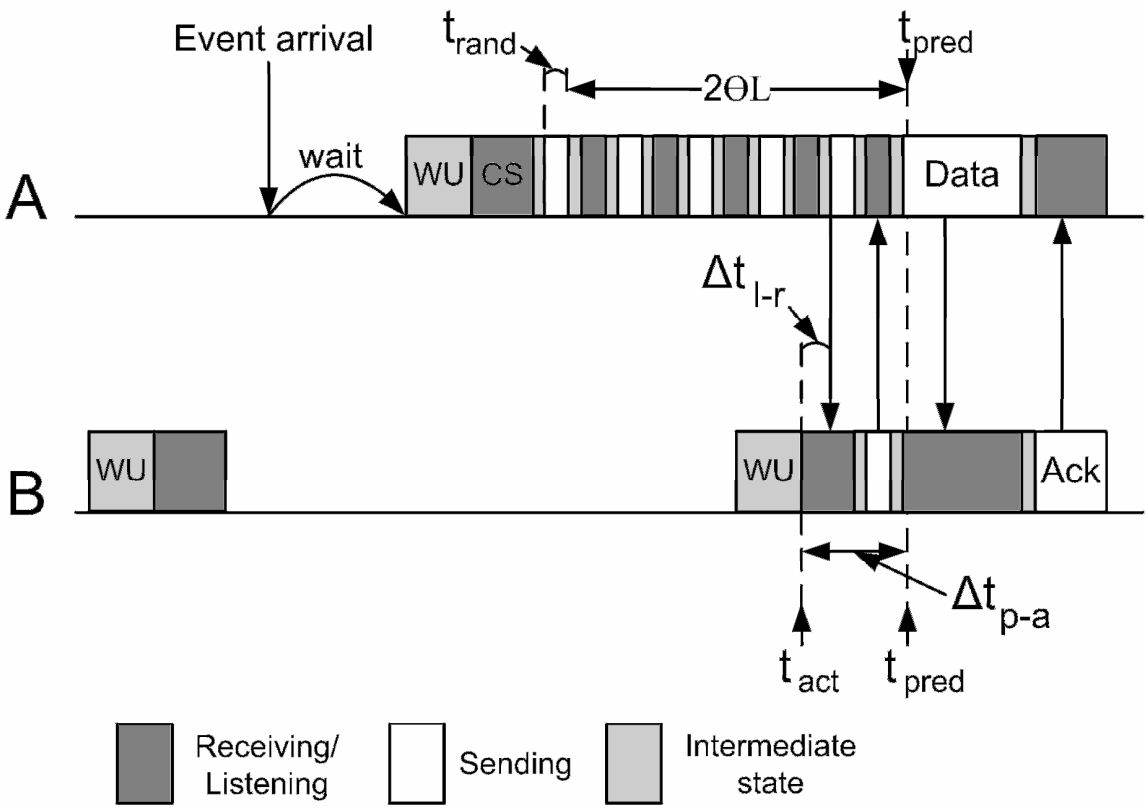

Figure 4.8. DPS-MAC operation in state GOT_SLOT_ESTIMATE.

As is seen in the above figure, node *A* predicts that node *B* will listen at time t_{pred}. The time between t_{pred} and the last time node *A* sent a packet to *B* is L. Thus, node *A* must start sending preamble packets 2ΘL seconds before its prediction. On top of this, to prevent nodes that are similarly well synchronized to node *B* from starting transmission at the same time as *A*, the communication is padded with a small random time t_{rand}. As explained in [1], collisions can occur when two nodes try to send data to a single destination during the same listen slot and their estimates of the destination's slot are accurate to within [$-T_{rx-tx}$, $+T_{rx-tx}$] seconds of each other. As with CSMA-MPS, the magnitude of the t_{rand} value is set to k x T_{rx-tx}, where k is a uniform random integer from the interval [0, N], N being the number of neighbours in the node's neighbour table. This causes the range of possible t_{rand} values to increase as the network becomes denser, which is desirable, because in a denser network, collisions are more probable, and thus a greater range for t_{rand} is needed to reduce collision probabilities to an acceptable level. Node *A* thus starts transmitting preambles for a maximum time of 4ΘL + t_{rand} as shown. Within this time, *A* is guaranteed to hit *B*'s listen slot. As soon as *A* receives a preamble ACK from *B*, it continues with the data as in the previous state. Again node *B* measures Δt_{l-r} and sends this value in the preamble ACK

packet. Thus, *A* calculates t_{act}, the time when *B* started listening and saves it as $t_{lastcomm}$ in its neighbour table entry for *B*, the time when *A* and *B* last communicated successfully.

4.3.1.3. Protocol state 3 – GOT_DRIFT_ESTIMATE

From this point onward, the new protocol, DPS-MAC, carries on where up to now, the functionality of the MAC protocols, specifically CSMA-MPS has ended. DPS-MAC adds a third state of synchronization, namely GOT_DRIFT_ESTIMATE. The way this state is reached is as follows. From the communication that took place in the previous state, node *A* knows the time when it *predicted* node *B* would start to listen, t_{pred}, and furthermore, it also knows the time when node *B actually* started to listen, t_{act}. *A* calculates the difference between these two values, $\Delta t_{p\text{-}a} = t_{act} - t_{pred}$ as shown in Figure 4.8. From this, node *A* can calculate the actual clock drift Θ_{actAB} that occurred during time L between *A* and *B* as

$$\theta_{actAB} = \frac{\Delta t_{p-a}}{L}. \tag{4.3}$$

Since it has been shown in section 4.2 that the biggest part of Θ_{actAB} is in fact a stable value (two average frequency offsets in *A* and *B*'s crystals), Θ_{actAB} will remain stable enough so as to accurately predict node *B*'s next listen slot. From here on forward, node *A* can predict the start of node *B*'s listen slot as

$$t_{pred} = t_{lastcomm} + (n \times T_w) + (\theta_{actAB} \times (n \times T_w)) \tag{4.4}$$

where $t_{lastcomm}$ is the time at which *A* last communicated successfully with *B*, and *n* is the number of listen periods T_w which need to be added to $t_{lastcomm}$ to estimate the start of *B*'s next listen slot (not taking drift into account). In other words, node *A* starts with $t_{lastcomm}$ and repetitively adds T_w until a value greater than the current time is found. This is the predicted next listening time of *B* without clock drift compensation. Then *A* adds the last term in Equation 4.4 to take into account the drift and thus has a very accurate estimate of *B*'s listen slot start. As was shown in section 4.2.4, the uncertainty in each node's clock drift is on the order of $\sigma_y(\tau) \times \tau$ seconds. Thus, since Θ_{actAB} incorporates *both* node *A* and node *B*'s clock drift, the predictions made using Θ_{actAB} will have an error on the order of $2(\sigma_y(\tau) \times \tau)$. From Figure 4.6 it can be seen that up to a synchronization interval of τ = 1000 seconds, $2(\sigma_y(\tau) \times \tau)$ will be less than 2×10^{-6} seconds. What this means is that even if nodes *A* and *B* last communicated 1000 seconds (approximately 16 minutes) ago,

node *A* can predict the start of node *B*'s next listen slot using Equation 4.4 and only be in error by approximately 2µs.

The communication in state GOT_DRIFT_ESTIMATE takes place as shown in Figure 4.9 below.

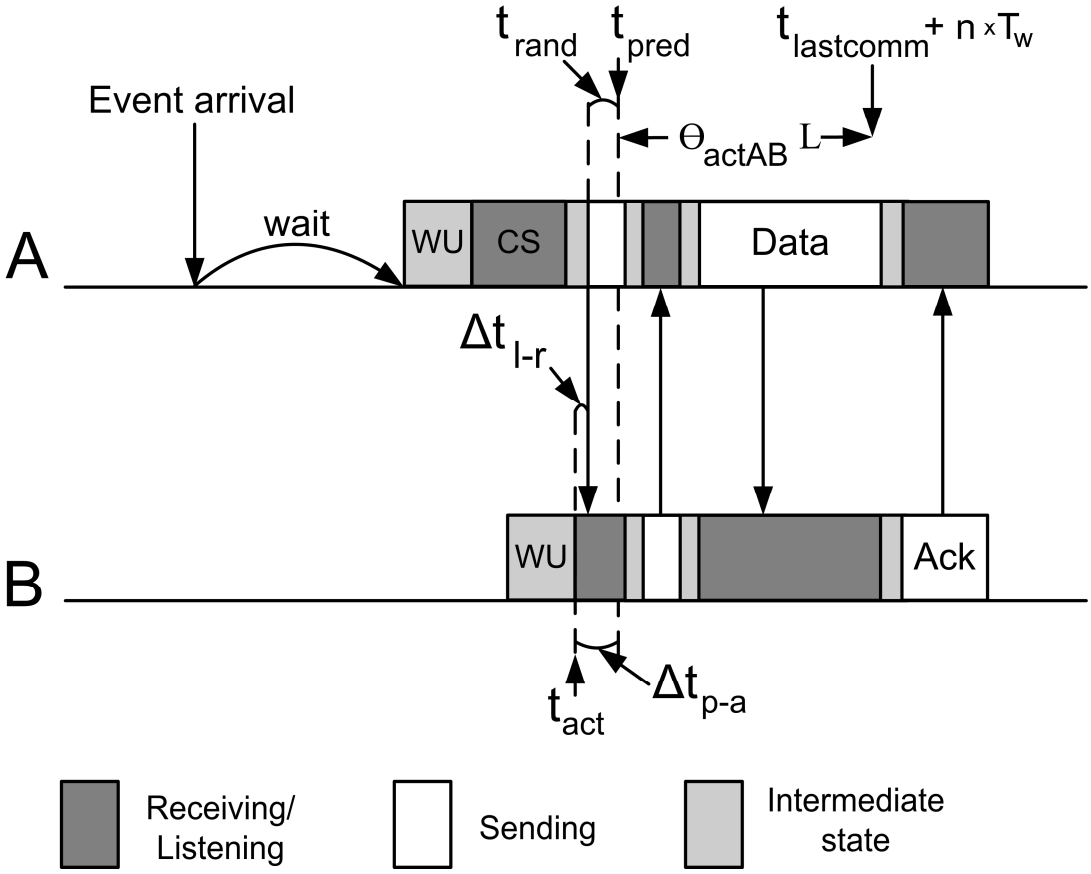

Figure 4.9. DPS-MAC operation in state GOT_DRIFT_ESTIMATE.

In the figure above, the value of the clock drift between *A* and *B*, Θ_{actAB}, is negative, thus t_{pred} occurs earlier than the time that would have usually been predicted ($t_{lastcomm}$ +nT_w). Furthermore, a small random time t_{rand} is again introduced so as to reduce the chances of collisions. The magnitude of t_{rand} is the same as in the GOT_SLOT_ESTIMATE state.

In Figure 4.9 it is seen that *A* can again calculate the difference between *B*'s actual listen slot start and the predicted listen slot start as $\Delta t_{p\text{-}a} = t_{act} - t_{pred}$. Since *A* already has a drift estimate for *B*, the new value of $\Delta t_{p\text{-}a}$ includes the instability of the quartz crystals on the

one hand, but also any long term changes in *A* and *B*'s clock drift. Thus it cannot be ignored. What node *A* does is to update its clock drift estimate for node *B* as follows.

$$\theta_{actAB} = \theta_{actAB} + \frac{\left(\frac{\Delta t_{p-a}}{L}\right)}{2} \tag{4.5}$$

Using Equation 4.5 essentially sets the new value for Θ_{actAB} as the average of its current value and the new value. This averaging, to a certain degree, prevents rare sudden noise bursts in *A* or *B*'s oscillators from affecting the value of Θ_{actAB} too much. Equation 4.5 should *not* be confused with Equation 4.4. Equation 4.4 is used when no estimate of the clock drift is available, whereas Equation 4.5 is only used to update an already existent clock drift estimate.

4.3.2 Long-term clock instability

As was discussed in the previous section, DPS-MAC continuously updates the value of the clock drift experienced relative to each neighbour. Each time a packet is sent to a neighbour, the value of the clock drift is adjusted. As such, any changes to the actual drifts of the nodes' oscillators caused by long-term oscillator instabilities are easily dealt with. Such long-term processes are mostly caused by oscillator component aging or gradual temperature changes.

It should be kept in mind that up to a synchronization interval of τ = 1000 seconds, the uncertainty of the neighbour listen slot predictions will be on the order of a few micro seconds, which is tolerable and will not lead to loss of synchronization. However, if the synchronization interval grows bigger than 1000 seconds, as seen in Figure 4.6, the uncertainty of the predictions will keep on growing and become intolerable. For example, if a node is using a radio with a 1Mbps bit rate, then one bit is the equivalent of 1µs. If the prediction uncertainty grows beyond approximately 100µs, then a neighbour's listen slot could be missed by as much as 100bits, which could be the size of a whole preamble packet. This could lead to loss of synchronization and resynchronization would be necessary, causing energy expenditure. There are two ways to deal with this problem in the case where a WSN has so little traffic that the time between node communications becomes larger than 1000 seconds.

- If DPS-MAC has a packet to send to a certain destination and notes that the time since last communication is more than 1000 seconds, it proceeds by starting the preamble sending by *X* seconds earlier than predicted. The value of *X* becomes larger as the time since the last communication increases, exactly as seen in Figure 4.6, and can be implemented as a simple lookup table.
- DPS-MAC can be forced to send a single short synchronization preamble packet every 1000 seconds to force a synchronization update.

Both of these methods cause negligible protocol overhead. However, it should be pointed out that the first method requires that temperature variations or other factors not cause the long-term clock instability to be of greater magnitude than shown in Figure 4.6. In general, the second approach would be more robust to uncertainties in the long-term clock instabilities and it is therefore the preferred approach.

4.3.3 Neighbour information

In order for the DPS-MAC protocol to function correctly, the protocol needs to maintain a database containing the information of all its neighbours. This database is referred to as the neighbour table. In fact, DPS-MAC can share this database with the routing protocol since both need to make use of such a structure. For DPS-MAC, every neighbour entry in the neighbour table contains the following information fields.

Name	Description
Neighbour address	16 bit unique address.
Actual clock drift	The value of the clock drift calculated for this neighbour.
Neighbour state	The synchronization state of the neighbour.
Last communication time	The time since the last communication/synchronization.
Missed communications	The number of times the neighbour could not be reached. If this value becomes too large, the neighbour is considered disconnected.

Table 4.2. DPS-MAC neighbour table entries.

4.3.4 Additional procedures

In this section, some additional procedures which DPS-MAC performs are explained.

4.3.4.1. Neighbour discovery

After a node switched is on, it immediately starts listening in periodic listen slots. The start of the first listen slot is chosen at random so as to distribute all the nodes' listen schedules over time. Then, for a user-defined period of time, $T_{n\text{-}disc}$, the node does nothing else but to wait to receive a neighbour discovery broadcast packet.

Case 1:

If a neighbour discovery packet is not received within the time $T_{n\text{-}disc}$, the node starts sending its own neighbour discovery broadcast. Such a broadcast takes the form of a train of neighbour discovery packets. This broadcast packet train is much like a preamble packet train; i.e. it consists of alternating transmit and receive slots. In the transmit slot, a neighbour discovery packet is sent. This packet contains mainly the sender's address. In the receive slot, a neighbour discovery ACK packet is expected. The ACK packet consists mainly of the responding node's address. The neighbour discovery train lasts for at least T_w seconds so as to be sure to reach all neighbours. For every neighbour discovery ACK which is received in this time, a new neighbour entry is created in the neighbour table.

Case 2:

If a neighbour discovery packet is received within $T_{n\text{-}disc}$, the node replies to the sender with a short neighbour discovery ACK packet and then it adds the sender to the neighbour table as the first entry. After a short random delay, the node starts its own neighbour discovery process as discussed above.

If no neighbours are found using the process above, a node will simply restart the process, i.e. wait for the set time for incoming neighbour discovery packets (say for 30 minutes) and if nothing is received, start its own discovery process.

In simple data gathering WSNs, the above procedure can be used by letting the sink (base station) node initiate the neighbour discovery process, which will then spread from the sink outwards in a more or less ordered manner until all nodes have populated their neighbour tables.

4.3.4.2. Neighbour downgrading

Neighbour downgrading is when a neighbour's synchronization state is set to a lower state of synchronization. This occurs when a node attempts to reach one of its neighbours but this attempt is unsuccessful, which is termed a missed communication. A missed communication is detected when the maximum number of preamble packets (*maxNoOfPreambles*) has been sent but still no preamble ACK packet has been received. The process is shown in Figure 4.10 below.

