A Power Saving Scheme for Application in Wireless Sensor Networks Predictive Transmission Success (PTS)

by

Christian Andrä

A Thesis
presented to the University of Strathclyde
in fulfilment of the requirements for the degree of Doctor of Philosophy
in
Department of Electronic and Electrical Engineering
Centre for Intelligent Dynamic Communications

January 2015

Declaration

This Thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

Acknowledgement

First of all, I would like to express my utmost thanks to Prof Ivan Andonovic. This Thesis would have never been there without his constant encouragement and his input. His constructive suggestions were incentive and motivation for me to find the strength to get things finished.

I would also like to thank Dr Craig Michie and Dr Robert Atkinson. Their permanent enquiry made me work at a high academic and analytical level.

Kwong, Swee for their encouragement and support to work at this high academic standard and for focusing me on the right contents.

Michael, Kostas and Christos for their support on technical, programming, and experimental design.

Also, Dr Ian Glover for offering the opportunity to getting involved in academic teaching. Thanks.

Für meine Eltern

Abstract

The research develops an approach to preserving energy in Wireless Sensor Networks (WSNs). Energy is a constrained resource, and thus increasing the lifetime of nodes to extend the flexibility of network deployments and ease maintenance is a core challenge; the network can operate unattended, autonomously and deliver applications for longer periods of time without human intervention. Energy is an especially limited resource for WSNs since invariably, nodes are battery powered and any scheme that extends the viability of the limited energy resource is much sought after. Inherent to the principles of WSNs is that each node is designed through a restricted set of resources and is equipped with the ability to gather, store, process and communicate data. In comparison to processing, transmitting/receiving data is very costly in terms of power consumption. Thus a viable strategy to conserve energy is to limit the amount energy owing to receiving and transmitting data through embedding intelligence within the protocol stack. The dynamic adjustment of the transmission power according to an application requested success rate, implemented through an extension of data embedded in the message packets is proposed and evaluated. Although a reduction in the transmission success rate results through lost packets, the level of energy savings is not compromised by an excessive increase in packet loss owing to a non-linear trade-off between the level of transmission power and transmission success. The dissertation introduces the motivation and background to the solution, defines the mathematical framework with which the approach to energy saving is founded and presents the emulation environment in which the performance of the solution has been evaluated. The expected energy savings owing to the utilisation of the scheme is presented and compared with techniques reported in the literature. Results show that notable energy saving is achievable with the proposed scheme.

Contents

Nomenclature

Latin Letters

E_b	Energy spent per bit	[J]	
e	Euler's Number		$\simeq 2.71828$
E[P]	Average packet payload size	[bits]	
G	Offered Traffic		
k_B	Boltzmann constant	$\left[\frac{\mathrm{W}}{\mathrm{kHz}}\right]$	$1.38 \cdot 10^{-23}$
N	Thermal Noise	[dBm]	
N_0	Channel noise	[dBm]	
N	Total number of devices in the network		
G_r	Receiver Antenna Gain	[dBm]	
G_t	Transmitter Antenna Gain	[dBm]	
L	System specific Loss Factor	[dBm]	
P_r	Received Signal Strength	[mW]	
P_t	Spent Power on Transmission	[mW]	
P_{Rx}	Received Power	[dBm]	
P_{Tx}	Transmission Power	[dBm]	
R	Square Side Length	[m]	
P_s	Successful Transmission		
P_{tr}	Probability of slot contains a Collision		
Q		[bits]	
S	Throughput	[E]	

T	Absolute Temperature	[K]
T_c	Average Time Channel is sensed busy due to collision	[s]
T_s	Average Time Channel is sensed busy due to successful transmission	[s]
T	Period	[s]
t	Time	[s]

Greek Letters

α	Path Loss Coefficient	
λ	Wavelength	[m]
σ	Slot Time	[s]

Subscripts

$_b$	per bit
$_{cf}$	Confidence
$_{cts}$	Clear to send message
$_{max}$	Maximum
$_{min}$	Minimum
$_r$	receiving
$_{rts}$	Ready to send message
$_{Rx}$	Reception
$_t$	transmitting
$_{Tx}$	Transmission

List of Figures

Listings

Chapter 1

Introduction

The research centres on the development of a power saving scheme that enables nodes within an extensive Wireless Sensor Network (WSN) to preserve energy and increase network lifetime [1]. The foundation of the strategy adopts the principle that since WSN nodes consume the most significant amount of energy on the transmission of data, a potentially viable approach is to develop a mechanism that preserves scarce energy resources by adjusting the level of power transmitted to an optimum for connections to next-hop neighbouring nodes [2].

1.1 Motivation

WSNs are an inherently a different class of network compared to classical wired or indeed established mobile networks [3]. The differences range from the fact they are data as opposed to connection centric, are based on distributed nodes equipped with scarce resources (power, memory and computational power) [1] and in the manner in which the nodes self-organise to implement the network.

A WSN is established, often not deterministically but randomly, by distributing nodes over a region of interest. On deployment, the network selforganises by discovering the neighbourhood, initiating a measurement task, data collection and forwarding over an extended period of time. To achieve this goal, unnecessary communication has to been kept to a minimum since it compromises one of the most constrained resources, the power supply. As an example, unnecessary communications are exchanges that send, retrieve or broadcast information that already exists within the network. Furthermore, data may be lost due to dynamic changes in the communication channel or movement of either the intended receiving or transmitting node. Mechanisms that manage the transmitting power to a minimum, meeting the probability of successful data reception so as not to compromise the application are potentially valuable solutions towards energy efficient operation.

Scenarios investigated comprise static and moving nodes e.g. sensors attached to monitor the health status of animals. Thus the network structure is dynamic and has to be operational for as long as possible. These boundary conditions impact the protocol design and the format of a range of message types.

The research will show that the scheme improves network lifetime when applied in tandem with existing access schemes under moderate load. Furthermore, when applied to a network experiencing high load, the proposed scheme outperforms others due to the increased frequency of updates inherent in its implementation.

1.2 Contributions

The main contribution of the research is the development and characterisation of a cross-layered extension to the communication stack that manages the power consumption for energy efficient WSN operation. Energy efficiency, and therefore lifetime, and reliable data transport are two of the primary functions within WSNs. Delivering data to the next hop neighbour without compromising on-going transmissions at optimised power consumption is the driver to the development of the solution.

Optimising the transmission power not only conserves energy and reduces the risk of perturbing ongoing transmission but also enables more extensive communication paths at the same time since less of the area of coverage is occupied by one transmission, in so doing increasing the data transmission rate per unit time and area. Therefore, the protocol can transport more information at any given time.

Intuitively, an increase in the number of active nodes participating in the relay of messages results in a concomitant increase in power consumption at the individual node level which then aggregates into a reduction of overall network lifetime. However in mobile node scenarios subject to dynamic changes in the channel characteristics, the ability to adjust the transmitting power level on a per packet basis to ensure successful data recovery between neighbouring nodes brings not only an overall power saving but addresses the problem of network partitioning owing to energy shortage of individual nodes. Partitioning [4] results in the creation of sub-networks unable to establish connections even though nodes are able to increase their transmission power to a maximum.

The dissertation develops performance evaluation frameworks embodying metrics that measure the improvement introduced by the proposed energy saving scheme. Simulation and analytical analyses are presented that express the benefits that accrue in terms of energy consumption and network lifetime which in turn facilitates WSN maintenance and in increasing data throughput per time unit and area and therefore spatial re-use.

Improvements in energy savings are evaluated through mathematical models and further corroborated through extensive analysis of the impact of the approach utilising simulations. The simulation environment is validated through comparison with published work and results are produced within representative network/application scenarios.

In order to manage ever increasing and differing demands on WSNs, an extension to the scheme is developed to allow nodes to classify and prioritise messages. This is central to the scheme since WSNs frequently exchange a mix of time and event triggered messages. Thus the option to decrease the transmission power for frequent reporting according to a pre-set level for the successful recovery of data is inherent; given the network knowledge acquired over the period of any application offers a trade-off between the energy savings achieved and the associated level of packet loss. For example, a number of approaches exist to compensate for packet loss by interpolating or extrapolating from acquired data at sinks that receive, manage and process.

A slowly changing application environment is assumed, such as found when measuring the temperature in a large volume of water or the condition of individual animals [5, 6]. The lifetime of this class of application is governed by the frequency of reporting. In such scenarios, the research shows that a significant saving in power consumption can be achieved without compromising the application.

1.3 Organisation of this Thesis

The dissertation is organised as follows. Chapter 2 presents the background to the research. An introduction to communication networks in general is given to accentuate the features and demands of wireless sensor networking including a description of physical channel characteristics. Movement is a distinct advantage of wireless and an introduction to methods that represent mobility patterns for the purpose of evaluation of the solution are also detailed.