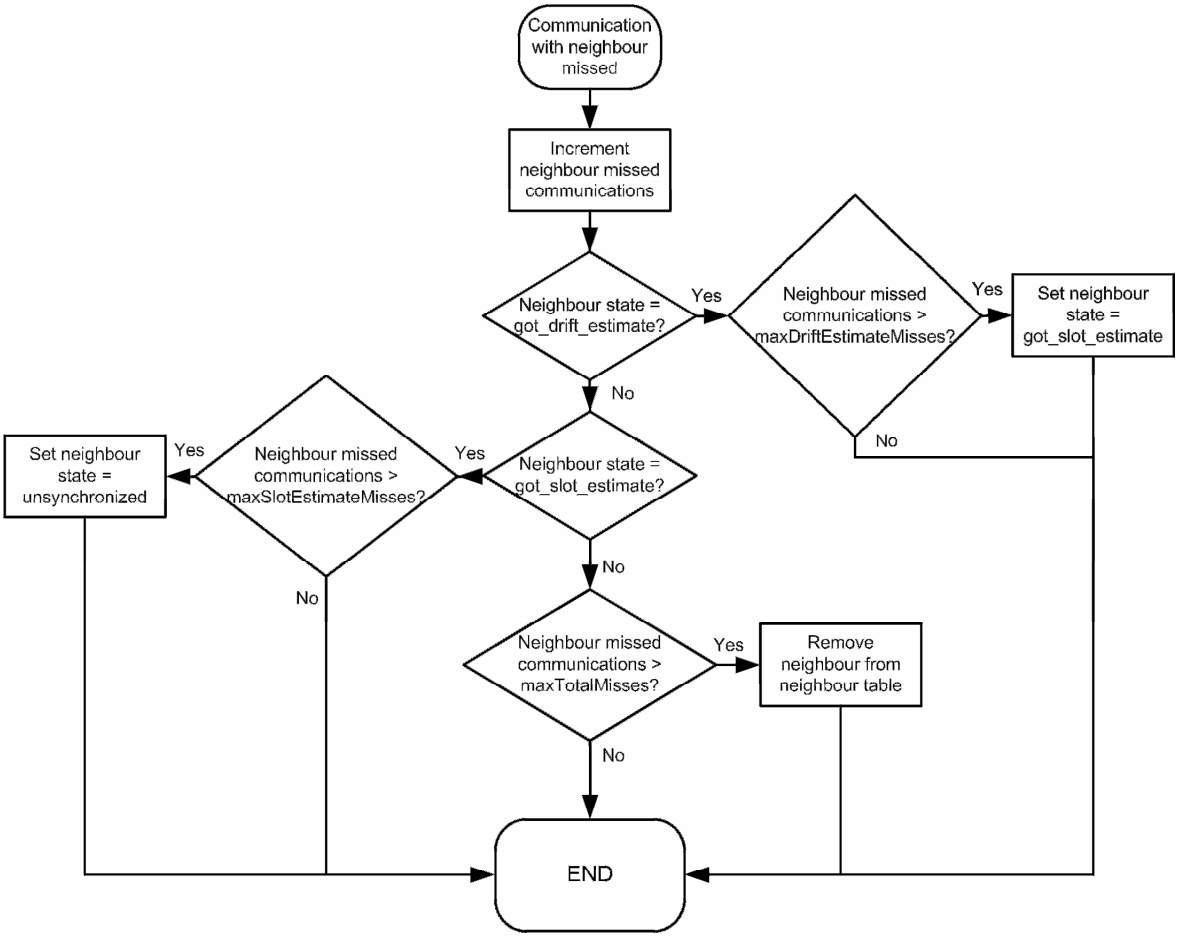

Figure 4.10. The neighbour downgrading process of DPS-MAC.

The values *maxDriftEstimateMisses*, *maxSlotEstimateMisses* and *maxTotalMisses* in the above figure are all user definable variables of the DPS-MAC protocol.

4.3.4.3. Back-off procedures

Back-off occurs firstly when a carrier sense operation indicates a busy medium and secondly when a communication with a neighbour is missed.

If the medium is found busy by a carrier sense operation, two cases are possible. In case the transmission is a neighbour discovery broadcast or a transmission to an unsynchronized neighbour, such transmissions need not be started at a particular point in time and thus the retransmission reattempt is scheduled after a short random time in the interval $[T_w/2, T_w]$. This is chosen so as to not reattempt a retransmission immediately, but also not to wait unnecessarily long before trying again. On the other hand, if the transmission is to a synchronized neighbour, the retransmission will simply be scheduled for the destination's next listen slot.

In the case of a neighbour communication being missed as described in the previous section, any rescheduling is only allowed after a certain back-off time. The back-off time in this case is set to $k \times T_w$, where k is a uniform random integer from the interval [0, N], N being the number of neighbouring nodes. Thus, when a communication with a neighbour is missed, the back-off time is longer than when the medium is found busy. The reason for this is that if a neighbour is missed, then either, this neighbour is experiencing some type of problems, or the hidden-terminal scenario is causing collisions. In both of these cases it is advisable to stop transmissions for a certain period of time in order to not cause further problems. Furthermore, the reason for increasing the range of the back-off time as the number of neighbours increases, is that the more neighbours there are, the more likely it is for congestion to occur, the more it is necessary to stagger the communications among the neighbours. Such staggering of communications is achieved by having longer back-off times.

4.3.4.4. Packet buffering

DPS-MAC utilizes a packet buffer which is used to queue up packets waiting to be sent out over the radio. The exact way in which this buffer is used depends on the WSN application type. In a very simple application, all normal data packets have the sink as their ultimate destination, i.e. data never flows from the sink to a single node. In this case DPS-MAC will add any data packet that is received straight to the outgoing packet buffer and schedule the forwarding of the packet to the next downstream neighbour. In a complex application on the other hand, DPS-MAC will simply pass every received packet up to the network layer which will then do with the packet whatever it wants. As discussed in section 3.4.5, the degree of integration of the protocol layers depends on the application.

In either of the cases above though, DPS-MAC provides a negative ACK packet (N_ACK) in case a full buffer is encountered. Suppose during node *A*'s listen slot a preamble is received from node *B* which indicates that a data packet is following. If *A* will need to continue to forward this packet downstream, but *A*'s outgoing packet buffer is already full, there are three options. Firstly, *A* can reply with a preamble ACK packet, receive the data as normal and then simply drop it. This will cause spurious data losses in the network which is not a good solution. Secondly, *A* can simply remain silent and not send an ACK packet back to *B*. In this case, *B* will continue sending preambles for up to T_w seconds, which will waste precious energy reserves. Lastly, *A* can respond with an N_ACK packet and thus let *B* know that its buffer is full. In this case, *B* can stop sending preamble packets and no spurious data losses will occur. The last method is therefore the most appropriate solution and is chosen for the DPS-MAC protocol.

4.3.4.5. Listen slot hijacking

Listen slot hijacking needs to occur when two neighbouring nodes have listen slots that are sufficiently close to each other so as to interfere with smooth communication among the two nodes. Suppose *A* has a data packet to forward to its only neighbour *B*, but every time *A* is supposed to wake up and send the packet to *B* it is already awake and busy with its own listen slot. This would mean that packets would start accumulating in *A*'s packet buffer and eventually packets would be dropped. In such a case, listen slot hijacking permits node *A* to abandon its ongoing periodic listen slot activity in favour of forwarding the packet to *B*. In essence, *A* has hijacked its own listen slot so as to rather send during this time.

4.3.4.6. Power control

Power control can optionally be implemented by adjusting the output power of the radio depending on how far away a neighbour is located. The idea is that if a neighbour is very close, then packets to that neighbour need not be sent at full power. However, as stated in [46], in practical radio implementations, the majority of the power in GHz frequency transmitters is used just to run the frequency synthesizers and other circuitry, regardless of output power. Therefore, although reducing output power can reduce energy consumption for a single transmission, this energy saving is not in direct proportion to the reduction in transmission distance. Having a shorter transmission distance means that more hops will

need to be traversed on route to the destination. The authors of [46] found that the application of a power control scheme to a WSN yielded negligible overall energy savings. The same observation is confirmed in [36] for the WiseMAC protocol.

4.3.5 Packet formats

In this section, the DPS-MAC packet formats will be discussed. They are based on the CSMA-MPS packet format [1] and aim for minimal size so as to reduce power consumption.

Every DPS-MAC packet is encapsulated in a physical layer frame header. The resulting frame represents the actual bits that are sent out over the radio interface. The frame format is shown in Figure 4.11 below.

Preamble (101010101…)	Sync Word	DPS-MAC packet	CRC
32 bits	16 bits	variable	16 bits

Figure 4.11. The format for physical layer frames.

The fields in the above figure are explained as follows.

- *Preamble*: repetitive 1's and 0's to bit synchronize the sending and receiving radios.
- *Sync Word*: specific pattern of bits to indicate the start of the actual information contained in the frame.
- *DPS-MAC packet*: the actual packet from the MAC protocol. Its length is different for each packet type.
- *Cyclic Redundancy Check (CRC)*: implements error detection.

In the CC2400 and CC2500 radios, the Preamble, Sync Word and CRC are all automatically added to the data packet.

The format for a normal preamble packet is shown in Figure 4.12 below.

Control	TxPower	Source address	Destination address
4 bits	4 bits	16 bits	16 bits

Figure 4.12. The format of a DPS-MAC preamble packet.

- *Control*: indicates the type of DPS-MAC packet.
- *TxPower*: indicates the transmit power that was used to send the packet. This can be used to implement power control.
- *Source address*: indicates the address of the sender of the packet.
- *Destination address*: indicates the address of the intended recipient.

The format of a preamble ACK, N_ACK or final ACK packet is shown in Figure 4.13 below.

Control	RSSI	Clock Offset
4 bits	4 bits	16 bits

Figure 4.13. The format of a DPS-MAC preamble ACK, N_ACK or final ACK packet.

The fields in the above figure are explained as follows. Only those fields that have not been encountered in previous packets will be explained for the remaining packets.

- *RSSI*: indicates the received signal strength with which the preceding preamble or data packet was received. This lets the sender of the preamble know how much signal strength was lost during transmission and thus gives an indication of the distance to the recipient. The RSSI field can be used for a power control scheme.
- *Clock Offset*: is the difference in time in micro seconds between when the node started listening and when it started receiving ($\Delta t_{l\text{-}r}$ in Figures 4.7, 4.8 and 4.9).

The packet format of a data packet is given in Figure 4.14 below.

Control	TxPower	Length	Source address	Destination address	Data
4 bits	4 bits	8 bits	16 bits	16 bits	variable

Figure 4.14. The format of a DPS-MAC data packet.

- *Length*: indicates the length of the packet. This is necessary since data packets can have variable length and thus, to assist the radio in knowing when to stop receiving, a length byte is added.
- *Data*: the actual data payload received from upper protocol layers.

The neighbour discovery preamble packet format is shown in Figure 4.15 below.

Control	TxPower	Source address	Hop Count (optional)
4 bits	4 bits	16 bits	16 bits

Figure 4.15. The format of a DPS-MAC neighbour discovery preamble packet.

- *Hop count*: optional field. For simple applications, the neighbour discovery process can simultaneously let each node learn its distance (hop count) from the sink. In complex applications, the routing layer will take care of this process.

Finally, the neighbour discovery ACK packet format is shown in Figure 4.16 below.

Control	RSSI	Source address	Clock Offset	Hop Count (optional)
4 bits	4 bits	16 bits	16 bits	16 bits

Figure 4.16. The format of a DPS-MAC neighbour discovery ACK packet.

The abovementioned packets represent all the packet formats used by DPS-MAC. The formats are as simple and short as possible so as to reduce protocol overhead to a minimum.

4.3.6 Summary of main parameters

The main parameters that have been discussed so far with respect to DPS-MAC are summarized in Table 4.3 I and II below.

Name/Symbol	Description	Value
Θ	Maximum oscillator frequency tolerance.	Determined by the type of quartz crystal used.
T_w	Listen slot interval.	Set at WSN deployment according to expected traffic.
$T_{p\text{-}tx}$	Time to transmit a preamble packet.	Determined by radio bit rate.
$T_{p\text{-}rx}$	Time to receive a preamble ACK packet.	Determined by radio bit rate.

Table 4.3. Part I. Summary of the main DPS-MAC parameters.

Name/Symbol	Description	Value
T_{rx-tx}	Radio receive-transmit turn-around time.	Determined by radio hardware.
T_{tx-rx}	Radio transmit- receive turn-around time.	Determined by radio hardware.
T_{wu}	Time for radio to wake up.	Determined by radio hardware.
T_{cs}	Duration of Carrier Sense operation.	$T_{cs} > T_{p-rx} + T_{rx-tx} + T_{tx-rx}$
T_{pl}	Length of periodic listen slot.	$T_{pl} > T_{p-rx} + T_{rx-tx} + T_{tx-rx}$
t_{rand}	Short random time to reduce collisions.	kT_{rx-tx}, k is a uniform random integer from [0, N], N is the number of neighbours
T_{n-disc}	Time to wait for incoming neighbour discovery packets upon first time node start-up.	Set at WSN deployment.
maxDriftEstimateMisses	The number of neighbour misses at which a node's state will be changed from GOT_DRIFT_ESTIMATE to GOT_SLOT_ESTIMATE	Set at WSN deployment.
maxSlotEstimateMisses	The number of neighbour misses at which a node's state will be changed from GOT_SLOT_ESTIMATE to UNSYNCHRONIZED.	Set at WSN deployment.
maxTotalMisses	The number of neighbour misses at which a node will be removed from the neighbour table.	Set at WSN deployment.
maxNoOfPreambles	The number of preambles that are sent out to try to reach a neighbour. If the neighbour still hasn't responded after this many preambles, a neighbour miss has occurred.	Depends on communicating neighbour's state: state=UNSYNCHRONIZED: T_w/(time per preamble) state= GOT_SLOT_ESTIMATE: 4ΘL/(time per preamble) state= GOT_DRIFT_ESTIMATE: small number set at WSN deployment (typically less than 10) (time per preamble is the time that it takes to send a preamble packet, switch the radio to receive, listen for a preamble ACK and switch radio back to transmit.)
BuffSize	Size of the outgoing packet buffer.	Depends on available memory of the node.

Table 4.3. Part II. Summary of the main DPS-MAC parameters.

4.3.7 Summary of protocol features

As was stated in section 1.4, the DPS-MAC protocol has three main design goals. The most important goal is energy efficiency. Secondary goals are for the protocol to be simple, so as to make it suitable for different hardware and for the protocol to be scalable to differing network sizes. The ways in which the design of DPS-MAC meets these goals are discussed in this section.

4.3.7.1. Energy efficiency

A short summary of DPS-MAC's main energy conservation features is given below.

- DPS-MAC is a preamble sampling protocol. All such protocols make use of the finding that in general, it is more energy efficient to blindly transmit that to blindly receive for the same amount of time [46] as already pointed out in section 3.4.1.15.
- *Idle listening* is reduced by a duty cycled approach. Nodes listen to the channel for very short periodic listen slots and spend the majority of time in a low power sleep state.
- *Overhearing* is reduced by each node having a distinct listen slot, not synchronized to other nodes' listen slots. This means that packets will be sent to different destinations at different times. If a node still receives a preamble packet destined for another node, it will simply go back to sleep and avoid overhearing.
- *Collisions* are reduced by using a listen-before-talk approach. A carrier sense operation is performed before every new transmission to ensure an idle medium
    - *The hidden terminal problem*, which also causes collisions, is reduced by letting the carrier sense range be greater than the range at which actual communication can take place. In other words, the carrier sense threshold at which the channel is deemed busy is lower than the receive sensitivity of the radio by about 10dB (depends on hardware capabilities).
- *Control packet overhead* is minimized by making packet header formats as short and simple as possible and by attempting to shorten preambles as much as possible.
- *Preamble length* is minimized with respect to other protocols by exploiting some oscillator accuracy and stability statistics.

4.3.7.2. Hardware applicability

The DPS-MAC protocol is very simple and does not require complex calculations, algorithms or procedures. It is well suited for implementation on any resource constrained microcontroller platform. In terms of the underlying radio hardware, DPS-MAC attempts to make as few assumptions as possible, however, some radio hardware features are advantageous to its efficient functioning.

- *High bit rate radio*: It has been shown in [39] that high bit rate radios are generally more power efficient than low bit rate radios. Additionally, in DPS-MAC, as the preamble packet transmission time gets shorter, the periodic listen slots can be made shorter and furthermore, the carrier sense duration can be made shorter. All of these factors result in valuable energy resources being saved.
- *Short rx-tx and tx-rx switching times*: Shorter switching times reduce the probability of collision as explained in [38]. Furthermore, short switching times also shorten the duration of the listen and carrier sense slots.
- *Accurate clock*: An accurate clock is required by any protocol that synchronizes neighbouring nodes or the whole network. Most schedule-based protocols as well as WiseMAC, CSMA-MPS and DPS-MAC amongst others fall in this category. Microcontrollers used in WSNs may either operate from a high-speed crystal oscillator or an RC (resistor capacitor) oscillator during normal operation. A high-speed crystal oscillator uses significantly more energy than an RC oscillator and thus, typically and RC oscillator is used, at least when the node enters sleep mode. RC oscillators are much less accurate and stable than crystal oscillators and thus, a low frequency secondary crystal (32.768kHz) running at full duty cycle is used to keep accurate track of time on a node. With a 32.768kHz crystal running at full duty cycle, the Texas Instruments MSP430 consumes only 1-2μW in Low Power Mode 3 (LPM3) [73]. On top of this, the MSP430 provides a sophisticated clock distribution unit [74] which can be used to deterministically switch between low and high frequency clock sources. Such functionality is beneficial in systems requiring low power consumption and accurate time synchronization.

4.3.7.3. Scalability

DPS-MAC is a fully distributed protocol. It therefore operates in isolation and autonomy at each node. Furthermore, each node operates in the exact same manner, i.e. no clusterheads

or other types of managing nodes which perform special functions are required. For this reason, DPS-MAC is scalable to different network sizes as each node only knows about its local neighbourhood. As with any MAC or routing protocol, DPS-MAC needs to maintain a table with information of its neighbours. Thus, depending on available memory on a node, the underlying hardware may only support a limited number of entries in the neighbour table which puts a limit on the node density of the network. This limit applies to all protocols and thus is not particular to the DPS-MAC protocol. It can however be envisioned that in high node density situations, the MAC and routing protocols could jointly decide to keep track of only a subset of neighbouring nodes so as to ease memory requirements.

4.4 Chapter Summary

In this chapter, the design of a new energy efficient WSN MAC protocol, DPS-MAC has been discussed. At first, the existing literature forming the design basis was reviewed and expanded on. In the discussion, it was shown that the existing literature has not sufficiently investigated clock drift as a MAC protocol factor. Thus, clock drift was researched in some more detail so as to determine a better way to deal with the problems it causes. The results from this research were then used in proposing DPS-MAC. It was shown how DPS-MAC is designed to reduce the length of preamble sending compared to the previously proposed preamble sampling MAC protocols. Lastly, it was discussed how the DPS-MAC design meets its design goals of energy efficiency, simplicity and scalability.

The aim of this chapter has been to describe in detail the characteristics and functioning of the newly proposed DPS-MAC protocol so as to show how its design builds and expands on existing protocols in order to yield increased energy efficiency.

Chapter 5
VERIFICATION PROCESS

In Chapter 4 the design of a new WSN MAC protocol was discussed. It was claimed that the new design should yield increased energy efficiency in relation to existing WSN MAC protocols. Such claims need to be verified. In this chapter, different available verification methods are first described. One of the methods is then chosen for verification of the DPS-MAC protocol. The implementation details of the chosen method of verification are then described. The results of the verification procedure are given in Chapter 6.

5.1 Method of Verification

5.1.1 Available methods

The three main methods that can be used in evaluating new proposed communication protocols are *implementation*, *analysis* or *simulation*.

5.1.1.1. Implementation

Implementation of a protocol is achieved by physically implementing the protocol on an actual piece of hardware, in this case, a sensor node. Experimental procedures can then be followed to verify the design of the protocol. The advantage of this approach is that it represents the ultimate test of a design in terms of whether the theoretical concepts can actually be applied to the real world. The downside of implementation is that this method is usually extremely expensive and time consuming. Furthermore, because of the expenses involved, the experiments are usually very inflexible, such as only having a very small number of nodes available to experiment with and thus scalability issues are difficult to

gauge. Lastly, an implementation on one platform might not say much about implementation on another platform, if the platforms are significantly different.

5.1.1.2. Analysis

Analysis verifies a protocol by mathematical procedures. Such methods can be simple or complex, depending on the level of detail of the analysis. Many underlying real processes are abstracted by mathematical simplifications. Such simplifications are not always very accurate. In essence, if verification by simulation makes certain assumptions to simplify complex problems, then verification by analysis mostly makes at least the same assumptions. To make an analytical approach very accurate thus requires very complex mathematical investigations and formulae.

5.1.1.3. Simulation

The verification of a protocol by simulation can also be done at varying degrees of complexity. Usually, a detailed and accurate simulation model can become rather intricate. Simulations are obviously not as accurate as real implementations. Especially for wireless communications, an exact model of RF transmissions can easily become a research project by itself. The advantage compared to implementation is of course the lower cost and the wide flexibility. For example, a WSN of 100 nodes can easily be simulated, but implementation of such a network purely for research purposes would mostly be infeasible.

5.1.2 Chosen method

The above discussion of verification methods is well summarized by Sadler [3]: “A MAC design typically comes with a large range of tuneable parameters. Consequently, analysis is very challenging, and we instead rely on simulations and small (often expensive) experiments”. Due to the large demands and inflexibility of implementation experiments, it was decided to use the method of simulation to verify DPS-MAC in comparison to its two main predecessors, namely WiseMAC and CSMA-MPS. Some additional concerns with the simulation verification method are shortly addressed.

A recent research article by Andel and Yasinsac [75] aims criticism at the current state of simulation results in the literature on MANETs. Three of the research’s main concerns are as follows.

- *Simulation results are often gathered by isolating certain parameters and only varying those parameters, while keeping other features constant. When such methods are used to compare two different protocols, this can yield inaccurate conclusions, since the static features can often dominate performance of individual protocols. E.g. protocol A could outperform protocol B at one bit rate, but at another bit rate, protocol B might be better. If the simulation is only done at the first bit rate, then any conclusions made about A being better than B would be biased. Interactions of different parameters should therefore be investigated as well to yield a fair comparison.* Although the above explanation by the authors seems plausible, this certainly cannot be a weakness particular to simulations. In fact, real implementation experiments are much less flexible and thus static features are even more difficult to overcome. Nevertheless, in the next chapter it will be attempted to give performance details of DPS-MAC and its predecessors under varying conditions, adjusting different parameters.
- *The majority of existing research is not independently verifiable due to a gross lack of information of protocol and simulation details (e.g. simulator version not specified, behaviour of protocol under certain conditions not specified etc.).* This problem was also observed during the literature review, as many papers do not give complete details of protocol behaviour and other information necessary to reproduce results claimed by the research. It has been attempted to give as many details of DPS-MAC behaviour as possible in the previous chapter. The rest of this chapter as well as the next chapter give details of the actual simulations.
- *The results of simulations of protocols often lack statistical validity. For example, if a new protocol is tested by a certain simulation which is run only once, then any conclusions based on this single simulation run could at worst be highly inaccurate, but at best, they are not trustworthy. Multiple runs should be conducted, preferably with different seed values for any Pseudo Random Number Generators (PRNGs) used.* The use of multiple simulation runs and varying PRNG parameters was taken into account in the simulations performed.

In other recent research, Ali et al. [76] also direct criticism at WSN simulations. Their comments are mainly aimed at the radio models used in simulations. Simulation radio models often make certain simplifications such as assuming circular transmission ranges, equal ranges for all radios, assuming that if a transmission can be heard, it can be heard

perfectly, etc. The criticism by Ali et al. can be countered by the research conducted by Zhou et al. [29]. These authors developed a detailed model of radio irregularity in WSNs. Their research shows that RF signal propagation is by no means a static process and received power at the destination is not simply a function of the distance between sender and receiver. Random fluctuations in perceived channel quality are caused among others by signal reflection, diffraction and scattering as well as variable antenna gains in different directions. In a detailed analysis of the impact of radio irregularity on two CSMA-type MAC protocols and some different routing protocols, they conclude that such irregularity has a much bigger impact on routing protocols than on MAC protocols.

Lastly, it should be kept in mind that DPS-MAC and its two main predecessors are based on the same foundation and are thus very similar to each other in their operational behaviour. DPS-MAC, WiseMAC and CSMA-MPS would therefore be affected by any simulation assumptions in a very similar way.

Based on the above discussion, it is the opinion of the author that the verification of DPS-MAC by simulation in comparison to its predecessors should yield acceptable results.

5.2 Protocol Simulation

In this section, the simulation verification method is explained in some more detail, including the aim of the simulation, a description of the simulation platform and details of the actual simulation implementation.

5.2.1 Simulation aim

The aim of the simulation of the DPS-MAC protocol is twofold. Firstly, the operation of the DPS-MAC protocol should be verified and secondly, the energy efficiency of DPS-MAC should be compared to that of its predecessors to verify that it indeed performs better than these.

The first goal of verifying the operation of the DPS-MAC protocol in fact consists of two activities. Once the protocol simulation has been implemented, the actual simulation is first verified by analyzing the simulation's debugging output. In other words, the simulation code is adjusted until the debugging output from the simulator indicates that the simulation

executes the state machine of the protocol in the correct order, with the correct timing and with the correct changes to internal variables and parameters as specified in the protocol design. Once this is achieved, the actual protocol design itself is verified by analyzing the measurement outputs of the simulation (e.g. packets generated, packets dropped, power used etc.). When the results indicate that the protocol actually achieves what it is supposed to achieve (e.g. 100% packet delivery ratio), the verification is successful. If certain simulation output indicates undesired behaviour, then the simulation output can be analyzed further to determine where the protocol design contains a flaw. After this flaw is corrected, the verification process can be started again. As such, the verification procedure is in fact a repetitive and cyclic process as shown in Figure 5.1 below.

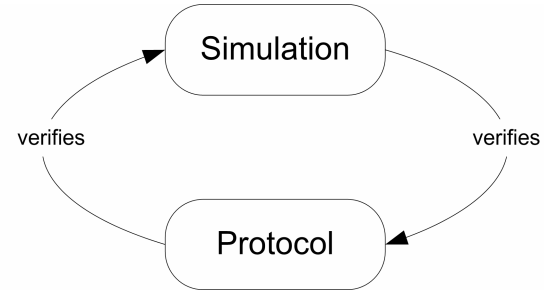

Figure 5.1. The iterative verification process by simulation.

A good example of the above process was encountered during simulation of the DPS-MAC protocol, where it was found that on certain rare occasions, a large number of packets were dropped. Upon investigation, it was discovered that this was caused by two neighbours having listen slots that were very close to each other, causing the sender to always be busy already with its own listen slot when it was supposed to send data, as described in section 4.3.4.5. The observation of this problem resulted in the listen slot hijacking procedure which is documented in the same section.

In terms of the second aim of comparing DPS-MAC's energy efficiency to that of its predecessors, it should be pointed out that the aim of the protocol simulations is *not* to derive exact values for power consumption. Rather, the aim is a relative analysis of the protocols.

5.2.2 Simulation platform

A number of different suitable simulation platforms have been observed from the WSN literature. Among them are ns-2, GloMoSim, Opnet, Qualnet and OMNeT++. Due to the

fact that OMNeT++ is open-source, its full version is freely available for academic purposes and there is an abundance of documentation and third party code available for OMNeT++, it was chosen as the simulation environment. A detailed investigation into each of the available simulators was not feasible.

Implemented on the simulation platform OMNeT++ are a number of actual simulators which are applicable to simulation of wireless networks. These include SENSIM, MAC Simulator, INET Framework and Mobility Framework. A discussion of the different attributes of these simulators is found on the OMNeT++ website [77]. It was noticed that the Mobility Framework was recommended for implementing PHY and MAC layer protocols. The MAC Simulator seemed in a somewhat immature state and also displayed less flexibility. Overall, the Mobility Framework was stated as being the current reference platform for doing mobile/ad-hoc simulations. The Mobility Framework also had the most documentation and was furthest developed at the time the research on simulators was done. This led to the choice of the Mobility Framework as the simulator on which simulations would be performed.

As the name suggests, the Mobility Framework (MF) provides a framework structure which can be used to simulate different network types. In essence, an MF simulation consists of the instantiation of a defined network. A network consists of a number of hosts and an overseeing ChannelControl module. What ChannelControl does is to dynamically establish connections between hosts, thus mobility of nodes is supported. The basic structure of each host is shown in Figure 5.2 below.

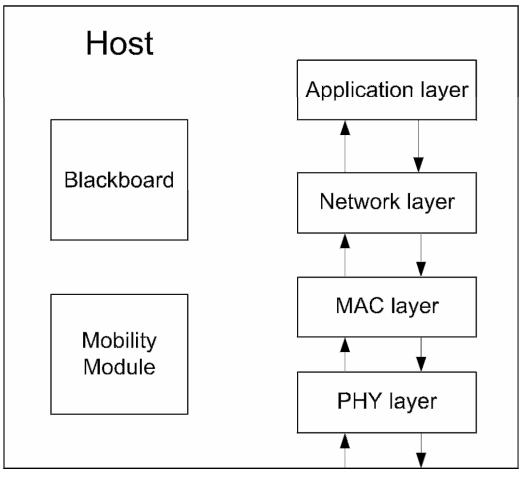

Figure 5.2. The structure of a Mobility Framework host.

In the MF, each of the protocol layers is implemented as a separate C++ module with an incoming and outgoing connection to the layer below and above. On these connections, only data structures of the OMNeT++ *Message* type can be exchanged. To provide for easier cooperation among protocols on different layers, MF furthermore provides the innovative Blackboard. Protocols can repetitively post updated information of various kinds on the Blackboard. Protocols on other layers can then subscribe to posted information of a certain desired type and be notified whenever this information is changed by the protocol posting the information. Such functionality is very useful for implementing any type of cross-layer data exchange (e.g. neighbour table sharing among layers etc.). Lastly, each host contains a Mobility Module which is responsible for communicating with the ChannelControl module to facilitate host mobility if desired.

The core of the MF provides BasicModules for each layer. These BasicModules do nothing except pass messages from bottom layers to top layers and vice versa. In order to build meaningful protocols at a certain layer, a new C++ class must be coded. This C++ class must extend the BasicModule of that layer. For example, the new class DPSMacLayer must inherit from the class BasicMACLayer.

In summary, the MF provides a flexible and well designed framework for WSN protocol simulations. DPS-MAC, along with WiseMAC and CSMA-MPS, its two predecessors, were implemented using the MF for comparison purposes.

5.2.3 Simulation details

In order to understand all the choices made for various aspects of the simulations, it is necessary to keep in mind that the DPS-MAC protocol is specifically aimed at low traffic WSNs. The simulation setup is based on a typical monitoring network, such as for example a precision agriculture WSN. This leads to the following overall simulation choices.

- A data gathering tree is simulated with a sink node as the root, collecting sensor update packets from the entire WSN.
- Nodes generate sensor update packets at regular intervals. The time between packet generations is up to 1000 seconds (approximately 16 minutes).
- The WSN consists of a bounded number of nodes (up to 50 nodes). Real implementations such as the vineyard monitoring, glacier monitoring or ocean

monitoring examples given in section 2.4.1 have typically shown sparse deployment of limited numbers of nodes.

The above choices were made so as to simulate a network that is typical of the application area for which DPS-MAC was designed. The details of the simulation implementation are given in the following sections.

5.2.3.1. Channel model

The channel model used in the simulations is based on the classic free space loss equation [78].

$$P_r = \frac{P_t}{\left(\frac{4\pi}{\lambda}\right)^2 \times (d)^2} \tag{5.1}$$

In the above equation, P_r is the received power, P_t is the transmitted power, λ is the wavelength of the transmitted signal and d is the distance between sender and receiver. Equation 5.1 is valid for line of sight (LOS) systems, however, in general, LOS conditions cannot be assumed in a WSN. To account for different types of signal environments, the above equation is modified as follows [1].

$$P_r = \frac{P_t}{\left(\frac{4\pi}{\lambda}\right)^2 \times (d)^{\alpha}} \tag{5.2}$$

In equation 5.2, α is referred to as the Path Loss Exponent (PLE). The PLE determines how sensitive the received power is to the distance travelled. The greater the PLE, the greater the signal loss/attenuation over a certain distance. The PLE is used to model different types of environments, where different amounts of signal scattering, reflection etc. occur. As just discussed, for a free space line-of-sight (LOS) system, α is set to 2. As more obstacles are added in and around the path of the signal transmission, α is increased. In [1], some typical values of α for different indoor conditions are reported, e.g. $\alpha = 2.4$ for an office with movable walls, $\alpha = 3$ for an office with fixed walls etc. For the DPS-MAC simulations, an outdoor environment is assumed. In [79], a number of signal attenuation measurements are done in outdoor environments. PLEs for various transmission distances and radio positions

are reported. The results indicate a PLE of between 2.3 and 2.6 for hilly and forested outdoor terrain. For the simulations in this dissertation, an α of 2.5 is used.

Table 5.1 shows the main parameters for the channel model.

Name	Description	Value
α	Path loss exponent.	2.5
λ	Signal wavelength.	*c*/2.4GHz (*c* - the speed of light) = $3.0x10^8/2.4x10^9$ = 0.125 m

Table 5.1. The main parameters of the channel model.

5.2.3.2. Radio Model

Two radio models are implemented in the simulations, one based on the CC2400 transceiver and the other on the CC2500 transceiver. The radio models are implemented as state machines. The state is mainly changed based on commands received from the MAC protocol. The radio has the following states.

RadioStates{ Sleep, Wake-up, Listen, Receive, Transmit, TX-RX, RX-TX, Carrier Sense}

Power consumption can be determined by evaluating the time a node spends in each of these states.

The radio model implements the idea of an interference range that is larger than the actual communication range, i.e. even if two nodes are too far apart to meaningfully communicate with each other, they can still interfere with each other's communications, as can be expected in a real-world scenario.

The occurrence of packet bit errors is determined based on Signal to Noise Ratio (SNR) and Bit Error Rate (BER) calculations as follows. The radio model contains a variable called the noise level. If a new packet is received during an ongoing reception of another packet, or if a packet is received at a power below the radio's sensitivity, the received power is added to the noise level. If on the other hand a packet is received and its received power is above the sensitivity and furthermore there is no ongoing reception, this is a valid new reception. For such a valid packet reception, the radio model calculates the SNR as

$SNR = \frac{P_r}{noise\,level}$. If during the packet reception, the noise level changes, a new SNR value is calculated and added to an SNR list. Once the packet reception is over, two steps are followed to determine whether the packet contains errors or not.

(1) If at any point in time during packet reception the SNR fell below the SNR threshold (transceiver parameter), the packet is deemed to contain bit errors. If the SNR was never below the threshold, the second step is performed.