Chapter 3 presents the state-of-the-art in the domain under investigation, in so doing highlighting the necessity for the research and the originality of the proposed solution. Based on this foundation, Chapter 4 details the challenges in the design of the network protocol together with the concepts and tools adopted in developing the implementation.

The core of the contribution, the power saving scheme – Predictable

Transmission Success (PTS) – is introduced, its principles defined and its design developed in Chapter 5. The mathematical background describing the scheme is presented and the potential scheme offers is shown. To validate the basis of the range of evaluations, the detail of the simulation environment is given. Results presented are obtained from both a C++ implementation of the emulated network and from a simulation platform, OMNeT++, specifically developed to treat wireless networks. A preliminary assessment of the impact of power consumption owing to PTS is executed using a C++ implementation and validation/enhancement of these results is undertaken through the OMNeT++ discrete event simulation platform. The evaluation not only considers the base protocol but also the impact of the hardware and channel. Both sets of results are presented, compared and discussed in Chapter 6.

The dissertation closes with conclusions drawn through a discussion of results and identifies future research directions in Chapter 7. Although a number of other enhancements that potentially improve the solution are highlighted, due to restrictions in time only the most significant results are presented.

Chapter 2

Network Standards, Definitions and State of the Art

The Chapter presents an overview of protocols currently in use within Wireless Sensor Networks (WSNs). The requirements of the Medium Access Control layer (MAC) layer [7] are defined, the Open System Intercommunication (OSI) standard seven layer framework [8] presented and the function of each layer described. Furthermore, an in depth description of the Physical Layer (PL), Routing and Transport Layers is given along with their concomitant implementation limitations. The Aloha [9] and Carrier Sense Multiple Access/Collision Avoidance (Csma/CA) [10] protocols are discussed in respect of their transmission capabilities and drawbacks through theory which in turn forms the basis for the evaluation of protocol performance presented in Chapter 4 where these protocols are implemented and validated.

2.1 Introduction

The nature of a wireless network of sensing nodes informs the protocol design process since the constituent elements are equipped with limited resources such as power, memory and computational capabilities. The topology is likely to change frequently and it is often the case that the node in not identified globally which in turn prompts the formation of cluster groups. To secure the maximum advantage of such networks fundamentally rests on the implementation of reliable transport protocols; such advantages are a large coverage area, high accuracy, and fusing or aggregating data through self-organisation [11]. The design of a transport layer protocol is also heavily influenced by the type of application it serves. Some applications might require a higher reliability on data/packet delivery; others require constant or event driven only reporting. To gain a better understanding of the network hierarchy, Section 2.1.1 presents an overview of the layered structure of protocol implementations, as defined in International Organisation for Standardisation (ISO) standards [12].

2.1.1 The Open System Intercommunication layered Node Model

The ISO standard for networking was first released in 1979 when the "TC97" committee began the development of standard protocols for OSI [8]. This standard was aimed at providing a common set of rules which enabled different systems to communicate with each other [13]. The main characteristic of the standard is its layered structure allowing single layers to be updated without the need for re-implementing the whole standard. An example of the elegance of the structure is that applications were defined and managed through the design of a certain layer, offering the same service and interfaces but with additional features. Each layer, referred to as (N^{th}) layer, is allocated responsibilities and releases higher layers, referred to as ($(N+1)^{th}$) layers [8] from such functionalities. Furthermore, layer N provides a service which the higher layer $(N+1)^{th}$ makes use of. Thus the (N^{th}) layer can be re-implemented whilst still providing the same service to the ($(N+1)^{th}$) layer. The OSI is often referred to as the 'Seven Layer Model', with all seven layers implementing fully functional network elements. The following is an overview of all the layers and their functionality.

Transport Layer; provides transparent data transfer between session entities; relieving such entities from any concern of the reliability of the data transfer.

Network Layer; establishes, maintains, and terminates network connections and exchanges network service data units between transport entities. It also takes care of routing and relaying function in order to transmit data across sub-networks.

Data Link Layer; provides the functional and procedural means to establish, maintain, and release data-link connections to the one-hop-neighbour.

Physical Layer; provides the mechanical, electrical, functional, and procedural means to activate, maintain, and deactivate the physical link between nodes/systems.

For the sake of completeness, the remaining layers are the Presentation Layer on top of the Session Layer servicing the Application Layer sitting on top of the ISO OSI seven layer model.

To better understand the operating mode of a layer (N^{th}) requires consideration of the underlying layers ($(N-x)^{th}$). In the case of the transport layer, the physical, data link and network layers must be considered before investigating transport layer functionalities and services. The PL, responsible for accessing the physical medium whether it is wired or wireless, needs to represent the data to be transmitted with the correct modulation scheme (voltage level or phase-shift are just two examples). Sending, receiving, demodulation, and timing is also managed by the PL. Often the PL is isolated from the wired medium to protect the device from faults that might occur on the transmission medium. On top of the PL is the MAC. The MAC, or Data Link Layer, controls the data exchange on the physical medium. According to [14] the OSI model consists of eight layers when dividing the Data Link

Layer in the MAC and the Logical Link Control as defined by the IEEE 802 Committee. At the receiver, the MAC identifies changes in power level, checks for the correct start of the transmission and the start of buffering the data intended for an end point. At the transmitter, the frame needs to start with a pre-defined/standardised sequence of bits acting as a trigger to announce the start; the end of a frame must also be identified by the sender in order to be recognised by the receiver. The pre-amble is used to synchronise the clock of the sender of a message and the intended receiver(s) [15]. Furthermore it needs to manage damaged, lost, and duplicated frames. A very important function is the negotiation of the transmission rate since the transmitter and the receiver maybe clocked at different speeds e.g. a faster transmitter may simply drown a slower receiver [10]. The organisation of the communication within a large network is the role of the Network Layer through maintaining subnets. Its main responsibilities are route discovery and addressing. Routes are allocated via a routing table, sent to each user on joining the network; therefore it is static. A route is generated in each connection set-up, or more frequently on each frame to be sent, allowing traffic distribution according to the load viz. load-balancing [16]. Networks exits that do not need a Network Layer [17]; such networks operate in broadcast mode, in which all messages are sent to every receiver. The receiver decides if the packet is meant for it; otherwise it is discarded. Finally, the role of the Transport Layer is to ensure that the data is recovered by the intended recipient. Section 2.2, Section 2.3, Section 2.4, and Section 2.7 present each layer in more detail, defining their tasks and responsibilities.

2.2 Physical Layer

The Physical Layer [18] is responsible for accessing the communication medium; in the case of WSNs, this medium is the wireless channel. The PL decides the channel to be used and the frequency of the transmission. The channel is usually segmented into bands and intuitively the wider the band, the more connections can be established at a time. The PL is also responsible for signal detection, modulation and encryption of the data packets. Commonly the nodes that form a WSN use the Industrial, Scientific and Medical (ISM) frequency bands [19]; as well as the ISM band, Infra Red (IR) is also a license free band [18]. The main drawback of IR communication is the requirement for a Line of Sight (LoS), rendering the approach less feasible for WSNs since LoS cannot always be guaranteed and is highly dependent on the application.

In general, transmissions in WSNs are expensive in terms of power consumption [20]. The power spent on transmitting a message is proportional to the distance

$$P_{Tx} \propto d^{\alpha} \tag{2.1}$$

with $\alpha \in (2, 4)$ and where P_{Tx} is the transmission power, d is the distance between sender and receiver and α is referred to as the path loss coefficient [21]. α is closer to four in near ground communications due to signal cancellation through interference with ground-reflected waves and for indoor environments, α can reach 6.5 [22]. The significant amount of energy consumed on data transmission presents a massive design and implementation challenge and the research in focussed on solutions and approaches that overcome this challenge and details the development of a scheme that can be used in a tandem with existing protocols. The goal is to extend existing protocols through a power-saving overlay scheme independent of the environment and thus save energy according to message classification.

2.3 Medium Access Control Layer

The MAC is the layer directly on top of the PL, managing access to the communication medium. Data from upper layers are encapsulated with a

MAC header. The MAC is also in control of the power level set for each transmission and for the selection of the channel. A range of techniques have been developed as an effective means of accessing the channel [23] e.g. Aloha [24]. In general a sensing phase is instigated prior to a node being permitted access [10]. Section 2.3.1 and Section 2.3.2 introduce Aloha and Csma/CA protocols, examples of the earliest and two of the most important and relevant MAC protocols. The background to the protocols is presented as they are used later for validating the implemented simulation models (Chapter 4).