(2) Based on the minimum SNR value over the reception time of the packet, a BER value is calculated. Both the CC2500 and CC2400 transceivers use Frequency Shift Keying (FSK) modulation. For FSK signals, the BER value can be calculated as follows [78].

$$P_b = \frac{1}{2} e^{\frac{-\varepsilon_b / N_0}{2}} \approx \frac{1}{2} e^{\frac{-\min SNR}{2}} \tag{5.3}$$

In the above equation, P_b is the probability of bit error and minSNR is the minimum SNR over the reception of the packet. If P_b denotes the probability that a bit is in error, then $1-P_b$ is the probability of no bit error. From this, the probability of no packet error is derived as follows.

$$P_{noPacketError} = (1-P_b)^{packet\,bit\,length} \tag{5.4}$$

Based on the value calculated with Equation 5.4, it is determined whether the packet has bit errors or not using a random number generator.

The main parameters used in the radio model are summarized in Table 5.2 below.

Name	Description	Value
SNR threshold	Minimum SNR value required to be able to recognize a packet at all	4 dB
Carrier Frequency	RF carrier frequency.	2.4 GHz
P_t	Transmit power.	0 dBm = 1 mW
Receiver sensitivity	Minimum receive power necessary for receiver radio to recognize as valid data.	CC2400: -87 dBm CC2500: -81 dBm
Thermal noise	Level of the thermal background noise.	-110 dBm
Carrier Sense threshold	The receive power above which the channel is deemed busy during the carrier sense operation.	-90 dBm

Table 5.2. The main parameters of the radio model.

5.2.3.3. MAC protocol model

The functionality of DPS-MAC, WiseMAC and CSMA-MPS are all implemented as described in section 4.3, 3.4.1.7 and 3.4.1.8 respectively. Again, the MAC protocol implementations are based on a state machine. The states of the MAC layer are listed below.

MACStates{ Sleep, Wake-up, Periodic_Listen, Receive_Preamble, Transmit_Preamble, Receive_Preamble_Ack, Transmit_Preamble_Ack, Transmit_Negative_Ack, Receive_Data, Transmit_Data, Receive_Final_Ack, Transmit_Final_Ack, Carrier_Sense, Listen_Preamble_Ack, Listen_Data, Listen_Final_Ack, TX-RX, RX-TX}

The MAC layer sends commands to the radio and receives events from the radio via the Mobility Framework's Blackboard feature.

The basic functionality of the three MAC protocols is implemented as already described, and thus, not much more will be elaborated here. The main parameters of the DPS-MAC protocol have already been listed in Table 4.3. The parameters for the other two protocols are sub-sets of those listed in Table 4.3. Some of these parameters are varied during the simulations, such as the periodic listen interval T_w. The MAC layer parameters that remain largely unchanged (unless otherwise stated) are listed in Table 5.3 below (see Table 4.3 for descriptions of parameters).

Parameter	Value (unless otherwise stated)
$T_{n\text{-}disc}$	Random value from the interval [900..1000] seconds
maxDriftEstimateMisses	2
maxSlotEstimateMisses	4
maxTotalMisses	6
maxNoOfPreambles	Depends on communicating neighbour's state: state=UNSYNCHRONIZED: T_w/(time per preamble) state= GOT_SLOT_ESTIMATE: 4ΘL/(time per preamble) state= GOT_DRIFT_ESTIMATE: 20 (Time per preamble is the time that it takes to send a preamble packet, switch the radio to receive, listen for a preamble ACK and switch radio back to transmit.)
BuffSize	10 packets

Table 5.3. The main parameters of the MAC protocol model.

5.2.3.4. Packet generator

A packet generator is incorporated into every node so that each node is a producer of network traffic. The traffic generated by the packet generator is classified by the time between successive packet generations. This time is defined as a truncated normal random variable (normal variable truncated to positive values only) with mean and standard deviation settable for each simulation. Thus, the traffic can be made increasingly periodic in nature by setting a small value for the standard deviation, or the traffic can be made irregular, by setting a large standard deviation. A further parameter of the packet generator is the size of the data payload in each sensor update packet.

As already pointed out at the beginning of section 5.2.3, for the monitoring-type applications that are considered in this research, sensor update packets are mostly generated at regular periodic intervals. Furthermore, since each sensor update contains a specific pattern of data (e.g. temperature reading and address of originating node etc.), fixed data sizes can be expected. The parameters of the packet generator are shown in Table 5.4 below.

Parameter	Value
Mean time between packet generation	variable (smaller than 1000 seconds)
Std. deviation of time between packet generation	variable
Payload data size	30 bytes

Table 5.4. The parameters of the packet generator.

5.2.3.5. Clock drift modelling

The Mobility Framework does not inherently support each host having its own clock. In fact, normally all hosts get their time base from the overall “absolute” simulation time. In order to simulate the clock drift that occurs in each node in a real WSN, a *getClock()* function is implemented at each node which converts the current absolute time to the local (drifted) node time. This occurs as follows.

Firstly, when a node is first initialized, it is assigned a drift value Θ_{node} from the interval $[-\Theta, +\Theta]$ by means of a triangular random distribution as shown in Figure 5.3 below.

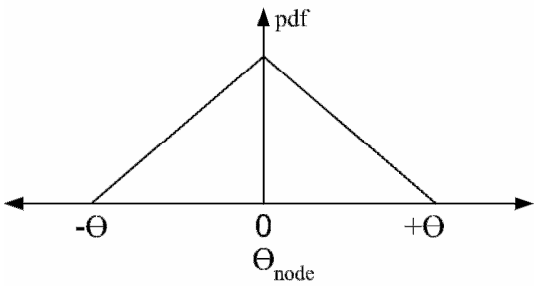

Figure 5.3. The shape of the probability density function for assigning of Θ_{node} values to nodes.

The triangular distribution is used since in practice, any crystal that does not meet the manufacturer's quality standards (i.e. maximum frequency tolerance of ±Θ) will be disposed of, so that the ±Θ tolerance limit is a hard limit, as can be observed in the triangular distribution, which ends completely at ±Θ.

The absolute simulation time is obtained using the OMNeT++ *simTime()* call, which returns a double value indicating the elapsed number of seconds since simulation start. The *getClock()* function of each node determines the local (drifted) time from the absolute simulation time as follows.

$$local\ time = simTime() + (simTime() \times \Theta_{node}); \tag{5.5}$$

When the node sets a timer in order to wake up at some point in the future (e.g. to send a packet to a neighbour), then the interval after which the timer should expire must be converted to an absolute interval. In other words, say a node wants to wake up in 1 second, this 1 second represents the local time that should have passed when the node wakes up. However, since the Mobility Framework only works with absolute simulation time, the local time interval of 1 second must be converted to an absolute time interval. The calculation to do this conversion is the inverse of Equation 5.5 above and is done as follows.

$$real\ interval = \frac{local\ interval}{(1+\theta_{node})} \tag{5.6}$$

Lastly, say a node wants to wake up in 1 second to send a packet to a neighbour. It uses the above equation to calculate the real interval after which to wake up, but furthermore, it adds a random variable to account for oscillator instability as follows.

$$time\ at\ which\ to\ wake\ up\ for\ sending = simTime() + real\ interval + \delta_{noiseA} + \delta_{noiseB} \quad (5.7)$$

The δ_{noise} variables in the above equation are the same as in Equation 4.2. They represent the random oscillator noise. Thus, δ_{noise} is a normal random variable with zero mean and standard deviation as shown in Figure 4.6. The reason why there are two δ_{noise} variables in the above equation is because in fact the clock instability of both the sending and receiving node must be simulated. Furthermore, since for the simulations, a time between communications, L, of less than 1000 seconds is assumed, it can be seen in Figure 4.6 that the standard deviation of δ_{noise} will never be larger than 10^{-6} seconds. However, for the sake of the simulations, the standard deviation is fixed at 10^{-6}. This is thus worst case scenario.

The parameters of the clock inaccuracy modelling are shown in Table 5.5 below.

Parameter	Value (unless otherwise stated)
Θ - maximum oscillator frequency tolerance	40 ppm
Θ_{node} – a specific node's average crystal oscillator drift	Triangular distribution random value with lower bound -Θ, mode zero and upper bound + Θ.
Std. deviation of oscillator instability	$1x10^{-6}$

Table 5.5. The parameters of the crystal inaccuracy modelling.

5.2.3.6. Network topology

Two different topologies are used in the simulations. For an initial comparison among the three MAC protocols, a simple two-hop network is used i.e. a sink node and two sensor nodes. After this, a WSN with nodes distributed randomly on the "playground" is used to perform larger simulations. The seed value of the random number generator used for node placement was adjusted until a network with a more or less even node distribution was found. Furthermore, it was ensured that the WSN was fully connected i.e. that every node had at least one path to the sink node. If not all nodes were connected, the unconnected nodes would not partake in network functionality and would thus be useless for the purpose of statistics gathering.

As stated at the beginning of section 5.2.3, typical networks that have been observed in practice have displayed limited numbers of sparsely distributed nodes and therefore, such a topology is used in the simulations.

The simulation parameters related to network topology are shown in Table 5.6 below.

Parameter	Value (unless otherwise stated)
Number of nodes	50
Playground size	400m x 400m

Table 5.6. The parameters related to network topology

5.2.3.7. Sink node

The sink node is simulated in the same way as a normal node except that it is assumed to have unlimited energy resources. It can thus listen on the medium continuously, which means that the sink node's neighbours can schedule a send to the sink node at any time.

5.2.3.8. Network setup and routing

The focus of this dissertation is the development of a MAC protocol. However, a MAC protocol is not designed to perform any routing functionality. Thus, a simple network setup and routing procedure is incorporated in the simulations to yield a meaningful and organized flow of traffic that mimics a real application scenario.

For the purpose of network setup, the DPS-MAC neighbour discovery feature is used. The sink node initiates the process by broadcasting a neighbour discovery preamble train. In the preamble, the sink node indicates that it is at hop count zero. The sink node's neighbours hear the broadcast and see that they have received a neighbour discovery packet with a hop count smaller than their own initial value hop count. They thus update their own hop count to the received hop count plus one. The nodes then continue with their own neighbour discovery, and so, the process spreads from the sink node outward. At the end, every node knows its hop count and has a populated neighbour table.

As soon as a node learns its hop count, it has a path to the sink and the packet generator is started. Once a packet is entered into the transmit queue, a transmission is scheduled to the

downstream neighbour who's next listen slot is the earliest from now. The fact that packets are always sent to downstream neighbours ensures that no routing loops can exist and furthermore, the packet always moves closer to the sink. The fact that the downstream neighbour with the earliest next listen slot is always chosen ensures that the delay is as small as possible, which is in fact not a critical issue in most environmental monitoring type WSNs.

5.2.3.9. Statistics gathering

A number of statistics are gathered in the simulation process. Most importantly, the time spent by the radio in each of its states is measured. At the end of the simulations, the average power consumption of a node's radio can be determined from the time it spent in each of the states. Other statistics that are gathered include the number of packets generated, the number of packets sent successfully, the number of packets dropped, as well as the number of packets gathered by the sink node so as to form a view of what is happening in the network.

Statistics gathering is only started after the initial network setup procedure is complete. The reason for doing this is that a view of protocol performance at steady-state network conditions is desired.

5.2.3.10. Overall simulation parameters

For the protocol simulations, version 3.2 of OMNeT++ and version 1.0-a-6 of the Mobility Framework were used.

Furthermore, OMNeT++ uses the Mersenne Twister Pseudo Random Number Generator (PRNG), which has a period of 2^{19937} - 1 [80]. In fact, for any OMNeT++ simulation, multiple PRNGs can be initialized. Then, in the simulation code, whenever a random number is required, the PRNG to be used can be specified. In this way, different aspects of a simulation can be isolated from each other so that no dependencies can develop. For the simulations documented in this dissertation, eight PRNGs were used for generating different random values. In order to bring statistical validity to the obtained results, each simulation is executed repetitively for a number of runs. During each run, different seed values are used for the PRNGs. This makes sure that every run is indeed different from the others.

Some overall simulation parameters are shown in Table 5.7 below.

Parameter	Value (unless otherwise stated)
Number of times each simulation is executed.	10
Duration of each simulation	1 day

Table 5.7. The overall simulation parameters.

5.3 Chapter Summary

In this chapter, the verification process of the newly designed MAC protocol and its predecessors has been discussed. First, it was shown that simulation is the most feasible method of verifying the DPS-MAC protocol and comparing it to its predecessors. Then, OMNeT++ and the Mobility Framework, on which the simulations are implemented, were described. Lastly, the details of the simulation implementation were discussed.

The aim of the chapter has been to provide as much information on the simulations as possible so as to make them repeatable, as well as to establish a basis for the validity of the simulation results obtained in the following chapter.

Chapter 6
RESULTS AND DISCUSSION

In the previous chapter, the process of verification of the newly proposed DPS-MAC protocol in comparison to its predecessors was discussed. In this chapter, the results of the verification and comparison process are presented. Furthermore, observations made from the results are analyzed and discussed for the purpose of gaining insight into the manner in which the protocols' designs affect their performance under various operating conditions.

6.1 Simulation Results

6.1.1 Initial protocol comparisons

For the purpose of conducting an initial comparison between the operation of the WiseMAC, CSMA-MPS and DPS-MAC protocols, two different networks were simulated. At first, the basic protocol operation was verified on a small two-hop network. After this, the protocols were compared using a more realistic network of 50 scattered nodes to display their properties with the influence of contention among nodes taken into account. The results are documented below.

6.1.1.1. Small-scale scenario

The two hop topology used for the small-scale comparison of the three protocols is shown in Figure 6.1 below.

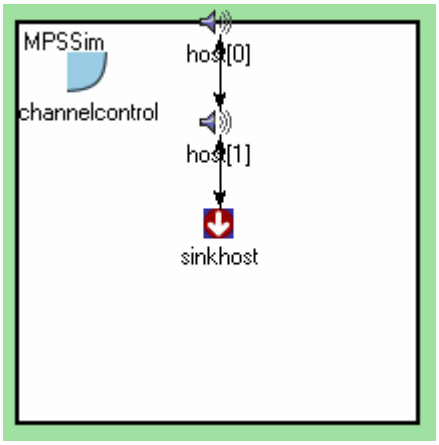

Figure 6.1. The simple two-hop WSN used for initial protocol comparison.

In the network shown in Figure 6.1 above, there is no contention among the nodes, since each node only has one downstream neighbour. Further, since the sink node does not generate traffic, the hidden-node problem does not play a role in this network. Collisions are therefore highly unlikely. The purpose of using the small network shown in the above figure is to show the protocol operation under very simple conditions.

The main parameters used for this simulation are shown in Table 6.1 below. All other parameters were set to the values already defined in the previous sections.

Parameter	Value
Number of times each simulation is executed.	10 runs
Duration of each simulation	1 day
Θ - maximum oscillator frequency tolerance	40 ppm
Mean time between packet generation	10 minutes
Std. deviation of time between packet generation	0.2 seconds
Number of nodes	2
Playground size	200m x 200m
T_w	100ms, 200ms, 500ms, 1s, 2s, 5s
Radio model	CC2400

Table 6.1. The main parameters of the initial comparison simulations.

The statistics that were gathered from the simulations are the percentage of generated packets delivered to the sink node, the average radio power consumption per node, the

average time spent listening or receiving and the average time spent sending. Each of these statistics was collected in every simulation run, where one simulation run represents one protocol, with one value of T_w. Thus, ten such runs were performed for each of the three protocols, for each of the six different T_w values.

The reason for needing to know the percentage of generated packets delivered to the sink node is to have a basis for comparing the radio power consumption. In other words, if protocol *A* displays lower radio power consumption during a simulation run than protocol *B*, then this result is only valid if both protocols indeed generated and delivered the same amount of data packets to the sink during the simulation run. If protocol *A* in fact delivered fewer packets to the sink node than protocol *B*, then, regardless of whether it consumed less power, it did not fulfil its purpose as well as protocol *B*. The number of packets actually generated in the network during a simulation run is the same for all protocols because the packet generator generates traffic at a constant rate. Thus the only difference between the protocols can be in the percentage of packets delivered to the sink.

The packet delivery ratio for each protocol is shown in Figure 6.2 below.

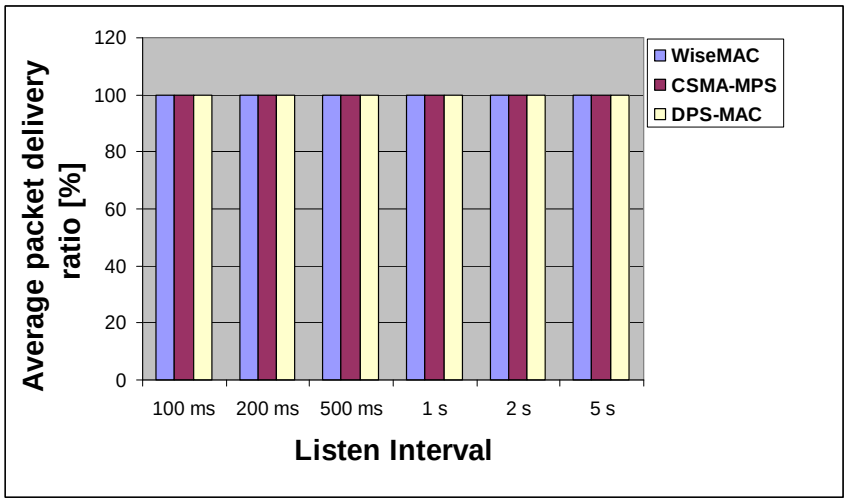

Figure 6.2. The average packet delivery ratio for the three protocols with various T_w values in a small two hop network.

It is clear from the above figure that 100% delivery rate was achieved by all protocols for all of the chosen T_w values, which should be expected in such a simple network. On this basis, the average radio power consumption per node can be compared as shown in Figure 6.3 below.