2.3.1 The Aloha Protocol

The Aloha protocol is one of the earliest wireless communication protocols developed by the University of Hawaii in 1970 [24]. It operates a single hop principle in which every node is able to transmit data packets to a central server; thus no channel access scheme exists. A fixed packet size and no scheduling of transmission times are used. Therefore, a node working in pure Aloha mode attempts to send its packet and waits for the server to acknowledge (ACK packet) its reception. If the ACK remains absent the node assumes the message has been lost and will resend the packet. After a certain number of retries, the packet is dropped. The transmission time for each packet is;

$$t_{\text{trans}} = \frac{\text{packetsize}}{\text{bitrate}} \tag{2.2}$$

Since the transmission fails if two packets collide, the vulnerable time of a packet is;

$$t_{\text{vul}} = 2 \cdot t_{\text{trans}} \tag{2.3}$$

Therefore, the Aloha throughput is given by [24];

$$S = G \cdot e^{-2G} \tag{2.4}$$

Differentiating equation (2.4) in respect to G twice yields to;

$$S' = \frac{dS}{dG} = (1 - 2G)\, e^{-2G} \tag{2.5}$$

$$S'' = \frac{d^2S}{dG^2} = (-4 + 4G)\, e^{-2G} \tag{2.6}$$

Equating Equation (2.5) with zero and solving the equation, returns an extremum at $G = \frac{1}{2}$. Solving Equation (2.6) returns;

$$S''\left(\frac{1}{2}\right) = \frac{-2}{e} < 0 \tag{2.7}$$

and is therefore a maximum. Substituting $G = \frac{1}{2}$ in Equation (2.4), returns the maximum throughput of pure Aloha to be;

$$S\left(\frac{1}{2}\right) = \frac{1}{2e} \approx 0.184 \tag{2.8}$$

. Simulations were carried out to corroborate that the implemented node MAC is valid and functioning according to the above values (Section 4.3).

As an improvement to this protocol, the channel can be divided into time slots, the exact same length as a packet and can be calculated to be of the length;

$$t_{\text{slot}} = \frac{\text{packet\_size}}{\text{bitrate}} \tag{2.9}$$

Every time a node has a data packet to send, it needs to wait until the next slot starts. Transmissions are only allowed to start at the beginning of a slot. This reduces the vulnerable time of a packet to;

$$t_{\text{vul}} = t_{\text{trans}} = t_{\text{slot}} \tag{2.10}$$

A packet is corrupted only if two or more nodes decide to transmit in the same slot. Therefore, the throughput of Slotted Aloha [25] protocols is calculated to be;

$$S_{\text{slotted}} = G \cdot e^{-G} \tag{2.11}$$

Differentiating Equation (2.11) twice in respect to G leads to;

$$S'_{\text{slotted}} = \frac{dS}{dG} = e^{-G} - G \cdot e^{-G} = (1 - G)\, e^{-G} \tag{2.12}$$

$$S''_{\text{slotted}} = \frac{d^2S}{dG^2} = -e^{-G} + (1 - G)e^{-G} = (G - 2)\, e^{-G} \tag{2.13}$$

Equating Equation (2.12) with zero and solving the equation, returns an extremum at $G = 1$. Solving Equation (2.13) at $G = 1$ returns;

$$S''_{\text{slotted}}(1) = -e^{-1} < 0 \tag{2.14}$$

a maximum. The maximum throughput of the slotted Aloha protocol can be calculated to be;

$$S_{\text{slotted}}(1) = \frac{1}{e} \approx 0.368 \tag{2.15}$$

where

S = Throughput
G = Normalised Channel Traffic
e = Eulers Number
$\frac{d}{dG}$ = Differential Operator

In Chapter 4 an introduction to the simulation framework is presented and the node model is verified using well-known protocols viz. Aloha and Slotted Aloha. Furthermore, Csma/CA is a well recorded MAC scheme [23]; the detail underpinning Csma/CA is presented in Section 2.3.2 and simulations of its operation are presented in Section 4.2.

2.3.2 Carrier Sense Multiple Access/Collision Avoidance

CSMA/CA is an advancement of the Carrier Sense Multiple Access (CSMA) and Carrier Sense Multiple Access/Collision Detection (CSMA/CD) protocols [26]. The CSMA protocol and all its advancements address a set of problems that arise when multiple devices need to access the same channel. Whereas full-duplex supports ongoing communications in both directions simultaneously, on a half-duplex channel communication can be established in both directions but only at different times. A wireless channel most commonly only supports half-duplex [27].

One of the key advantages of devices able to connect through a wireless channel is mobility within their transmission range. Furthermore, devices can join and leave the network or be handed over to another access point without the knowledge of the user. Since multiple devices routinely share one communication medium (Master Aggregations), it is desirable that ongoing transmissions are not interrupted. In order to achieve this, every node must sense the channel (Carrier Sense) before it can send its data. In principle this is the scheme all nodes need to follow to participate in a CSMA enabled network.

In a Carrier Sense (CS) scheme [28] the core challenge is to determine when a channel is busy, when a transmission is ongoing and when it is idle. The implementation relies on defining a *busy* Received Signal Strength Indication (RSSI) [29]. This value needs to selected carefully, set between the thermal noise level that every channel is characterised by and the level of signal transmission strength required to successfully recover a message at the node. A typical noise level is as calculated in Equation (3.15) is $N_{\text{thermal}} = -141$ dBm [30] and the threshold to recover a message successfully is mostly dependent on the receiver sensitivity [29, 31, 32]. The threshold corresponds to the minimum signal level at a receiver that can be recovered successfully, commonly -98 dBm and -90 dBm [33, 34] i.e. the receiver in this particular case is capable of recovering signals with a RSSI of at least

−96 dBm.

In an ideal scenario, the RSSI level can be considered as indicative of a 'busy' channel; thus in CSMA/CA, all signals detected above the threshold represent ongoing transmissions. Since CSMA/CA is a slotted protocol, the nodes 'freeze' their counters if they are in sensing mode during ongoing transmissions; otherwise packet collisions result since two or more nodes choose the same initial interval or start their transmission at the same slot. Furthermore, a wireless device has no means of listening to its own transmission and hence relies on Acknowledgement (ACK) packets [35]. After the packet is transmitted, the node reverts to receiving mode and a time-out timer is initiated with the set reception period of the ACK. If the ACK is not received within that time, a re-transmission of the packet is scheduled. A node receiving an intended packet immediately generates an ACK, senses the channel for Short Inter-Frame Space (SIFS) seconds and sends the packet; SIFS is one the four Interframe Spaces (IFSs) defined as a time interval between two consecutive frames [35] (Figure 2.1). The short time interval of a SIFS restricts the idle time for other nodes that need to sense and access the channel, therefore delaying the sending of an ACK. Since SIFS is much shorter than Distributed Inter-Frame Space (DIFS) – in the simulation carried out and presented in Section 4.3, it is set to be 28 µs compared to the DIFS setting of 128 µs [36]) – still more than 50% of a slot (50 µs) is required to detect on-going transmissions. The $\mathrm{ACK}_{\mathrm{time\_out}}$ can be calculated as [37];

$$\frac{\mathrm{ACK}_{\mathrm{packet\_length}}}{\mathrm{bitrate}} + \mathrm{SIFS} = 268\mu\mathrm{s} \tag{2.16}$$

rounded to 300 µs.

If a node fails to access the channel after the initial back off, it will increase the back off time exponentially, decreasing the chance of another node choosing the same back-off thereby creating collisions. The exponent is chosen from randomly distributed integer values. If the packet does not access the channel after the sixth attempt, the node drops the packet.

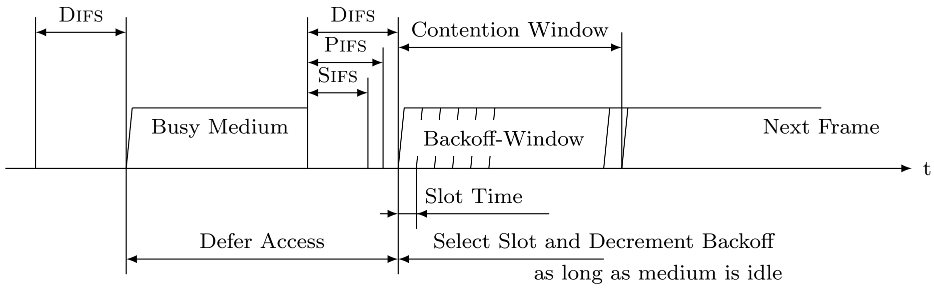


Figure 2.1: IFS relationships [35]

2.3.3 Problems while Channel Access

Two serious challenges arise in channel access with WSNs implementations; hidden and exposed terminal scenarios [38]. In the hidden node scenario, (Figure 2.2(a)), node A is connected to node B at the same time as node C is connected to node B. Since node C is not aware of an on-going transmission between node A and node B, both transmissions collide and neither node A nor node C receives an ACK.

The exposed node problem involves one more node as shown in Figure 2.2(b). In this scenario node C is prevented from transmitting to node D when node B is transmitting to node A. C senses the channel to be busy even though a transmission from C to D does not interfere with the on-going transmission between A to B since node C is out of the transmission range of node A.

Both problems can be overcome through use of the Ready to Send/Clear to Send (RTS/CTS) scheme also defined within the CSMA/CA protocol [38]. In case of the exposed node, the RTS/CTS scheme prevents either of the remote nodes starting a transmission because it is able to overhear the Clear to Send (CTS) message sent by the otherwise exposed node B. In the case of the hidden node, a transmission between node C and node D can still be carried out as C senses the CTS message from node B only but is not able to sense the Ready to Send (RTS) message sent by node A. This handshaking

(a) Hidden Node (b) Exposed Node

Figure 2.2: Schematic drawing of the hidden and exposed Node problem

results in a further decrease in the number of colliding packets, but at the cost of an additional overhead which in turn reduces throughput of useful data.

2.4 The Routing Layer

The MAC enables a node to send messages to neighbouring nodes but if the network grows in extent, the node will lose full network connectivity as not every node remains within the transmission range of any other node. Even though a particular node can successfully connect to another within its transmission range, it does not follow that the receiving node is able to reply, since its transmission range might be shorter. This scenario is characteristic of a non-symmetric network [39].

The Routing Layer is used to overcome such issues, taking care of forwarding messages from source to destination through a multi-hop route. In scenarios such as described in Section 5.5, forwarding through multiple neighbours is necessary. Furthermore, it is the role of the Transport Layer to distribute traffic over different routes in order not to overload one route while leaving others idle [10]. These routes are either available to the nodes through static, pre-programmed tables or are generated on demand [40]. With static routes, the network is prone to transmission failures since node links are more likely to fail; also with static routing tables, route recovery is not possible. In contrast, on demand route establishment consumes more energy since nodes need to negotiate the route which results in an increased communication overhead. Routing protocols can be categorised according to the following types [41];

- Flat [42]
- Hierarchical [43]
- Geographical assisted [44]
- Power-aware [45]
- Security-aware [46]
- Efficient flooding [47]
- Multicast [48]

A flat routing scheme seems best aligned to the needs of an extensive WSN; there is no need to be organised in groups or clusters, like in a hierarchical routing scheme, nor do the nodes need to be aware of their position, as would be a requirement of geographical assisted routing schemes. Energy awareness and security are issues that are beginning to be demanded from evolving WSN applications e.g. in health related environments [49]. In medical applications, patient data is considered personal and steps must be implemented that robustly protect that information. Furthermore, node failure as a consequence of power shortages is to be avoided for quality of service requirements. Medical wireless networks are mostly single hop environments due to the high resilience demands that are placed on deployments and efficient flooding broadcasts the data across recipients. Many nodes execute forwarding and a lot of redundant data is generated and transmitted. Other schemes limit the number of nodes involved in the routing path [50, 51, 52].

Table 2.1: Connection-oriented routing versus connectionless routing

Connection-oriented	Connectionless
Setup needed	No setup needed
Packet contains Virtual Circuit (VC)-number	Packet contains full address
Packet routed according to VC-number	Packet routed according to address
On failure a new route will be established	All VC on failed node will be terminated
Congestion control is easy, if enough buffers are allocated in advance	Congestion control is difficult

Assuming a network in a first response environment has the following properties:

- Nodes are randomly distributed
- Nodes are not aware of their position
- Nodes do not know their neighbourhood
- Multiple base stations
- Moving base stations

If such a network forms an infrastructure platform for the exchange of data, the time taken for the nodes to organise is prohibitive since a significant period is required to negotiate the structure. It is also desirable to begin transmitting data immediately. This then infers a routing scheme not reliant on a specific network infrastructure. Table 2.1 presents the characteristics of connection-oriented and connectionless routing schemes.

A detailed survey of routing protocols used in WSNs is given in Section 2.5 where the protocols are introduced, classified and an overview of available solutions is given.

2.5 Protocol Overview

Depending on the underlying network structure, routing protocols can be classified as either hierarchical or location based [53]. Furthermore, these protocols can be further segmented into multi-path [54], query-based [55], negotiation-based [56], Quality of Service (QoS)-based [57], and coherent-based [53], terms defining the principles of protocol operation. A large number of sensors allow for a more granular mapping of the environment and a large number of nodes deployed randomly need to coordinate amongst themselves in order to transport high quality data.

Each sensor needs the capability to route data through the network. The node can either forward the message to another node or transport it to the base station. As a consequence of the densely deployed number of nodes, addressing cannot be utilised as in other networks [58]; the addressing overhead would result in gross inefficiencies in the amount of data delivered. Deploying nodes randomly requires self-organisation to form a network irrespective of distribution. To minimise energy consumption, the routing approaches adopt strategies specific to WSNs e.g. data aggregation [59], network processing [60], clustering [61], different node role assignment [62], and data centric methods [53].

The challenges can be summarised as:

Node deployment; a major influence on the performance of the routing protocol. Nodes can either be deployed randomly e.g. within a disaster area recovery [63], or deterministic, deployed by humans. If the nodes are placed in planned positions across the environment, data can be routed through given, pre-determined paths. If the nodes are deployed randomly, the network infrastructure needs to be established in an ad-hoc manner. Randomly deploying nodes usually results in a non-homogeneously distributed node density. In order to form a network infrastructure, nodes may need to cluster. Cluster design becomes crucial and needs to be optimised with the predominant goal of prolonging network lifetime [64]. As a consequence of the short transmission ranges between inter-node communications, routing paths are most likely multi-hop in nature [65].

Lifetime versus accuracy; in most applications, nodes are deployed to perform a measuring/monitoring task comprising data acquisition, processing and storage. However, unless the node can establish connections, the generated data has no value and viable routing strategies are a core consideration. Most often, the recipient of the data is not reachable within a single hop which requires nodes to forward the acquired data to other nodes. Thus energy-efficient ways of transmitting data is essential. Unnecessary drain of the limited energy resource can cause huge perturbations in network operation, due to compromised forward/routing nodes. Therefore, packages may need to be re-routed and the network reorganised.

Data reporting model; the sensing and data reporting policy in a WSN is highly dependent on the application e.g. some application are time critical [66]. Two modes can be identified; nodes reporting measurements to the base station or the base station requests data reports from one node, nodes within a certain area (cluster) or all nodes [67]. A combination of the two modes can also be desirable. The data reporting modes can be further classified as frequent and event driven. In the latter mode, nodes only transmit measurements on significant changes in the environment or when a threshold is exceeded. These reporting modes place very different requirements on the routing layer and in turn impact on the power consumption and lifetime. Nodes that operate in an event reporting or query mode are invariably in active state even if there is no data to be sent in case a query or an event needs to be forwarded. Nodes in the low power listening mode turn on and off their radio periodically to take a physical reading of the channel [68]. Only an event at the node can be recognised by that node. But if the node does not listen to the transmission on the channel (which drains energy and therefore shortens lifetime) it is not available for forwarding, and cannot receive data queries from a neighbour or a base station.

Link/Node Heterogeneity; sensor nodes usually support different capabilities viz. differences may be computational power, the energy resource, sensing or transmitting capabilities. Furthermore, different amounts of data can be generated by each node. Therefore node capabilities are another factor in application implementation; the goal is to utilise whatever resources to the maximum.

Fault Tolerance; a number of node failure modes can be identified [69] e.g. power drain. Failure of nodes should not affect network functionality. Connections need to be managed through a new path when nodes drop out of the network [70]. The new path might impose different requirements on the transmission power or path length. Load-balancing is a valid approach to treating node failure [71].

Scalability; in many advanced WSN applications, the network comprises in excess of hundreds or thousands nodes. Therefore, it is desirable for the routing protocol to be adaptable. In case of an event reporting mode, a large percentage of nodes can be placed into a sleep state until an event occurs.

Network Dynamics; in the case when either the sink or nodes are mobile, the routing becomes more challenging than within a static networks [72]. The main challenge is routing stability. Compared to static reporting, reporting dynamic events like object motion, needs frequent, periodic reporting and therefore generate significantly more traffic.