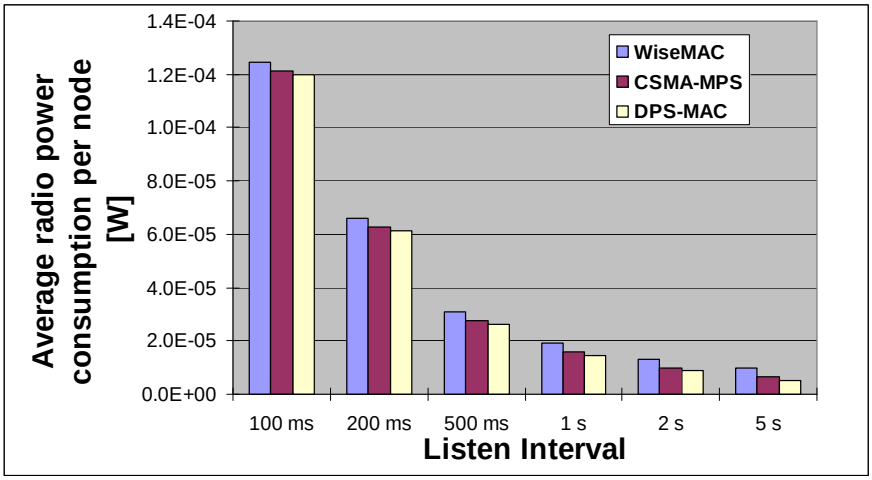

Figure 6.3. The average radio power consumption per node for the three protocols with various T_w values in a small two hop network.

From the above figure it is seen that, as expected, the radio power consumption reduces drastically as T_w is increased. This is because the node wakes up much less often to check for incoming data when T_w is increased. At low T_w values, the power consumption is thus dominated by the listening/receiving time, which is more or less similar for each of the three protocols as shown in Figure 6.4 below.

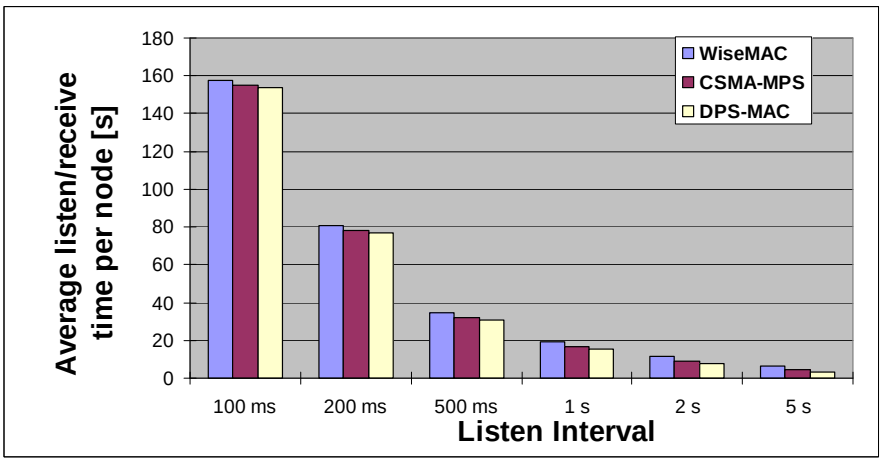

Figure 6.4. The average listen/receive time per node for the three protocols with various T_w values in a small two hop network.

Since for small values of T_w, the listening/receiving time is dominated by the periodic listen slots, which are the same for all three protocols, the figure above shows very similar listening/receiving times for the three protocols. The slight differences are explained as follows. With WiseMAC, a node detecting an incoming preamble has to carry on listening

to the whole preamble until the actual data arrives. This causes reception overhead. With CSMA-MPS on the other hand, the receiving node can end the preamble sending early by replying to the sender with a preamble ACK packet. Lastly, DPS-MAC shortens the preamble sending even further. During preamble sending, the sender repetitively listens for incoming preamble ACK packets. Thus, shortening the preamble also shortens the overall listen/receive time.

The shortening of the preamble sending is most clearly demonstrated in Figure 6.5 below.

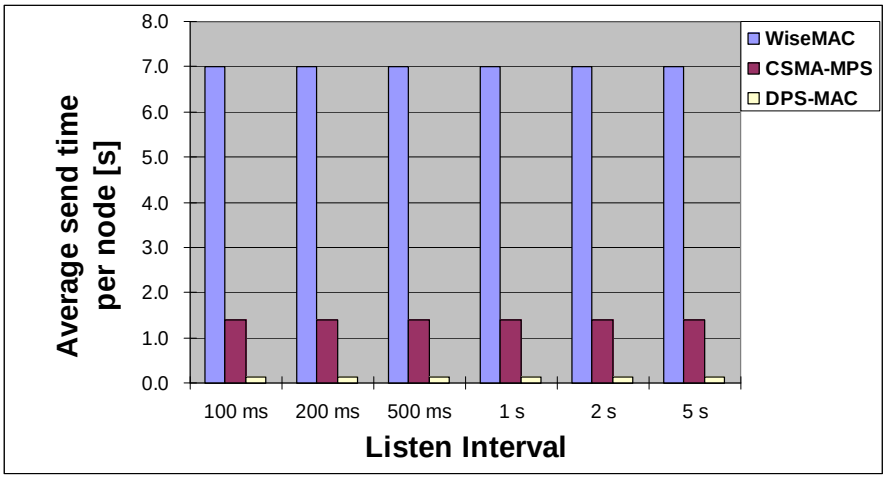

Figure 6.5. The average send time per node for the three protocols with various T_w values in a small two hop network.

From the above figure it is seen that WiseMAC has the longest preambles by catering for the worst case clock drift always. CSMA-MPS also caters for the worst case clock drift but can end the preamble as soon as the receiver's listen slot is reached. Lastly, DPS-MAC dynamically adjusts to the actual clock drift experienced between two nodes and thus starts preamble sending a few microseconds before the receiver's listen slot starts. The reason for the sending time being independent of T_w is that for the simple network under consideration, with no contention and collisions, the sending time is only dependent on the message generation rate.

What can be deduced from the above simulations is that in general, WiseMAC consumes the most power, followed by CSMA-MPS and then DPS-MAC. Furthermore, the differences in power consumption become increasingly significant as T_w is increased,

because at higher T_w values, the power consumption is more and more dominated by the time spent sending, which is shortest for DPS-MAC and longest for WiseMAC.

6.1.1.2. Realistic scenario

In order to compare the three protocols in a more realistic scenario, a 50 node network topology was used. The nodes were randomly distributed in a 400m x 400m area. After a number of repetitions of varying the seed value of the random number generator used for placing the nodes, a suitably distributed network topology with more or less even node placements was found. It is shown in Figure 6.6 below.

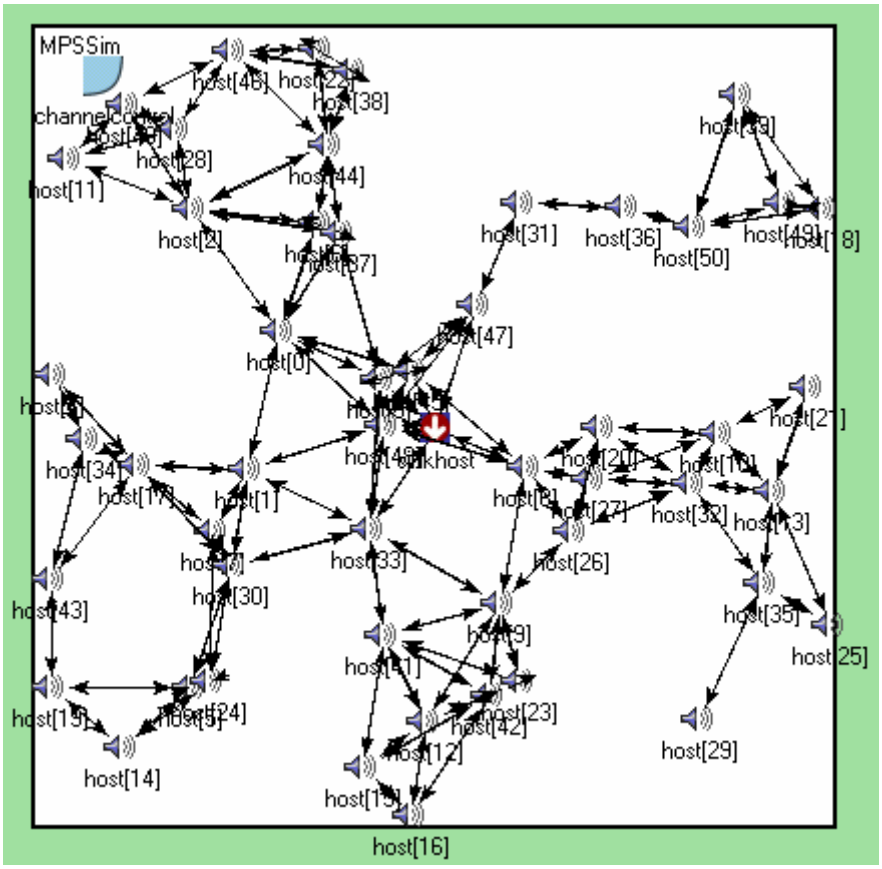

Figure 6.6. The 50 node network topology used for simulating a realistic environmental monitoring-type WSN scenario.

It can be seen from the above network topology that if more nodes would be added to fill the visible gaps in the network, this would actually improve network performance as nodes would have additional routes to the sink host. However, in reality this is not always possible, and the above topology is viewed as a realistic representation of what an environmental monitoring WSN might look like.

All three protocols were again simulated on the network shown in Figure 6.6. The main parameters for the simulations are the same as those in Table 6.1, except that 50 nodes are used instead of two and the playground size is 400m x 400m.

The packet delivery ratios are shown in Figure 6.7 below.

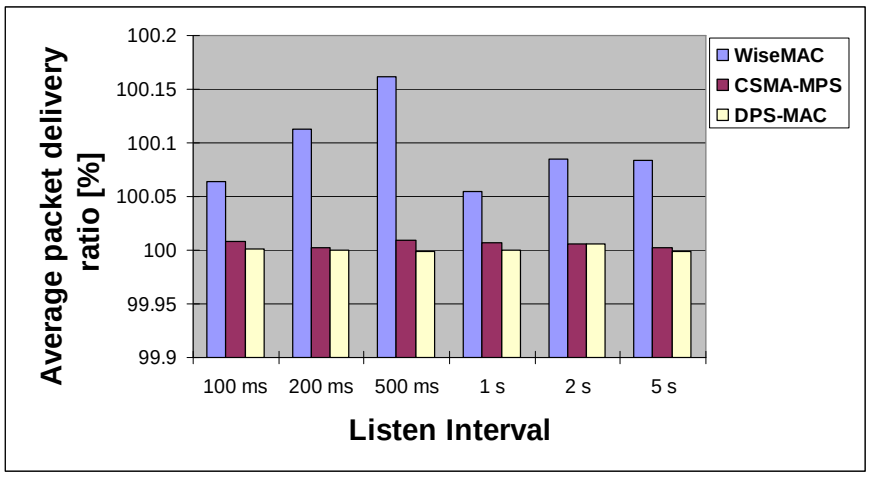

Figure 6.7. The average packet delivery ratio for the three protocols with various T_w values in a realistic 50 node network.

The above figure shows that the packet delivery ratio is sometimes in fact higher than 100%, which happens when data packets are duplicated during the forwarding process. Duplication of packets can occur when a data packet is successfully received by a recipient, but the ACK packet that is returned to the sender is corrupted due to a collision. When this happens, the sender believes that the transaction failed, whereas the receiver believes the transaction was successful. The sender will resend the same data packet again and so, two copies of the same data are now present in the network. Such a sequence of events is an example of an unsuccessful send attempt. A successful send attempt occurs when the final ACK packet is correctly received and no duplicate packet is created.

The average radio power consumption per node for the realistic scenario simulations is shown in Figure 6.8 below.

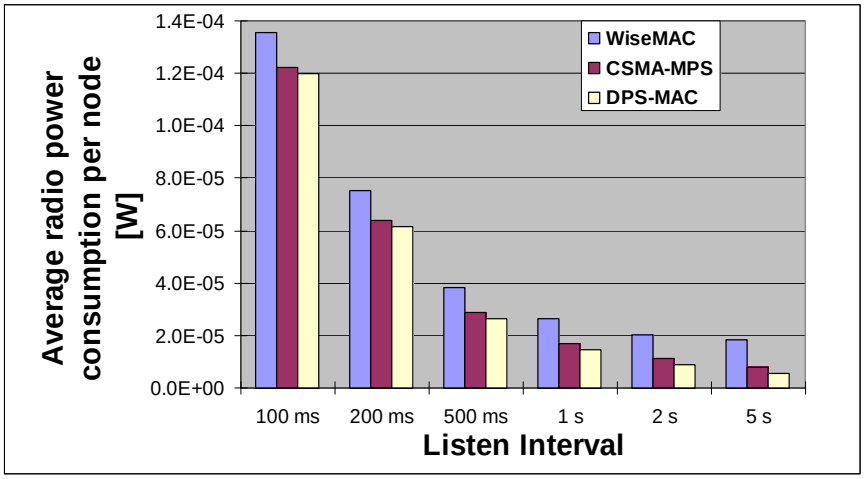

Figure 6.8. The average radio power consumption per node for the three protocols with various T_w values in a realistic 50 node network.

As in the previous simple network, WiseMAC consistently performs worse than CSMA-MPS, and DPS-MAC is the most energy efficient of the protocols.

It was observed during the simulations that the WiseMAC protocol displays a higher sensitivity to contention than the CSMA-MPS and DPS-MAC protocols. In other words, during high contention levels (more nodes competing for the medium), WiseMAC's energy consumption increases significantly compared to CSMA-MPS and DPS-MAC. There are two main reasons for this. Firstly, the *hidden-node* problem causes collisions in situations where there is contention among neighbouring nodes. The longer the preamble, the more energy is wasted when such a collision occurs during data transmission. Since WiseMAC has the longest preambles of the three protocols, the energy wasted by the hidden-node problem is amplified. Secondly, with WiseMAC, any node that detects a preamble during its listen slot has no option but to carry on listening to the whole preamble and data frame, even if it is not the intended recipient. The reason for this is that, unlike with CSMA-MPS and DPS-MAC, the preamble contains no indication of who the recipient of the ensuing data packet is. A node could thus waste significant energy by listening to a packet destined for another node, which is the *overhearing* problem.

A further observation made from the simulations is that the average amount of time spent sending actually increases with increasing T_w values. The WiseMAC protocol suffers the most from this behaviour, whereas CSMA-MPS and DPS-MAC do so to a lesser extent as shown in Figure 6.9 below.

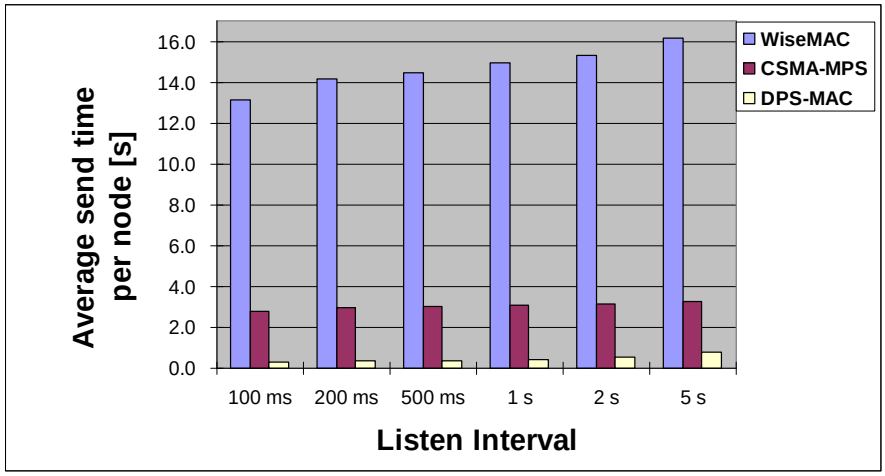

Figure 6.9. The average send time per node for the three protocols with various T_w values in a realistic 50 node network.

In theory, the sending time should only be dependent on the traffic generated and not on T_w. The increase in sending time is explained as follows. When contention causes a node to repetitively fail to communicate with one of its neighbours, the neighbour's state is changed to *unsynchronized*. When this happens, the next time a packet must be sent to that neighbour, the maximum preamble length is set to T_w, as explained in section 4.3.1.1. If the neighbour can still not be reached, it is eventually removed from the neighbour table. This can cause the node to start the neighbour discovery process again in case it has no other downstream neighbours. A neighbour discovery preamble train needs to be sent out for a duration of T_w seconds. Thus, both neighbour resynchronization and neighbour discovery require preambles of length T_w. As T_w is increased, the energy dissipated in resynchronization and neighbour discovery can therefore become profuse. Furthermore, the long preambles can cause additional contention among neighbouring nodes and the whole process is amplified as shown in Figure 6.10 below.

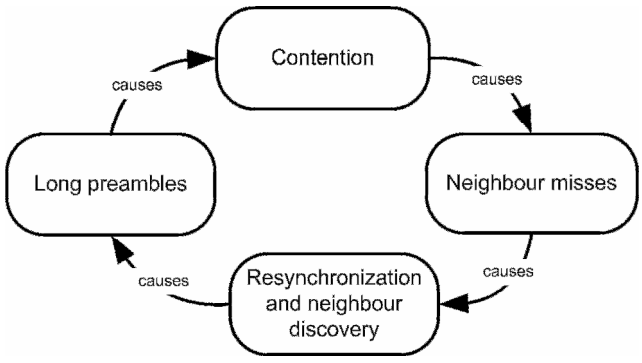

Figure 6.10. The contention amplification process observed with long preambles.

WiseMAC by design has the longest preambles of the three protocols. It was shown by simulation to perform the worst of the three MAC protocols, and furthermore it displays a higher sensitivity to contention among multiple neighbouring nodes. It was thus decided to exclude WiseMAC from the remaining comparison simulations. CSMA-MPS and DPS-MAC show similar characteristics in the initial protocol simulations and thus, they are analyzed in more detail in the following sections.

6.1.2 Sensitivity to varying listen interval

In order to show the sensitivity of CSMA-MPS and DPS-MAC to the magnitude of the listen interval T_w, simulations were performed with all parameters fixed to chosen values except for a varying listen interval. In the previous sections, it was seen that at small T_w values, the radio power consumption is totally dominated by the periodic listen slot time, because of the high frequency of the listen slots. As seen in Figure 6.8, only at T_w values of 1 second and above, the average power consumption starts to stabilize and true low power operation becomes possible. This is the operation area of interest for the low traffic WSNs considered in this dissertation. The parameters used for the simulations are shown in Table 6.2 below.

Parameter	Value
Number of times each simulation is executed.	10 runs
Duration of each simulation	1 day
Θ - maximum oscillator frequency tolerance	40 ppm
Mean time between packet generation	15 minutes
Std. deviation of time between packet generation	0.2 seconds
Number of nodes	50
Playground size	400m x 400m
T_w	1s, 5s, 10s, 15s, 20s
Radio model	CC2400

Table 6.2. The main parameters of the T_w sensitivity simulations.

The packet delivery ratio of the two protocols is shown in Figure 6.11.

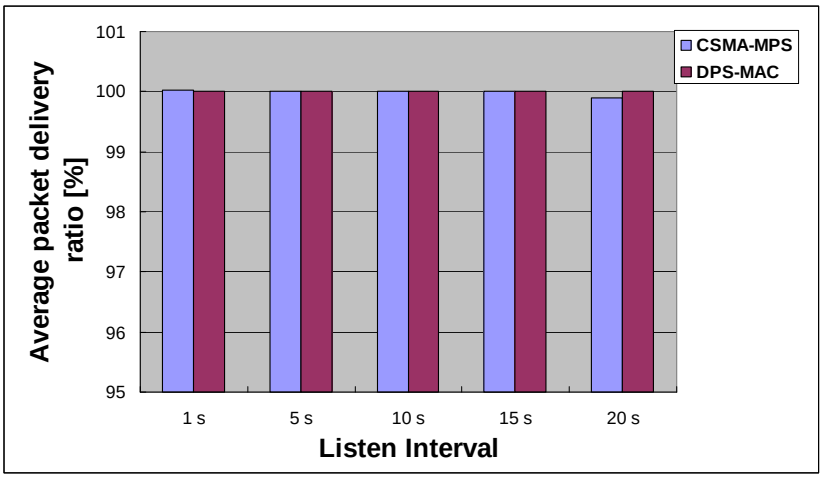

Figure 6.11. The average packet delivery ratios of CSMA-MPS and DPS-MAC for various values of T_w.

A packet delivery ratio slightly below 100% for CSMA-MPS at a T_w of 20 seconds is observed in Figure 6.11. This is solely attributed to some packets still being in transit in the network when the simulation was stopped. The average radio power consumption is compared in Figure 6.12 below.

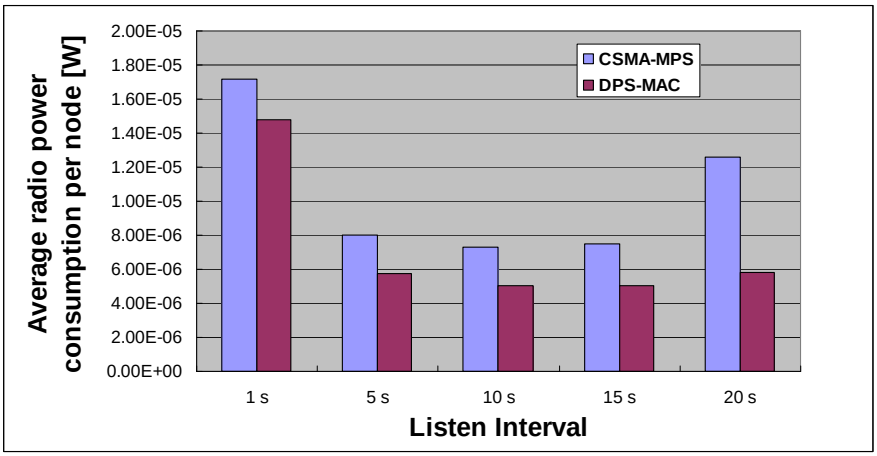

Figure 6.12. The average radio power consumption per node of CSMA-MPS and DPS-MAC for various values of T_w.

The graph shown in the above figure behaves as expected, except for the significant rise in power consumption for a T_w of 20 seconds. The reason for this rise is that at higher T_w values, the energy cost of resynchronization and neighbour discovery becomes very large and dominates the power consumption as explained in the previous section. The longer the preambles, the more pronounced this problem becomes, which is why CSMA-MPS

displays a sharper rise in power consumption than DPS-MAC. The reason for not simulating even higher values of T_w is that the power consumption values rise more and more drastically for both CSMA-MPS and DPS-MAC, causing the network to become highly unstable. It is postulated that in a sparse network with negligible contention and thus no repetitive resynchronization and neighbour discovery, the power consumption would continue to decrease as the listen interval is increased. However, for most realistic scenarios, it must be assumed that resynchronization and neighbour discovery are likely to occur, at least to some extent. Thus, there will be a threshold T_w value where the network becomes unstable if T_w is increased any further because of the contention amplification process shown in Figure 6.10. It is seen from Figure 6.12 that for a T_w of 5, 10 and 15 seconds, both CSMA-MPS and DPS-MAC consume less than 10µW of transceiver power, DPS-MAC always having lower power consumption than CSMA-MPS. The average energy savings of DPS-MAC over CSMA-MPS are shown in Figure 6.13 below.

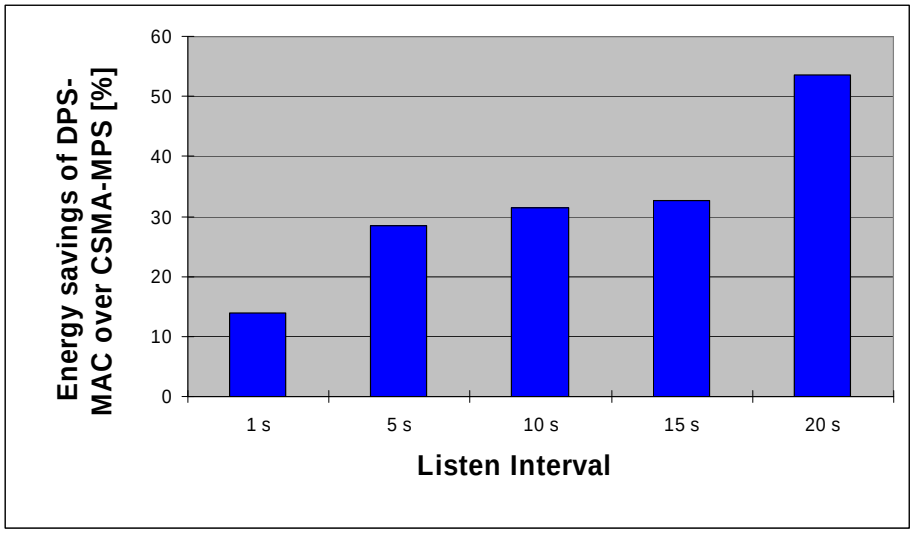

Figure 6.13. The average percentage energy savings of DPS-MAC over CSMA-MPS for various values of T_w.

As just discussed, the final bar on the graph above (T_w = 20s) represents the point at which CSMA-MPS becomes unstable. For even higher T_w values, DPS-MAC would soon follow suit, and the energy savings of DPS-MAC would become negligible.

6.1.3 Sensitivity to varying traffic load

To show the protocols' sensitivities to varying traffic loads, again all other parameters were fixed to chosen values and the traffic load was then changed. To change the traffic

load, the time between periodic packet generations was varied, so as to make packet generation more or less frequent in each node. Furthermore, one non-periodic packet generation scenario was also simulated to show the protocols' behaviour under non-regular traffic patterns. This was achieved by setting a high value for the standard deviation of the time between packet generations. The parameters used in the simulations are shown in Table 6.3 below.

Parameter	Value
Number of times each simulation is executed.	10 runs
Duration of each simulation	1 day
Θ - maximum oscillator frequency tolerance	40 ppm
Number of nodes	50
Playground size	400m x 400m
T_w	10s
Radio model	CC2400
Mean time between packet generation, Std. deviation of time between packet generation	a) 20s, 0.2s b) 60s, 0.2s c) 5min, 0.2s d) 10min, 0.2s e) 15min, 0.2s f) 500s, 500s

Table 6.3. The main parameters of the traffic sensitivity simulations.

As is seen from the above table, five different periodic traffic scenarios were simulated (a-e in the last row). The last entry of the last row (f) shows the variable traffic scenario. The mean time between packet generations is set to 500 seconds, and the standard deviation is also set to 500 seconds. This causes a widely varying time between packet generations, except for the constraint that this time is limited to be no less than 20 seconds and no more than 1000 seconds.

The variation in traffic load means that in each simulation, a different amount of packets are generated per node. The variation of generated packets is shown in Figure 6.14 below. Note that a logarithmic Y-scale is used.

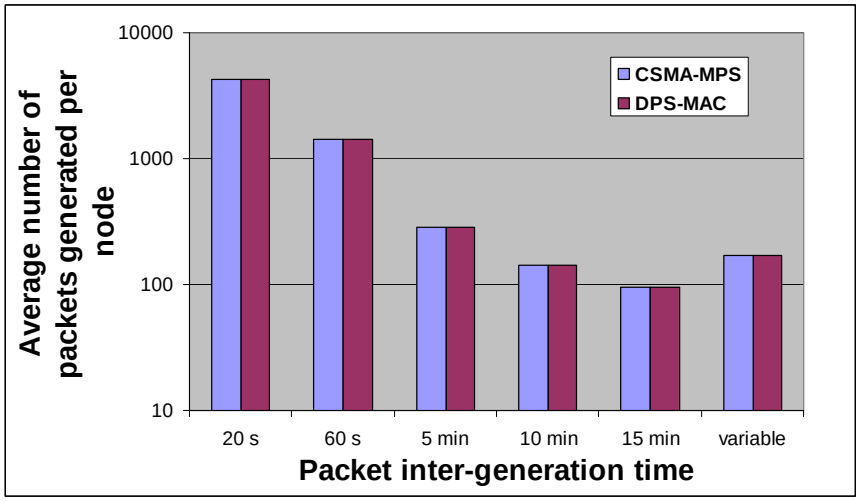

Figure 6.14. The average number of packets generated by each node for the simulations of various traffic loads.

The percentages of generated packets actually delivered to the sink node for the different loads are shown in Figure 6.15 below.

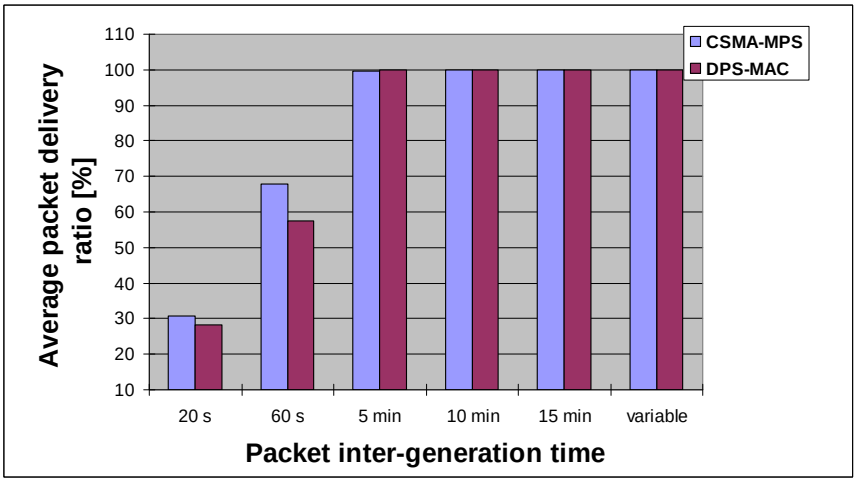

Figure 6.15. The average packet delivery ratios for CSMA-MPS and DPS-MAC for various traffic loads.

From the above figure it is seen that for packet inter-generation times of 20 or 60 seconds the network is overloaded for both protocols, causing low packet delivery ratios. The reason why the network becomes overloaded for short packet inter-generation times is because the listen interval T_w puts a limit on the amount of traffic that can be carried in the network. For these simulations, T_w is set to 10 seconds. In order to support higher traffic loads, a shorter T_w needs to be chosen.

A second observation is that during network overload, CSMA-MPS delivers a higher percentage of packets than DPS-MAC, which means that it is more robust to network overload. The reason for this is as follows. With DPS-MAC, nodes are accurately synchronized to each other and thus, two nodes wanting to send to the same destination at the same time are likely to cause collisions. The random time t_{rand} which is added to the predicted wake-up time can reduce collisions to some extent, but total prevention is not possible. With CSMA-MPS on the other hand, the degree of synchronization between nodes depends to a greater extent on the time since last communication and thus, two nodes trying to send to the same destination at the same time are likely to have differing degrees of synchronization to the destination. This can further reduce collisions, as is observed in the network overload scenarios above. It must however be kept in mind that this reduction in collision probabilities comes at the cost of longer preambles and higher power consumption, as seen in Figure 6.16 below.

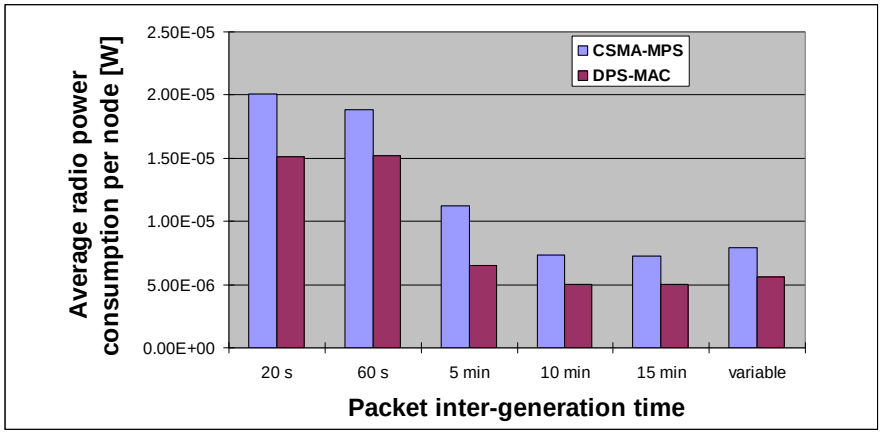

Figure 6.16. The average radio power consumption per node for CSMA-MPS and DPS-MAC under varying traffic loads.

CSMA-MPS's longer preambles therefore cause it to be more robust and show higher packet delivery ratios under overloaded conditions, but they also expend more energy. Furthermore, it was noted that the longer preambles in fact keep the CSMA-MPS network in the overload condition for longer. In other words, at traffic loads where DPS-MAC is already functioning without overload, CSMA-MPS is still overloaded. The reason for this is that the longer preambles keep the medium occupied for longer. The longer the medium is occupied by an ongoing transmission, the less opportunities neighbouring nodes have to conduct their own communications, the fewer packets can be delivered, the more energy is spent in attempting transmission. It was observed that the CSMA-MPS network is still on

the verge of overload at a packet inter-generation time of five minutes. At this traffic load, DPS-MAC is already functioning normally, whereas CSMA-MPS is still expending energy trying to battle against network overload, which explains the peak value of energy savings of DPS-MAC over CSMA-MPS at the five minute mark in Figure 6.17 below.

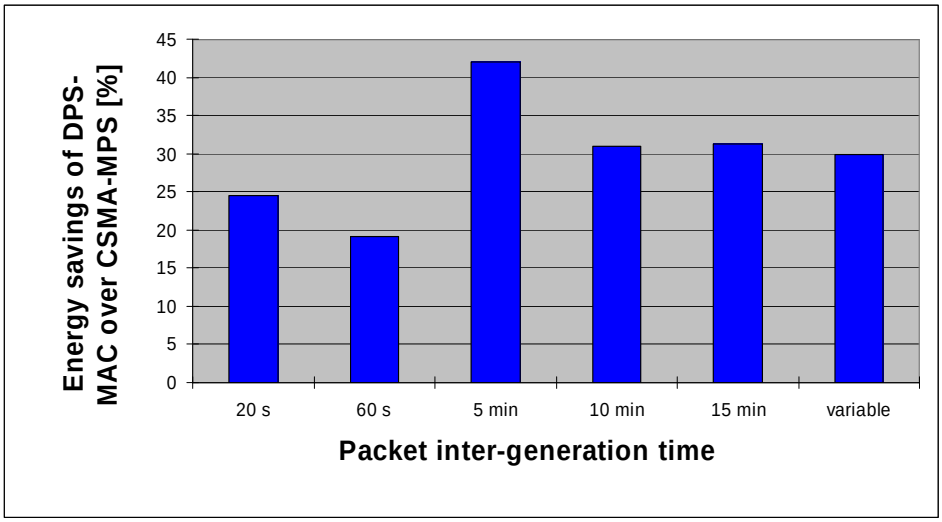

Figure 6.17. The average percentage energy savings of DPS-MAC over CSMA-MPS for various network traffic loads.