Transmission Medium; traditional problems associated with a wireless channel, e.g. fading, also plague the performance of a WSN [73], although the required bandwidth of around 1 kbs^{-1} to 100 kbs^{-1} in WSNs is comparatively low compared to wired networks. Various protocols have been reported that manage energy consumption efficiently [74]. For example, since no channel contention occurs in Time Division Multiple Access (TDMA) once a time slot is assigned, the approach offers lower energy consumption compared to a contention based protocol such as CSMA [75].

Connectivity; within networks with a high number of nodes per area i.e. high density deployments, nodes are less likely to be isolated.

Coverage; an important design issue owing to the limited coverage area of each sensor node; a small physical area is covered with restricted accuracy.

Data Aggregation; in order to limit the number of transmissions, data can be aggregated if the routing node manages the aggregation of data destined to the same recipient [76]. The routing node can combine incoming data in different ways; e.g. duplicate suppression, calculating minima, maxima or the average value [76].

Quality of Service; for some applications it is necessary to transfer data to a sink within a certain period of time; otherwise the data is rendered useless. Therefore, bounded latency is another condition for time-constrained applications [77]. The lifetime of the network, strongly related to the lifetime of single nodes, is often more important than the quality of the data [78]. Network and routing protocols may react to the critical drain energy conditions to lengthen network lifetime [79].

2.6 Routing Protocols

Routing protocols can be segmented into flat, hierarchical and location based [80]. In flat-based routing all nodes carry out the same role in comparison to hierarchical-based routing where different nodes carry out different roles [81]. In location based routing, the location of the node is exploited to route data through the network. Routing protocols are considered adaptive in order to manage traffic loads or available power levels. Furthermore, routing protocols can be classified into multi-path-based, query-based, negotiation-based, QoS-based, or coherent-based techniques depending on protocol operation [82, 56, 83]. In addition to this classification, routing protocols can be further divided into reactive, proactive and hybrid, depending on the data-path between source to destination [84]. In proactive routing protocols all routes are computed before needed [85]; in reactive protocols routes are computed just in time [86]; and hybrid uses a combination of both [87].

2.6.1 Flat Routing

In a flat network, nodes co-operate to perform the task. All play the same role while routing packets through the network. In this so-called data-centric routing, the sink sends queries to certain areas of the network and waits for the data from nodes in the region. Sensor Protocol for Information via Negotiation (Sensor Protocol for Information via Negotiation (SPIN)) [56] is an example of this type of protocol, treating every node in the network as a potential sink and broadcasting its information to every other node. Under certain latency restrictions, each node receives the same data, allowing the user to poll any node for information from anywhere in the network. SPIN assumes that nodes in the near neighbourhood have the same data available, so exchanges with neighbours updates the data. Nodes running this protocol use metadata [88] to describe their data. Before any data is transmitted, metadata negotiation is carried out, ensuring no redundant information is exchanged throughout the network. Metadata negotiation is not specific to Spin which adapts its operation to the current energy level. These protocols are time-driven and distribute data throughout the network, even if a neighbour does not request any data. Since Spin also adapts to the power level of the node and performs data negotiation, it overcomes some of the problems of the classic flooding approach [89]. The protocol is based on fundamental principles:

1. energy efficiency since the node only transmits changes in the data, rather than transmitting already known data.

2. flooding or gossiping routing protocols [89] are wasteful of energy since all information is broadcast throughput the network, creating unnecessary or extra copies of the data.

Spin is a three stage protocol and utilises three types of messages: Advertisement message (ADV), Request message (REQ), and Data message (Data) [56]. After a node has completed its sensing task and derived the metadata, it advertises its new data through ADV. Any neighbour interested in the new data replies with. a request for data (REQ). Data transfer then takes places. The receiving node repeats the procedure with its neighbours until the entire network has the new information. Spin offers a family of protocols, including Spin-1 and Spin−2 [90]; both consider power consumption as a core factor in its operation. Other protocols in the Spin family [91] are;

1. SPIN-BC, for broadcast channels

2. SPIN-PP, for hop-by-hop communication

3. SPIN-EC, similar to the PP protocols with an added energy heuristic

4. SPIN-RL, adjusts the Spin-PP to treat a lossy channel

Although the Spin protocol is much more energy efficient than the classic flooding protocols, the delivery of data is not guaranteed.

Direct Diffusion (DD) [50] is driven by date centric and application-aware paradigms. The main goal of DD is to combine data from different sources to reduce redundancy and therefore minimise transmissions throughout the network. A reduction in traffic in turn results in energy savings. Unlike traditional end-to-end routing protocols, DD identifies routes from multiple sources to one destination which allows in-network consolidation of redundant data. If the sink requires data, it transmits a broadcast of interest throughout the network. This interest diffuses through the network hop-by-hop and is broadcast by every node to its neighbours. An interest is best viewed as a task to be done by the network. The sensing nodes generate a gradient of information in their respective neighbourhood [50].

After the broadcast, a gradient is established to queue the requested data back to the requesting node (sink). As the request diffuses through the network, every node sets up a gradient towards the node it received the request from. Loops cannot be excluded at this stage. When the sink starts receiving data, it resends a request frequently to manage data which have not been received reliably. All sensor nodes in a diffusion-based network are application-aware, which enables diffusion to achieve energy savings by selecting empirically good paths, by caching and processing data in the network. Caching can increase the efficiency, robustness and scalability of coordination between sensor nodes, the essences of data diffusion paradigm. Another usage of DD is to spontaneously propagate an important event to some areas of the network. Such type of information retrieval is well suited only for persistent queries where a requesting node is expecting data to be satisfied within a specified duration of time. Thus, it is unsuitable for one-time queries, as it is not efficient to establish a gradient for queries which use the path only once.

To investigate network behaviour, two models of source placement have been studied; the event radius model (ER) [92] and the random source model (RS). In ER, a single point is assumed to be the origin of an event. All its neighbours within the distance S (so called sensing area) that are not base stations are considered to be data sources. In RS, k of nodes that are not base stations are randomly selected to be sources. Unlike ER, sources are not necessarily clustered near each other meaning a greater number of nodes, for a given energy-budget, can be connected to the sink. Depending on the application, each one performs better than the other, in respect of energy consumption. DD is different to SPIN in two features; firstly, DD issues on-demand queries as the sink sends requests to nodes by flooding the network. In SPIN, nodes advertise their information and a willing node, a potential sink, can request the data by polling. Second, all communications in DD is neighbour-to-neighbour, with each node having the capability to perform aggregation and caching. DD is thus not suitable for applications where data are needed periodically, since the sink needs to request data every time.

Rumour Routing (RR) [93] is a variation of DD where geographic routing is not feasible. Under certain scenarios, the flooding of the network with a request is unnecessary since a limited amount of data is required. RR routes the query to nodes where a certain event occurred rather than flooding the entire network with the query. In order to flood events through the network, RR employs long life packets, referred to as agents. Nodes detecting a certain event create an entry in their local table and a corresponding agent. The agent is then propagated through the network and a node interested in that data can create an agent to request the data. The node, after discovering the route to the source node through the inspection of its local event table, responds on the request. Consequently the whole network does not need to be flooded with the request for information, reducing transmissions and therefore energy consumption. Nodes need to maintain less path data compared to DD, where nodes need to maintain multiple paths. RR reduces energy consumption compared to event flooding and managing node failures. A drawback is that it can only manage a limited amount of events. For a large number of events, maintaining agents and event tables becomes infeasible especially if insufficient interest in some data is weighted by the base station.

Minimum cost forwarding algorithm (MCFA) [94] exploits the fact that the direction of routing is always known in the case of an external fixed data sink. Hence, a node does not need to have a unique ID nor does it need to maintain a routing table. Instead, every node maintains its own estimated least cost routing path to the sink. If a node, that is not the base station, receives a message it checks if it is on the least cost path between the source node and the sink; if it is, it forwards the message to its neighbours. Each neighbour carries out the same check and the one on the least cost path forwards the message onwards. This procedure is repeated until the sink is reached.

Discovery of the least cost path between the source node and base station is obtained as follows; the sink broadcasts a message with the cost set to zero. Every node in the network initiates its least cost value to infinity and on receiving the broadcast message from the base station, checks to see if the estimate in the message plus the link on which it received the message is less than the current estimate. If that is the case, the current estimate and the estimate in the broadcast are updated. If the broadcast message was updated by the node it will resend the broadcast; if it did not update the estimate, nothing else happens. To avoid nodes further away from the sink receiving a flood of updates, MCFA was extended to incorporate a wait time $a \cdot l_c$ before resending the broadcast, where l_c is the link at the node receiving the message and a is a constant time.