In terms of the variable traffic load simulations, it is observed from the above figures that both DPS-MAC and CSMA-MPS are capable of supporting an irregular traffic pattern. No packet losses are observed, as both protocols show 100% delivery ratios for this traffic scenario. DPS-MAC however outperforms MPS-MAC in terms of energy savings, as is seen in the right-most bar of Figure 6.17 above.

6.1.4 Sensitivity to oscillator frequency tolerance rating

In this section, the impact of different oscillator frequency tolerance ratings on the performance of CSMA-MPS and DPS-MAC is investigated. By oscillator frequency tolerance rating, the maximum Θ value of a crystal oscillator as specified by its manufacturer is meant. Once again, all other simulation parameters were fixed and only the Θ value was varied. The simulation parameters are shown in Table 6.4 below.

Parameter	Value
Number of times each simulation is executed.	10 runs
Duration of each simulation	1 day
Θ - maximum oscillator frequency tolerance	10, 20, 40, 60, 80 ppm
Number of nodes	50
Playground size	400m x 400m
T_w	10 seconds
Mean time between packet generation	15 minutes
Std. deviation of time between packet generation	0.2 seconds
Radio model	CC2400

Table 6.4. The main parameters of the frequency tolerance sensitivity simulations.

For brevity, the packet delivery ratio graph is not shown. All simulations yielded 100% delivery ratios. The average per node radio power consumption for the two protocols with varying clock inaccuracy ratings is shown in Figure 6.18 below.

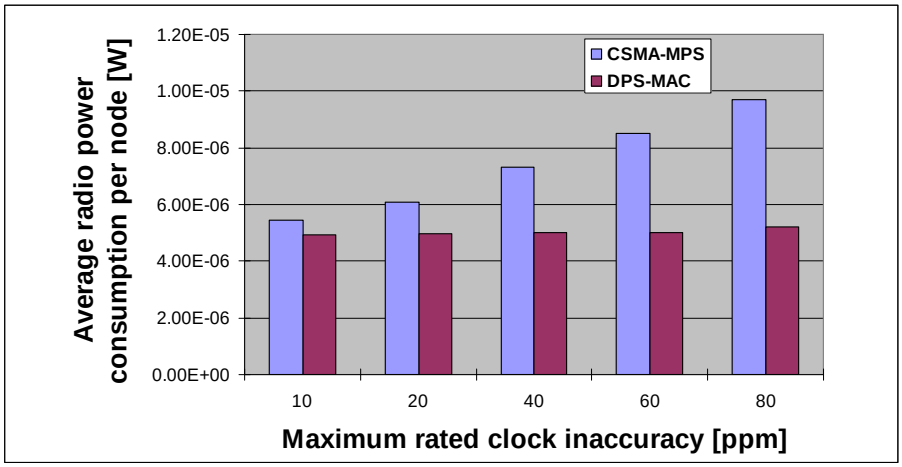

Figure 6.18. The average radio power consumption per node for CSMA-MPS and DPS-MAC with various frequency tolerance (Θ) ratings.

CSMA-MPS always starts sending a preamble 2ΘL seconds before the neighbour's listen slot is predicted to start, where L is the time since last communication. Thus, intuitively, the larger the value of Θ, the longer the preamble sending, the more energy is expended. With DPS-MAC on the other hand, the protocol dynamically adjusts its predictions of the neighbour's listen slot to the actual observed clock inaccuracy and not the worst case clock

inaccuracy. Thus, the DPS-MAC radio energy consumption is largely independent of the Θ rating. This is seen in Figure 6.18 above. An alternative view is shown in Figure 6.19 below, which displays the percentage energy saving of DPS-MAC over CSMA-MPS for different Θ values.

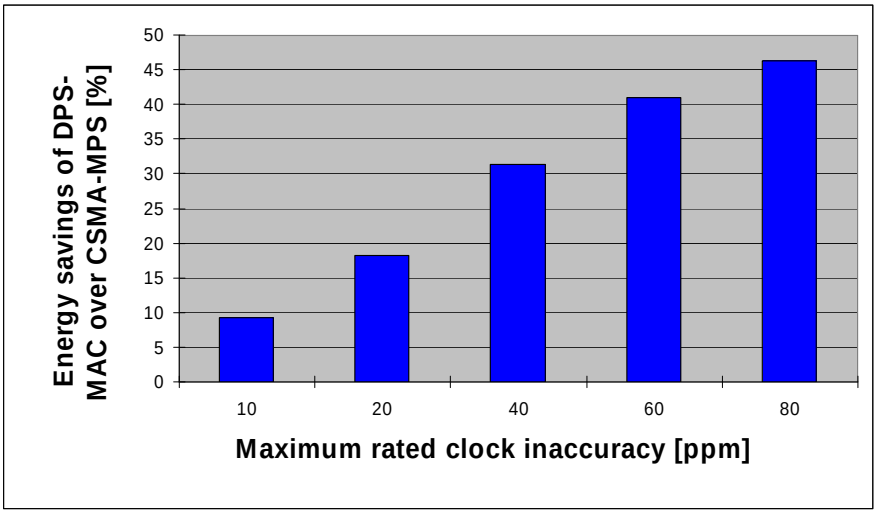

Figure 6.19. The average percentage energy savings of DPS-MAC over CSMA-MPS for various frequency tolerance (Θ) ratings.

As expected, the figure above shows that DPS-MAC shows increasing energy savings compared to CSMA-MPS as the maximum oscillator frequency tolerance is increased.

6.1.5 Sensitivity to varying radio parameters

In this section, the operation of DPS-MAC on two different radios is analyzed. Up to now, simulations have been based on the CC2400 radio. Its operation is now briefly compared to the CC2500 radio. It should be noted that the CC2500 is in fact less sensitive than the CC2400, meaning that it has a shorter transmission range. Thus, in order to show the same network area coverage, nodes with CC2500 radios would have to be more densely spaced. However, for the simulations in this section, the focus is on determining the impact of the radios' timing parameters (bit rate etc.) on the energy consumption of communications. It was thus desired to keep the network topology the same for the simulations of both radios. Differing topologies for the two radios could cause unwanted side effects on the simulation results, such as different numbers of neighbours, resulting in different contention levels. It was therefore decided to simulate the CC2500 radio with the same -87dBm sensitivity as the CC2400 radio instead of its actual -81dBm sensitivity.

The parameters used in the simulations are shown in Table 6.5 below.

Parameter	Value
Number of times each simulation is executed.	10 runs
Duration of each simulation	1 day
Θ - maximum oscillator frequency tolerance	40 ppm
Number of nodes	50
Playground size	400m x 400m
T_w	10 seconds
Mean time between packet generation	15 minutes
Std. deviation of time between packet generation	0.2 seconds
Radio model	CC2400 & CC2500

Table 6.5. The main parameters of the radio parameter sensitivity simulations.

With respect to the simulations in this section, the most important difference between the CC2400 and CC2500 radios is that the CC2400 has a maximum bit rate of 1Mbps, whereas the CC2500 only reaches half of this capacity at 500kbps. Theoretically, the impact of a lower bit rate on the sending and receiving times should be as follows.

- *Sending time*: The sending time can be divided into preamble time and data time.
    - *Preamble time*: The preamble time will be the same, regardless of bit rate. This is because the preamble time of DPS-MAC depends only on the accuracy of the clock drift estimate for each neighbour as well as the random time t_{rand} (see section 4.3.1.2 and 4.3.1.3 for details of the t_{rand} value). A lower bit rate will only mean that fewer but longer preamble packets are sent during the preamble time.
    - *Data time*: The data time will increase with a lower bit rate. This is because the data size is fixed and thus, the lower the bit rate, the longer it takes to transmit the data.

  Since preamble time is the same for high and low bit rate radios and data time is longer for low bit rate radios, thus total sending time can only be higher for lower bit rate radios.
- *Receiving time*: The receiving time is divided into periodic listening and data receiving time.

- *Periodic listening time*: The periodic listen time is always greater for lower bit rates because the periodic listen time must be longer than the time to transmit a preamble ACK packet. A preamble ACK packet is of a fixed length and thus, the lower the bit rate, the longer it takes to transmit such a packet.
- *Data receiving time*: The data receiving time in fact consists of the preamble and the data packets. A receiver will receive at most one preamble packet, answer with a preamble ACK packet and then receive the data packet. Since both the preamble and the data packets are of a fixed length, the data receiving time will increase as the bit rate is decreased.

Both periodic listen time and data receiving time are longer for lower bit rates, thus total receive time will also be longer for lower bit rates.

The conclusion from the above discussion is that the total time spent communicating using the DPS-MAC protocol will always be longer for nodes with lower bit rate radios. The increased communication times would only be countered if the low bit rate radio were to display significantly lower power consumption than a high bit rate radio in the active states. However, in terms of the CC2500 and the CC2400, this is not the case. The average radio power consumption per node for the CC2400 and CC2500 radios is shown in Figure 6.20 below.

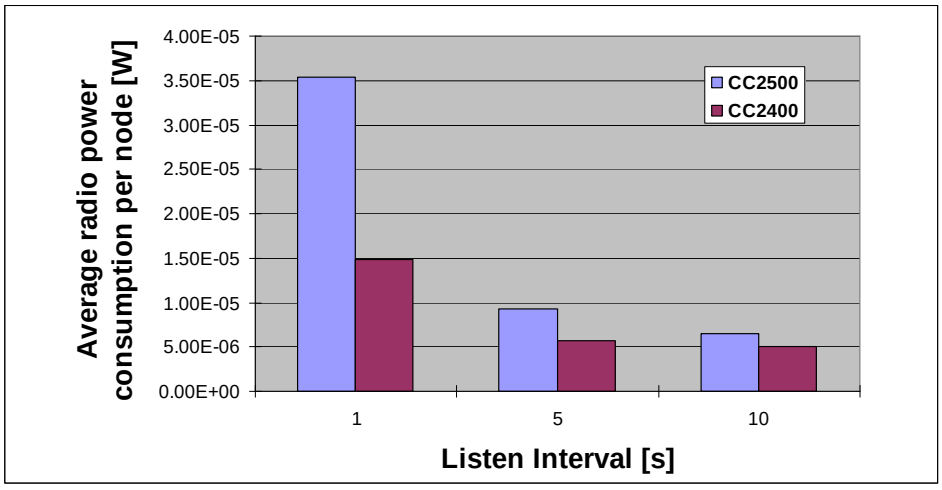

Figure 6.20. The average radio power consumption per node for the CC2400 and CC2500 radios with different T_w values.

The reason why the relative difference in power consumption between the radios can be observed to decrease as T_w is increased is because at higher listen intervals, the power consumption is dominated by the length of the preambles that are sent, which are the same for both of the radios, as discussed earlier in this section.

In summary, it is clear that high bit rate radios are advantageous to the operation of preamble sampling protocols.

6.2 Discussion

Some of the main observations that were made in this chapter are discussed in this section.

It was noted that for the preamble sampling protocols under consideration, there is a trade-off in the choice of listening interval T_w. If the listening interval is made too short, then the power consumption of the nodes in the network will become profuse, On the other hand, for a given traffic load in the network, if the listen interval is made too large, the traffic will not be supported any more, contention will increase and the long preambles used for resynchronization or neighbour discovery will cause an amplification of the contention levels, yielding an unstable network. The listen interval must thus be chosen to support the expected traffic patterns in the network, while yielding as much energy savings as possible.

It was shown that protocols with longer preambles tend to become unstable more easily and, once unstable, become stable again less easily. On the other hand, while in the unstable or overloaded state, they tend to be more robust in terms of delivering a higher percentage of offered traffic to the sink node, albeit at higher energy expenditure.

Taking the above results into account, DPS-MAC was shown to have lower energy consumption than both WiseMAC and CSMA-MPS. The main reason for this is that DPS-MAC has the shortest preambles of the three protocols and thus yields the most energy savings under various operating conditions.

6.3 Chapter Summary

In this chapter, the results of the simulations conducted to verify the DPS-MAC design in comparison to its predecessors, WiseMAC and CSMA-MPS, were presented and discussed. The DPS-MAC protocol was shown to consistently outperform its predecessors under various operating conditions. DPS-MAC achieves this by reducing preamble lengths as far as possible. The reduced preamble lengths cause not only energy savings under normal operating conditions, but lead to the network having a lower tendency to become unstable or overloaded.

The aim of this chapter has been to verify the proposed MAC protocol design and to show that it indeed yields an increase in energy efficiency compared to existing protocols.

Chapter 7
CONCLUSION

7.1 Summary

In this dissertation, a new energy-efficient Medium Access Control protocol for Wireless Sensor Networks has been proposed.

At first, it was shown that the growing field of WSNs indeed presents vast potential for application in different domains. Environmental monitoring, battlefield surveillance, medical applications, agricultural monitoring and disaster relief operations are just some of the many possible application domains for which the WSN concept can be exploited. However, a number of challenges need to be addressed in order for WSNs to truly be useful in real applications. WSNs are required to operate autonomously, without aid or maintenance, for long periods of time. Energy efficiency is thus one of the most important challenges, and needs to be dealt with at every stage of a WSN design. It was shown that radio communications among sensor nodes in a network are the main consumer of energy, and thus, they need to be carefully controlled so as to extend the lifetime of a node. The protocol that is directly in control of the radio interface of a node, managing access to the shared wireless medium, is the MAC protocol.

A thorough review of existing literature on WSN MAC protocols was performed. It was found that these protocols can be grouped into two major classes, namely *contention-based* and *schedule-based* protocols. Various protocols in each of these classes, as well as the two classes themselves were investigated as to their applicability to different WSN types. It

was found that insufficient attention has been given to WSNs with truly low traffic rates. It was shown that in fact a number of useful applications would indeed display such low traffic rates, including environmental monitoring, precision agriculture and other applications. The research was thus focused on this WSN domain, the aim being to add to the available literature by developing a new MAC protocol specifically for such WSNs.

It was found that the MAC protocols that most closely resemble applicability to low traffic monitoring WSNs are the preamble sampling family of protocols, of which WiseMAC and CSMA-MPS were shown to be the most energy efficient designs to date. It was observed that both of these protocols always make provision for the worst case clock drift that can occur between two sensor nodes. The topic of clock drift was thus researched in more detail and it was found to in fact consist of two main parts. A fixed frequency offset in the quartz crystals of two nodes causes their clocks to drift apart over time at a fixed rate. This is the *inaccuracy* of the clock. On the other hand, noise in oscillator circuitry and other processes can cause short term fluctuations in crystal frequency. This is the *instability* of the clock. It was found in relevant literature that the greater of the two components is clock inaccuracy.

7.2 Critical Evaluation of Own Work

The new proposed MAC protocol exploits the finding that clock drift is mainly caused by a fixed frequency offset. Unlike WiseMAC and CSMA-MPS, the protocol does not make provision for the worst case clock drift, but for each pair of communicating nodes learns the clock drift experienced between the two nodes. By doing this, the protocol dynamically adjusts its operation to the clock drift between two nodes and can thus shorten preamble sending significantly. The dynamic nature of the preamble sending caused the protocol to be named the Dynamic Preamble Sampling MAC (DPS-MAC) protocol.

Through simulations, the operation of DPS-MAC was verified in comparison to its predecessors WiseMAC and CSMA-MPS. It was shown that DPS-MAC indeed causes reduced energy consumption by shortening preamble sending times under various network operating conditions. This reduction in sending time also caused the WSN employing DPS-MAC to be less prone to become unstable or overloaded due to neighbour resynchronization or high traffic loads respectively.

7.3 Future Work

Some of the possible future research opportunities identified during the work on this dissertation are as follows.

Firstly, the development of a dedicated hardware platform specifically aimed at low power WSNs would be advantageous to future research. So far, implementations have often consisted of discreet components, and furthermore, certain attractive hardware features such as high bit rate radios, sophisticated clock distribution units and so forth have often been encountered in isolation and not integrated in one package. It is postulated that in order to see the realization of truly energy efficient and affordable sensor nodes in the future, hardware will need to be developed that will incorporate as many advantageous features as possible into a single-chip solution.

Secondly, the interaction of DPS-MAC with various routing protocols could be researched. It is believed that the routing strategy used in a WSN can have a great influence on the operation of the MAC protocol, both positively and negatively. This influence should be determined so as to devise optimal protocol stacks for different WSN types. Furthermore, the opportunity for cross-layering between the routing and MAC protocols for increased energy efficiency in certain scenarios such as extremely dense WSNs should be researched.

Finally, an implementation of DPS-MAC and suitable routing protocols on real hardware would be advantageous to determining the actual interaction between such protocols in a real WSN scenario.

REFERENCES

[1] S. Mahlknecht, "Energy-Self-Sufficient Wireless Sensor Networks for the Home and Building Environment," Ph.D. thesis, Institute of Computer Technology, Technical University of Vienna, Vienna, Austria, 2004.