Gradient Based Routing (GBR) [95] is another DD protocol based on recording the number of hops when a query is diffused throughout the network. Each node checks the 'height of the node', representing the minimum number of hops to reach the base station. The difference between a node's height and that of its neighbour is considered the gradient on the link. Therefore, a message is forwarded on the largest gradient. GBR uses some auxiliary techniques such as data aggregation and traffic balancing in order to uniformly divide the load over the network. If more than one route leads through the same node, that node may aggregate and combine the data according to a certain function. There are three schemes in which the data can be disseminated:

1. Stochastic: if there are two or more hops on the same gradient, one is selected randomly

2. Energy-based: if the energy resource of a node falls below a certain threshold, its height is increased in order to limit the number of messages received

3. Stream based: new traffic streams are not routed through nodes that are already part of other paths.

The main goal is to increase network lifetime by balancing traffic over the entire network.

Information Driven Sensor Querying (IDSQ) and Constrained Anisotropic Diffusion Routing (CADR) [96] are general forms of DD. The key aim of CADR is to query nodes and route data through the network such that the information gain is maximised while latency and bandwidth are minimised. Therefore CADR uses a set of criteria to decide the data from which node is made available by switching on sensors that are close to a certain event and dynamically adjusting the routing path.

The main difference from DD is consideration of the information gain in addition to the communication cost. Each node in CADR evaluates an information/cost objective on its local data and end user requirements; estimation theory is used to model information. In IDSQ, the queuing node can determine the data providing node with the advantage of balancing the energy. The routing path between sink and the nodes is not specifically defined by IDSQ and can be viewed as a complementary optimisation procedure.

COUGAR [97] maps the entire network into a huge distributed database in order to abstract query processing from the network layer functions and uses in-network data aggregation to obtain additional energy savings. The abstraction is achieved by an additional layer between the network and application layer. In COUGAR, nodes select a head that receives all data, aggregates and forwards it to the sink. The base station is responsible for generating a query plan which specifies the necessary information on data flow and in-network computation for incoming queries and disseminates it to all relevant nodes. The query plan also designates the leader for the query. COUGAR provides a layer independent method for data queries but with some drawbacks. First, an additional layer is needed consuming additional memory and generating additional overheads. Second, synchronisation amongst nodes is required to execute in-network computation. Third, the leader should be dynamically maintained to prevent hotspots.

Active QUery in sensoR nEtworks (AQUIRE) [98]], like COUGAR, maps the network in a distributed database where complex queries can be further divided into sub queries. In AQUIRE, the sink sends a query to nodes it can reach. All nodes forward the query; each node attempts to answer the query by using its own pre-cached information and then forwards it to other nodes. If the information does not satisfy the query, the nodes gather new information from neighbours with a look-ahead of n hops. If the query is solved, it is sent back to the sink either along the reverse or shortest paths. Hence, AQUIRE handles complex queries by allowing multiple nodes to send back responses. Note that Direct Diffusion may not be used for complex queries due to energy consumption issues. AQUIRE can provide efficient querying by adjusting the look-ahead parameter n. When n is equal to the network diameter, AQUIRE behaves similar to flooding; if the parameter is too small, the query needs to travel more hops.

Energy Aware Routing (EAR) [99], the target is to increase network lifetime by destination initiated diffusion. Although also similar to Direct Diffusion, the difference is that EAR maintains a set of paths rather than maintaining or enforcing the optimal path. These paths are maintained and chosen by means of a certain probability. The paths are chosen on how low the energy consumption is on each path. Therefore, the energy of single path does not deplete faster and thus the energy consumption is more evenly distributed on the network. Each node needs to be addressable through class-based addressing which includes the location and type of the nodes. During the initiation phase, the network is flooded to localise all nodes and determine the paths between each source/destination pair and the related costs. High-cost paths are discarded and a forwarding table is built by choosing neighbouring nodes in a manner proportional to their cost. Forwarding tables are then used to send data to the destination with a probability that is inversely proportional to their cost. Localised flooding is performed to keep the paths alive.

Routing Protocols with Random Walks [100] are used to distribute the load statistically over the network by using multi-path routing, only for large scale networks where nodes have limited mobility. It is assumed that nodes can be turned off and on at random times. Furthermore, each node is identified unambiguously and no location information is needed. Every node is placed on a crossing point of a regular grid on a plane. Route establishment is based on the Bellmann-Ford algorithm [101], considering location information or lattice coordination.

Dynamic Source Routing [102] is a routing protocol most suitable for static network and slow moving nodes. Dynamic Source Routing (DSR) is separated into two phases; route discovery, which takes place every a node does not have a valid route to its destination node, and the route maintenance, where the established route cache is updated and new routes are learned.

Maintaining route caches is highly beneficial when nodes are moving slowly and routes can be reused several time. Furthermore, intermediate nodes can also benefit from the gathered information and will not need to discover the complete route but rather utilise its neighbours information.

When nodes are moving with higher speed the increasing maintenance overhead of the routes degrades the overall performance of DSR.

2.6.2 Hierarchical Routing

Hierarchical routing protocols are cluster based [43, 103], the main advantage being scalability and which in turn enables the network to operate in a more energy efficient manner. Nodes with higher energy levels are selected to forward data and nodes with lower power level do the sensing i.e. the allocation of the task per node depends on the remaining energy level. During initialisation, a cluster head is designated, performing data aggregation and fusion before transmitting data to the sink to decrease the communication requirement further. All these factors increase the scalability, energy efficiency and therefore network lifetime. Typically hierarchical routing is a two layer routing protocol. One layer is used to select the cluster head, the second to execute on the routing.

Low Level Adaptive Clustering Hierarchy (LEACH) [104] is a cluster based routing protocol with distributed cluster information. The cluster head is chosen randomly to manage the higher energy consuming nodes and compresses and forwards the data that arrives from nodes belonging to its cluster. Compression reduces the amount of data forwarded to the sink. LEACH uses TDMA/CSMA [105, 106] in the MAC layer to reduce inter-cluster and intra-cluster collisions created by the periodic data collection and aggregation. This protocol is most appropriate when the environment needs to be observed constantly and periodic transmission is unnecessary. Based on simulation, only 5% of the nodes need to act as cluster heads [107].

The protocol is divided into two phases; the set up phase where the cluster heads are chosen, and secondly the steady state phase during which the data transmission to the sink is performed. During the set-up phase, a predetermined fraction of nodes, p, self-select cluster heads. A sensor node chooses a random number, r, between 0 and 1. If the number is less than a certain threshold value, $T(n)$, the node becomes a cluster head for the current round. The threshold value is calculated based on a relationship that incorporates the desired percentage to become a cluster head, the current round, and the set of nodes that have not been selected as a cluster head in last rounds, denoted by G.

$$T(n) = \frac{p}{1 - p\left(r \cdot \bmod\left(\frac{1}{p}\right)\right)} \forall n \in G \tag{2.17}$$

where

$T(n)$ = calculated threshold
P = desired number of cluster heads
r = current round

Every new cluster head sends an advertisement message alerting each other node – not cluster heads – on reception of the advertisement decide based on the RSSI to which cluster they want to belong to. These nodes inform the head that they are a member of the cluster. After discovering the number of nodes belonging to its cluster, the head follows a TDMA schedule and assigns every node a time slot. The schedule is broadcast to all nodes belonging to the cluster. The set-up is followed by the steady state phase, where nodes transmit data. After a certain time, defined a priori, the network re-initiates the start-up phase to elect new cluster heads. Each cluster uses a Code Division Multiple Access [108] to prevent collisions with other clusters. LEACH assumes that every node can transmit with sufficient power to reach the base station. Furthermore it assumes that every node has sufficient computational power to support different MAC protocols.

Therefore, its applicability to large scale network is questionable. It also assumes that nodes always have data to send and nodes located close to each other have correlated data. It is not obvious how the number of predetermined cluster heads is distributed uniformly through the network. Therefore, cluster heads can be concentrated in one part of the network and hence, some of the nodes will not be associated with a cluster head in their vicinity. The dynamic clustering generates an overhead which consumes additional energy. Finally, it assumes all cluster heads consume the same amount of energy during that time. Table 2.2 provides a comparison of LEACH with other routing protocols, Direct Diffusion and SPIN [53].

Power-Efficient Gathering in Sensor Information System (PEGASIS) [109]

Table 2.2: Overview of the Protocol Features

	SPIN	LEACH	Direct Diffusion
Optimal Route	No	No	Yes
Network Lifetime	Good	Very Good	Good
Resource Awareness	Yes	Yes	Yes
Use of Meta-Data	Yes	No	Yes

is an enhancement of LEACH, a near optimal chain based protocol. Nodes only need to communicate with their closest neighbour and take turns in communicating with the base station. After a cycle of nodes communicating to the sink is complete, a new cycle is initiated and so forth, resulting in a saving of power transmitting data per round as the power usage is spread homogeneously across participating nodes. The two main objectives of PEGASIS are to increase the lifetime of each node by collaborative techniques allowing only local co-ordination between nodes close together so that the bandwidth used for communication is reduced. PEGASIS avoids the formation of clusters. Instead of using multiple nodes to forward messages, PEGASIS only uses one node. In order to avoid the unnecessary participation of nodes, a sender adjusts its transmission power, based on Signal To Noise Ratio (SNR), to only reach the nearest node. PEGASIS assumes that every node has complete knowledge of the network topology, its maintenance being at the expense of an overhead. Furthermore, it assumes that all nodes have the same power level and are likely to drain their energy following the same profile. For nodes that are furthest away from the sink, huge delays result as well as contention at the base station. An extension of the protocol – Hierarchical PEGASIS [110] – forms a tree-like topology with each node at a particular level transmitting data to a node in an upper level in the hierarchy. This method ensures parallel transmissions and can reduce the delay by a factor up to 60.

Threshold-Sensitive Energy Efficient Sensor Network (TEEN) [111] and Adaptive Periodic Threshold Sensitive Energy Efficient Sensor Network (APTEEN) [112] are two protocols developed for time-critical applications. In TEEN, the node samples the environment at a higher rate than data is transmitted. In the process of nodes reporting changes, the cluster head applies two thresholds; a hard threshold is applied to the values of the sensed attribute, whereas the soft threshold is applied to the change in the value to be reported. Therefore, nodes only send information about the observed attribute when it falls in the range of interest, which reduces the communication need. The soft threshold reduces transmissions further, since data is only sent when the attribute value changes. The main drawback of this scheme is that, if a node does not receive the threshold value no transmission occurs. If the measured value reaches its hard threshold, the node sends data and stores the sensed value as an internal variable, sensed value (SV). A re-transmission only occurs only if both of the following conditions are met; first, the current value of the attribute is higher/lower than the hard threshold; second, the absolute values of the current measurement (CV) minus the SV from the last measurement differ at least within the range of the soft threshold. The main advantage of TEEN is its suitability to time critical applications. Energy consumption is reduced since this network is reactive. Sensing data and comparing values consume less energy than transmitting data. Another advantage is that the threshold values can be changed each time the cluster heads changes. In APTEEN the period or threshold values can be adapted according to user needs and type of application. In this protocol the cluster heads broadcast the following attributes.

i) A set of parameters the user/application needs the nodes to acquire in order to fulfil the measuring/monitoring task
ii) Thresholds: Soft Threshold (ST) and the Hard Threshold (HT)

iii) Schedule: a TDMA slot time for each node
iv) Count Time (CT): a time-based threshold in which a successful transmission of the sensed value(s) occurs

Nodes permanently monitor the surroundings and the transmission of data is governed by the same conditions as in TEEN. If a node remains silent for a time T > CT, it is forced to sense and transmit the measurement. Therefore, APTEEN is a hybrid network protocol and consequently a modified TDMA schedule is implemented. The main advantage of APTEEN is flexibility, allowing the user to define the value for HT, ST, and CT. The increase in the complexity of the implementation and synchronisation in order to realise timely updates are the main drawbacks of the protocol.

Minimum Energy Communication Network (MECN) [113] uses a low power Global Positioning System module to calculate a minimum power sub-network of the WSN. The main goal is to find a sub-network with the least number of nodes where the transmission between any two nodes is consumes the minimum energy. In this way, a node can find a minimum power path to the intended recipient without knowledge of the entire network. A node carries out a local search in its list of relays and finds a global minimum energy path to the base station. This local tree search makes MECN more robust to node failure and newly deployed nodes. MECN assumes a network in which every node can reach any other, difficult to achieve in practice. Therefore, an extension was introduced, Small Minimum Energy Communication Network (SMECN) [114] to circumvent that assumption. The following conditions are applied to the sub-graph G' that SMECN constructs;

i) The number of edges in G' is less than the number in G, $G' \in G$

ii) Transmitting to all neighbours in G' is less costly than reaching all neighbours in G.

The path between node u and v can be represented as;

$$r = (u, u_1; \ldots, u)k - 1, v) \tag{2.18}$$

The total consumption of energy along the path r can be calculated to be;

$$C(r) = \sum_{i=1}^{k-1} \left(p\left(u_i, u_{i+1}\right) + c\right) \tag{2.19}$$

where the power required to transmit data in this environment is given by:

$$p(u, v) = t \cdot d\left(u, v\right)^n \tag{2.20}$$

The constant n represents the path-loss-coefficient, depending on the channel model or the environment the network operates in, t is defined appropriately. The function $d(u, v)$ is the Euclidean distance [115] between two communicating nodes. Receiving and transmitting modes are assumed to consume a constant amount of energy, c. Furthermore, the sub-network is only constructed on the basis of minimum energy path and any decrease in the size introduces additional overheads to discovery.

Self-Organising Protocol (SOP) can either be used for mobile or stationary nodes. Node identification is achieved through the routing node that they are connected to. Some nodes gather data and forward it to a designated set of nodes acting as routers; each routing nodes must be stationary. The routing nodes, the backbone of the network, collect data and forward it to a base station. For a node to be part of the network, it must have a connection to a routing node. Employing a Local Markov Loop ((LML)) algorithm [116] to navigate the spanning tree adds a level of fault tolerance. Nodes can be addressed individually in this routing architecture, amenable to polling based data collection applications. Furthermore, the algorithm provides a low cost way of maintaining routing tables and a balanced routing hierarchy.

Sensor Aggregate Routing (Sar) is a set of algorithms that establish and maintain data aggregation [117] through monitoring specified goals in a region of interest. Nodes are segmented into clusters according to the signal strength, with only one head per cluster. A head is selected within the cluster executed through negotiation between neighbouring nodes. A node at the highest level, providing it has a local knowledge of the area amongst its one-hop-neighbours, is declared the head. Three algorithms that adopt leader based trafficking principles with knowledge of the geographical region are;

i) Distributed Aggregate Management (Dam) [118] is a lightweight protocol for targeted monitoring tasks. The protocol adopts a decision predicate at each node to decide if it participates in the monitoring task or not. Furthermore, a message exchange scheme is used to declare the manner the grouping predicate is applied. In order for a node to decide whether to take part in the grouping, it applies the predicate to the data. Furthermore, taking information on its neighbours into account allows the grouping to converge and aggregation then takes place.

ii) Energy Based Activity Monitoring (Ebam) [118] estimates the power required to achieve each target application. Nodes assigned to the aggregation generate the desired information.

iii) The Expectation-Maximisation Like Activity Monitoring algorithm (Emlam) [118] maps targets in every communication cycle. Although no a priori knowledge is used, due to the spatial continuity of the targets, Emlam exploits this knowledge to predict new positions.

Virtual Grid Architecture (Vga) [119] is a routing paradigm that uses data aggregation and in-network processing for longer network lifetime. The approach is predicated on the basis that due to the low mobility of WSN-nodes, it is reasonable to cluster nodes in a square grid topology. The squares are built without Global Positioning System (GPS), are fixed and equal size and adjacent with no overlaps. In each zone a node is chosen to act as cluster head. Data aggregation is performed in two stages, locally and globally. Every cluster head – referred to as Local Aggregators (LAs) – performs local aggregation. Global aggregation is performed by a subset of heads, called Master Aggregationss (MAs). Routing with data aggregation can be developed through Integer Linear Programming (Ilp) [119] and Clustering Based Aggregation Heuristic (Cbah) [119]. The main goal of these algorithms is to select a number of MAs out of the LAs to minimise energy consumption; LA nodes can be part of overlapping zones. Nodes that sense the overlapping area most likely generate the same data and send data to every MA it is related to. Near-optimal solutions for MA selection, in respect to their position, even if the network grows to a large scale are generated by these algorithms.

In Hierarchical Power-Aware Routing (Hpar) [120] the nodes of the network are divided into groups and zones are generated by groups that are in geographic proximity. Maximisation of the residual energy is derived from local, zone-based routing decisions, referred to as the max-min-paths i.e. the path with a maximum of the minimal power left. This path potentially consumes additional power locally, but globally, throughout the path, consumes less energy, compared to the path where each transmission segment is optimised for minimum energy consumption. On the one hand power consumption needs to be minimised whereas on the other hand the total remaining power needs to be maximised. Two assumptions are made in order to tackle the problem and enhance the outcome; first, the path using the minimum of energy is determined by a Dijkstra algorithm [121]; second, the algorithm chooses the path allowing minimisation of the remaining energy in the network. Optimisation taking into consideration both criteria is performed by the algorithm. Therefore, the minimum power consumption criterion is not a hard threshold but a range, with an upper bound of;

$$z \cdot P_{\min}, z \geq 1 \tag{2.21}$$

where z is a relaxation factor for the goal to reach the next hop neighbour with P_{min} energy spend on the transmission. With this upper bound of the minimum residual power, the algorithm can maximise the overall energy remaining.