[2] V. Raghunathan, C. Schurgers, S. Park and M. B. Srivastava, "Energy-aware wireless microsensor networks," *IEEE Signal Processing Mag.*, vol. 19, no. 2, pp. 40-50, Mar. 2002.

[3] B. M. Sadler, "Fundamentals of Energy-Constrained Sensor Network Systems," *IEEE A&E Systems Mag.*, vol. 20, no. 8, pp. 17-35, Aug. 2005.

[4] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393-422, March 2002.

[5] K. Romer, and F. Mattern, "The Design Space of Wireless Sensor Networks," *IEEE Wireless Communications Mag.*, vol. 11, no. 6, pp. 54-61, December 2004.

[6] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "A Survey on Sensor Networks", *IEEE Communications Mag.*, vol. 40, no. 8, pp. 102-114, August 2002.

[7] Crossbow Technology Inc., http://www.xbow.com. Last accessed October 2006.

[8] Chipcon AS, CC2400 datasheet (Rev. 1.5), March 2006, Available: http://www.chipcon.com. Last accessed August 2006.

[9] Chipcon AS, CC2500 datasheet (Rev. 1.2), June 2006, Available: http://www.chipcon.com. Last accessed August 2006.

[10] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*. Chichester: Wiley, 2005, ch. 5.

[11] N. Abramson, "The aloha system–another alternative for computer communications," *Proc. Fall Joint Comput. Conf., AFIPS Conf.*, Montvale, NJ, 1970, vol. 44, pp. 281–285.

[12] L. Kleinrock and F. Tobagi, "Packet Switching in Radio Channels: Part I-Carrier Sense Multiple Access Modes and their Throughput Delay Characteristics," *IEEE Trans. Communications*, vol. COM-23, pp. 1400-1416, Dec. 1975.

[13] F. A. Tobagi and L. Kleinrock, "Packet Switching in Radio Channels: Part II-The Hidden Terminal Problem in Carrier Sense Multiple-Access and the Busy-Tone Solution," *IEEE Trans. Communications*, vol. COM-23, pp. 1417-1433, Dec. 1975.

[14] P. Karn, "MACA - A New Channel Access Method for Packet Radio," presented at the ARRL/CRRL Amateur Radio 9th Computer Networking Conference, September 22, 1990.

[15] V. Bharghavan, A. Demers, S. Shenker and L. Zhang, "MACAW: A Media Access Protocol for Wireless LAN's," *1994 Proc. ACM SIGCOMM Conf.*, pp. 212-225.

[16] *IEEE Wireless LAN medium access control (MAC) and physical layer (PHY) specification*, IEEE Standard 802.11, 1997.

[17] S. Singh and C. S. Raghavendra, "PAMAS - Power aware multi-access protocol with signalling for Ad Hoc networks," *Computer Communication Review*, vol. 28, no. 3, pp. 5-25, 1998.

[18] W. Hoferkamp and S. Olariu, "A power and mobility-aware wireless protocol for ad-hoc networks," *2000 Proc. MILCOM Conf.*, pp. 292-296.

[19] W. Ye, J. Heidemann and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," *2002 Proc. INFOCOM Conf.*, pp. 1567-1576.

[20] W. Ye, J. Heidemann and D. Estrin, "Medium Access Control With Coordinated Adaptive Sleeping for Wireless Sensor Networks," *IEEE/ACM Trans. Networking*, vol. 12, pp. 493-506, June 2004.

[21] I. Demirkol, C. Ersoy and F. Alagöz, "MAC Protocols for Wireless Sensor Networks: A Survey," *IEEE Communications Mag.*, vol. 44, no. 4, pp. 115-121, April 2006.

[22] L. C. Zhong, R. Shah, C. Guo and J. Rabaey, "An Ultra-Low Power and Distributed Access Protocol for Broadband Wireless Sensor Networks," Berkeley Wireless Research Center, Berkeley, CA, Tech. Rep., 2001, [Online]. Available: http://bwrc.eecs.berkeley.edu/Publications/2001/Ultra-Low_pwr_dist_access_prtcl/NETWORLD_2001.pdf. Last accessed October 2006.

[23] A. Woo and D. E. Culler, "A Transmission Control Scheme for Media Access in Sensor Networks," *2001 Proc. International Conference on Mobile Computing and Networking, ACM*, pp. 221-235.

[24] C. Suh and Y.-B. Ko, "A Traffic Aware, Energy Efficient MAC Protocol for Wireless Sensor Networks," *2005 Proc. ISCAS Symp.*, pp. 2975 – 2978.

[25] T. van Dam and K. Langendoen, "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks," *2003 Proc. SenSys Conf.*, pp. 171-180.

[26] J. Ai, J. Kong and D. Turgut, "An Adaptive Coordinated Medium Access Control for Wireless Sensor Networks," *2004 Proc. ISCC Conf.*, pp. 214 – 219.

[27] G. Lu, B. Krishnamachari and C. S. Raghavendra, "An Adaptive Energy-Efficient and Low-Latency MAC for Data Gathering in Wireless Sensor Networks," *2004 Proc. IPDPS*, pp. 224-232.

[28] H. Pham and S. Jha, "An Adaptive Mobility-Aware MAC Protocol for Sensor Networks (MS-MAC)," *2004 Proc. International Conf. on Mobile Ad-hoc and Sensor Systems*, pp. 558-560.

[29] G. Zhou, T. He, S. Krishnamurthy and J. A. Stankovic, "Models and Solutions for Radio Irregularity in Wireless Sensor Networks," *ACM Trans. on Sensor Networks*, vol. 2, no. 2, pp.221-262, May 2006.

[30] P. Lin, C. Qiao and X. Wang, "Medium Access Control With A Dynamic Duty Cycle For Sensor Networks," *2004 Proc. IEEE Wireless Communications and Networking Conf.*, vol. 3, pp. 1534-1539.

[31] C. Suh, Y.-B. Ko and D.-M. Son, "An Energy Efficient Cross-Layer MAC Protocol for Wireless Sensor Networks," *Lecture Notes in Computer Science*, vol. 3842, pp. 410-419, 2006.

[32] C. Schurgers, V. Tsiatsis, S. Ganeriwal and M. Srivastava, "Optimizing Sensor Networks in the Energy-Latency-Density Design Space," *IEEE Trans. on Mobile Computing*, vol. 1, pp. 70-80, March 2002.

[33] C. Schurgers, V. Tsiatsis, M. B. Srivastava, "STEM: Topology management for energy efficient sensor networks," *2002 Proc. Aerospace Conf.*, pp 3-1099 - 3-1108.

[34] A. El-Hoiydi, "Spatial TDMA and CSMA with Preamble Sampling for Low Power Ad Hoc Wireless Sensor Networks," *2002 Proc. ISCC Conf.*, pp. 685 – 692.

[35] A. El-Hoiydi, "Aloha with Preamble Sampling for Sporadic Traffic in Ad HocWireless Sensor Networks," *2002 Proc. ISCC Conf.*, pp. 3418 - 3423.

[36] C. C. Enz, A. El-Hoiydi, J.-D. Decotignie, V. Peiris, "WiseNET: an ultralow-power wireless sensor network solution," *IEEE Computer* Mag., vol. 37, no. 8, pp. 62-70, Aug 2004.

[37] A. El-Hoiydi and J.-D. Decotignie, "WiseMAC: an ultra low power MAC protocol for the downlink of infrastructure wireless sensor networks," *2004 Proc. ISCC Symp.*, pp. 244 – 251.

[38] S. Mahlknecht, M. Böck, "CSMA-MPS: a minimum preamble sampling MAC protocol for low power wireless sensor networks," *2004 Proc. IEEE Workshop on Factory Comm. Systems*, pp. 73-80.

[39] S. Mahlknecht, M. Böck, "On the use of High Bit Rate Transceivers for Low Duty Cycle Wireless Sensor Networks," *2004 Proc. IEEE AFRICON Conf.*, vol. 2, pp. 1235-1238.

[40] J. Polastre, J. Hill, D. Culler, "Versatile low power media access for wireless sensor networks," *2004 Proc. ACM SenSys Conf.*, pp. 95-107.

[41] K.-J. Wong and D. K. Arvind, "SpeckMAC: Low-power Decentralised MAC Protocols for Low Data Rate Transmissions in Specknets," *2006 Proc. 2nd ACM International Workshop on Multi-hop Ad Hoc Networks: from Theory to Reality*, pp. 71-78.

[42] T. Boscardin, S. Cai, R. X. Gao, and W. Gong, "Energy Efficient MAC protocol for Condition Monitoring Sensor Networks," *2004 Proc. 43rd IEEE Conf. on Decision and Control*, vol. 2, pp. 1496-1501.

[43] K. Nuli, H. Sharif, S. Ci and L. Cheng, "An Adaptive Medium Access Control for Energy Efficiency in Wireless Sensor Networks," *2004 Proc. IEEE International Conf. on Networking, Sensing and Control*, vol. 1, pp. 34-39.

[44] J. Haapola, "MAC energy performance in duty cycle constrained sensor network and effect of sleep," *2004 IEEE International Workshop on Wireless Ad-Hoc Networks*, pp. 320-324.

[45] M. C. Vuran and I. F. Akyildiz, "Spatial Correlation-Based Collaborative Medium Access Control in Wireless Sensor Networks," *IEEE/ACM Trans. Networking*, vol. 14, no. 2, pp. 316-329, April 2006.

[46] M. Srivastava, "Wireless Sensor & Actuator Networks," Center for Embedded Networked Sensing, University of California at Los Angeles, Tutorial, 2005, [Online]. Available: http://nesl.ee.ucla.edu/tutorials/mobicom05/. Last accessed October 2006.

[47] W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," *2000 Proc. Ann. Hawaii International Conference on System Sciences*, pp. 908-918.

[48] K. Sohrabi and G. J. Pottie, "Performance of a novel self-organization protocol for wireless ad-hoc sensor networks," *1999 Proc. VTC Conf.*, pp. 1222-1226.

[49] K. Sohrabi, J. Gao, V. Ailawadhi and G. J. Pottie, "Protocols for Self-Organization of a Wireless Sensor Network," *IEEE J. Personal Comm.*, vol. 7, no. 5, pp. 16-27, Oct. 2000.

[50] V. Rajendran, K. Obraczka and J. J. Garcia-Luna-Aceves, "Energy-efficient, collision-free medium access control for wireless sensor networks," *2003 Proc. SenSys Conf.*, pp. 181-192.

[51] G. Pei and C. Chien, "Low Power TDMA in Large Wireless Sensor Networks," *2001 Proc. MILCOM Conf.*, pp. 347-351.

[52] B. Tavli and W. B. Heinzelman, "TRACE: Time Reservation Using Adaptive Control for Energy Efficiency," *IEEE J. on Selected Areas in Communications*, vol. 21, no. 10, pp. 1506-1515.

[53] M. Ringwald and K. Römer, "BitMAC: A Deterministic, Collision-Free, and Robust MAC Protocol for Sensor Networks," *2005 Proc. IEEE 2nd European Workshop on Wireless Sensor Networks*, pp. 57-69.

[54] R. Kalidindi, L. Ray, R. Kannan and S. Iyengar, "Distributed Energy Aware MAC Layer Protocol for Wireless Sensor Networks," *2003 Proc. International Conf. on Wireless Networks*, pp. 282-286.

[55] R. Kannan, R. Kalidindi and S. S. Iyengar, "Energy and rate based MAC protocol for wireless sensor networks," *ACM SIGMOD Record*, vol. 32, no. 4, pp. 60-65, 2003.

[56] Z. Chen and A. Khokhar, "Self Organization and Energy Efficient TDMA MAC Protocol by Wake Up For Wireless Sensor Networks," *2004 Proc. IEEE Conf. on Sensor and Ad Hoc Communications and Networks*, pp. 335-341.

[57] I. Rhee, A. Warrier, M. Aia and J. Min, "ZMAC: a Hybrid MAC for Wireless Sensor Networks," *2005 Proc. ACM SenSys Conf.*, pp. 90-101.

[58] L. F. W. van Hoesel, T. Nieberg, H. J. Kip and P. J. M. Havinga, "Advantages of a TDMA based, energy-efficient, self-organizing MAC protocol for WSNs," *2004 Proc. IEEE Vehicular Technology Conf.*, vol. 3, pp. 1598-1602.

[59] J. Li and G. Y. Lazaruo, "A bit-map-assisted energy-efficient MAC scheme for wireless sensor networks," *2004 Proc. IEEE/ACM 3rd International Symposium on Information Processing in Sensor Networks*, pp. 55-60.

[60] L. F. W. van Hoesel and P. J. M. Havinga, "A Lightweight Medium Access Protocol (LMAC) for Wireless Sensor Networks," 2004. [Online]. Available: http://wwwhome.cs.utwente.nl/~hoesel/publications/VanHoesel-INSS2004.pdf. Last accessed October 2006.

[61] M. Ali, T. Suleman and Z. A. Uzmi, "MMAC: A Mobility-Adaptive, Collision-Free MAC Protocol for Wireless Sensor Networks," *2005 Proc. IEEE International Performance, Computing and Communications Conference*, pp. 401-407.

[62] B. H. Liu, N. Bulusu, H. Pham and S. Jha, "A Self-Organizing, Location-Aware Media Access Control Protocol for DS-CDMA Sensor Networks," *2004 Proc. IEEE International Conf. on Mobile Ad-hoc and Sensor Systems*, pp528-530.

[63] S. Chatterjea, L. F. W. van Hoesel and P. J. M. Havinga, "AI-LMAC: An Adaptive, Information-centric and Lightweight MAC Protocol for Wireless Sensor Networks," *2004 Proc. IEEE Intelligent Sensors, Sensor Networks and Information Processing Conf.*, pp. 381-388.

[64] V. Rajendran, J. J. Garcia-Luna-Aceves and K. Obraczka, "Energy-Efficient, Application-Aware Medium Access for Sensor Networks," *2005 Proc. IEEE International Mobile Ad-hoc and Sensor Systems Conf.*

[65] Bluetooth SIG, "Specification of the Bluetooth System," version 1.2, 2003. [Online]. Available: http://www.bluetooth.org. Last accessed October 2006.

[66] M. C. Vuran, Ö. B. Akan and I. F. Akyildiz, "Spatio-temporal correlation: theory and applications for wireless sensor networks," *Computer Networks*, vol. 45, no. 3, pp. 245-259, 2004.

[67] Z. Yang, M. Dong, L. Tong and B. M. Sadler, "MAC Protocols for Optimal Information Retrieval Pattern in Sensor Networks with Mobile Access," *EURASIP Journal on Wireless Communications and Networking*, vol. 2005, Issue 4, pp. 493-504, 2005.

[68] Anritsu, "Understanding Frequency Accuracy in Crystal Controlled Instruments," Application Note, March 2001. [Online]. Available: http://www.anritsu.co.uk/files/freq_accuracy.pdf. Last accessed October 2006.

[69] J. R. Vig and A. Ballato, *Ultrasonic Instruments and Devices*. Academic Press Inc., 1999, Ch. 7, Frequency Control Devices, pp. 637-701.

[70] J. R. Vig, "Quartz Crystal Resonators and Oscillators for Frequency Control and Timing Applications - A Tutorial," Rev. 8.5.2.3, April 2006. [Online]. Available: http://www.ieee-uffc.org/freqcontrol/tutorials/vig2/tutorial2.ppt. Last accessed October 2006.

[71] J. R. Clynch, "Precise Time and Time Interval Clocks, Time Frames and Frequency," February 2002. [Online]. Available: http://www.gmat.unsw.edu.au/snap/gps/clynch_pdfs/pttinote.pdf. Last accessed October 2006.

[72] D. A. Howe, D. W. Allan and J. A. Barnes, "Properties of Oscillator Signals and Measurement Methods," [Online]. Available: http://tf.nist.gov/phase/Properties/main.htm. Last accessed October 2006.

[73] Texas Instruments, MSP430x11x1 datasheet (Rev. H), September 2004, Available: http://www.ti.com. Last accessed October 2006.

[74] Texas Instruments, MSP430x1xx Family User's Guide (Rev. F), February 2006, Available: http://www.ti.com. Last accessed October 2006.

[75] T. R. Andel and A. Yasinsac, "On the credibility of manet simulations," *IEEE Computer Mag.*, vol. 39 no. 7, pp. 48-59, July 2006.

[76] M. Ali, T. Voigt, U. Saif, K. Romer, A. Dunkels, K. Langendoen, J. Polastre, Z. A. Uzmi, "Medium access control issues in sensor networks," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 2, pp. 33-36, April 2006.

[77] OMNeT++ Community Site, "Discussion of the various mobile/ad-hoc frameworks for OMNeT++," Available: www.omnetpp.org/pmwiki/index.php?n=Main.MobileAdhocFrameworks. Last accessed October 2006.

[78] J. G. Proakis and M. Salehi, *Communication Systems Engineering*. 2nd Ed. Upper Saddle River: Prentice Hall, 2002.

[79] M. Di Renzo, F. Graziosi, R. Minutolo, M. Montanari and F. Santucci, "The ultra-wide bandwidth outdoor channel: From measurement campaign to statistical modelling," *Mobile Networks and Applications*, vol. 11, no. 4, pp. 451-467, August 2006.

[80] A. Varga, OMNeT++ Discrete Event Simulation System User Manual, version 3.2, March 2005. [Online]. Available: http://www.omnetpp.org/doc/manual/usman.html. Last accessed October 2006.