In Two-Tier Data Dissemination (TTDD) [122] stationary nodes deliver data to, potentially, multiple base stations. The nodes have to be location aware and establish a grid structure created by an event based on greedy geographical forwarding [123]. This grid structure is used to disseminate data to mobile sinks. An event is defined by the task not subject to frequent change. The grid operates by forwarding data between adjacent nodes; nodes forward data to all adjacent nodes – dissemination points – except the originating node. Nodes that forward also store the content. This process continues throughout the network until the edge is reached. The base station can then flood a query throughout the network, forwarded along the dissemination nodes to the source of the measurement. The reverse path is re-planned by trajectory forwarding since the base station might not be at the same location from where it sent the query.

2.6.3 Location Based Routing

In location based routing, nodes are aware of either their absolute position, when (say) employing low power GPS receivers or of their relative position through the exchanges of RSSIs and thus can determine which zone/grid they belong to. Depending on how the grids are established, not all nodes participate in forwarding or monitoring; therefore, some nodes can enter a sleep mode, conserving energy. The sleep mode can however introduce difficulties when nodes are mobile, or queries are sent through the network. The following protocols are designed to solve these problems. Some of the well reported protocols are introduced, reviewing their advantages and drawbacks.

Geographical Assisted Fidelity (GAF) [124] is an energy aware location based routing algorithm, primarily designed for mobile ad-hoc networks. The network area is divided into fixed zones and form virtual grids. During the negotiation phase, nodes elect one node to remain active for the next cycle. This node then becomes the representative of the virtual grid it is assigned to and communicates data generated in the grid to the base station. Since turning off individual nodes affects network layout and routing fidelity, the nodes in GAF are organised in a way such that each active node can communicate with the neighbouring grid's head. The required geographical information to establish the virtual grids is most readily generated using GPS. Nodes mapped in the same geographical location are assumed to generate equivalent data. Therefore, all nodes, except the one with the maximum remaining energy, can enter the sleep mode in order to save overall network energy. GAF is organised in three phases repeated frequently; first, discovering neighbours in the grid, used to decide which nodes is to remain active to monitor and to forward data; second, executing on the task over a negotiated period of time; third, transitioning into a sleep state, applicable to all nodes not assigned to the grid head. There are extensions to GAF in order to support mobility [125]. Each node estimates the time to leave the grid it is currently associated with and communicates that information to its neighbours.

Geographic and Energy Aware Routing (Gear) [126] exploits the fact that queries often include geographical information. Based on this information, the query is not flooded throughout the network, but rather to a region of interest. The routing is based on energy awareness of the nodes and knowledge on where the information is located in the network. Energy is conserved compared to Direct Diffusion, where the query is forwarded throughout the network. Nodes in a Gear network keep track of an estimated cost, calculated using the residual energy and transmission distance. Furthermore, it uses the learned cost, a refinement over the estimated cost taking into consideration includes the routing cost around path loss that emerge when a node is not sufficiently close to any neighbouring node closer to the region of interest than itself. In this case these nodes generate a message transmitted through hop back in order to inform neighbouring nodes, so that the learned cost can be adjusted. Two phases are integral the Gear; first, packet forwarding where a node transmits the data one path closer to the target region. If there is more than one neighbouring node, the data is forwarded to the closest node. If no node exists closer to the region of interest, the path is broken and a second packet delivery within the region of interest is required. Within the target area, a message is routed via restricted flooding or recursive geographical forwarding depending on the density of nodes in the target area; restricted flooding is more energy efficient in low density networks whilst recursive geographical flooding is more energy efficient in high density areas [126].

Most Forward within Radius (Mfr) [127], Compass Routing Method (Dir) [128], Geographic Distance Routing (Gedir) [129] are examples of localised routing algorithms based on distance, progress and direction.

The next hop selection is executed by either forward or backward direction. All three use the same criteria based on positioning relative to the next hop or final destination node. GEDIR minimises the distance by choosing the neighbouring node closest to the final destination node. If a message is sent twice along the same path, the path is declared invalid. MFR generates the same path following a Greedy algorithm approach; DIR uses angular criteria. The node that differs least from the direct line of sight to the intended recipient is selected as the next hop. Therefore, the best neighbour minimises the product of $\overline{DA} \cdot \overline{DS}$. `S`, `D` and `A`, representing the source, destination and intermediate nodes, respectively. If the best next hop neighbour is found, the transmitting node stops forwarding the message. DIR is only loop-free if a time stamp is forced or past traffic is memorised; the others are loop-free.

The Greedy Other Adaptive Face Routing (GOAFR) [130] is a geographic ad-hoc routing algorithm combining a greedy algorithm and face routing. Greedy algorithms always choose the nearest neighbour as the next hop forwarding node which can converge to a local minima, thereby stopping the process. Face Routing (FR) [128] however can guarantee delivery under the assumption that the source and destination nodes are connected through some path. The energy consumption with FR exhibits an upper bound proportional the number of nodes in the network. Adaptive FR (AFR) also tackles the problem of energy consumption and is shown to be "asymptotically worst case optimal" [130]. Greedy algorithms perform well in high density network where nodes are close together but performance is compromised in low density scenarios.

Span [131] relies on several nodes to establish a backbone consisting of 'coordinators'. If a node locates neighbours – two at least – that cannot reach a coordinator in a three hop neighbourhood, it decides to become a coordinator itself. As a consequence of the three hop distance, the newly assigned coordinator node need not necessarily have a coordinator neighbour, reducing the energy efficiency of the algorithm. Coordinators must maintain communication partners that are in their three hop proximity, which results in the maintenance of a complicated spanning tree.

This section aimed to give an overview the state of the art research on WSNs and the protocols that are in use to disseminate data. Protocols were segmented into three categories; flat, hierarchical and location aware routing. In summary the more engineered the protocols are, the more efficient the routing but at the expense of increased communication overheads. The overheads are driven for example by the need to discover the location of the nodes such as in geographically assisted routing schemes. Other exchanges of messages are necessary to negotiate a cluster head. Adding mobility to the routing scenario further increases complexity. Nodes can no longer relay on an established route. Frequent updates are essential to maintain up to date knowledge of where the data is to be routed. Additional cost especially in terms of energy consumption is introduced when nodes need to locate using GPS. Only on acquiring location information can a node make a decision on whether so join the cluster or another, neighboured one. A different approach to the absolute positioning is relative positioning to a communication partner, an approach used later in the research to estimate the required transmission power. The term ‘virtual distance’ is introduced to treat problems of broken LoS paths. With knowledge of the distance to the communication partner, nodes can adjust their transmission power and save energy compared to the case where messages are transmitted at a pre-set power level. Furthermore this behaviour increases spatial re-use, also introduced in Section 6.4.2.

2.7 The Transport Layer

The Section presents the tasks and responsibilities assigned to the Transport Layer. Within the context of the overall network, this layer is responsible for reliable data transport, multiplexing, flow control, congestion control, network abstraction, connection establishment and termination [132, 133]. For an end-to-end connection, the layer might employ more than one network entity, or multiplex several connections on the same network entity to either guarantee a high throughput or save on the cost of employing and managing multiple network entities respectively. The transport entities involved in data transfer communicate with each other using a message header and/or control messages [10].

2.7.1 Reliable Data Transport

The most common problems with wireless communications are; the channel is prone to a significant number of transmission errors; packets are lost due to high network congestion; the intended receiver might lose packets due to overload [132]. All these potential sources of error are managed by the transport layer. Packet loss due to a fading channel is overcome by retransmission or recovery, either local or if necessary globally, the second by controlling congestion and the third through flow control [134].

2.7.2 Addressing

In terms of addressing, and for reasons of completeness, the transport layer addresses are Transport Service Access Points (TSAPs) and the network addresses are Network Service Access Points (NSAPs). TSAPs identify the transport entity, where a transport entity can establish more than one session at a time. The Transport Layer maps TSAPs into NSAPs, the latter an end-system transport entity. To establish a connection to another transport entity, the entity about to transmit data specifies the receiver with the following attributes: user, transport entity, and end-system identification. Users are specified by TSAPs and used by the receiver to identify the received message. User and process is used here respectively to describe an entity subject to the transport protocol. Usually just one transport entity operates on the system, so an identification of the transport services is not required. End system identification is executed through the NSAPs. The transport layer is not concerned about the routing; it passes the message to the network layer which deals with routing. In order to establish a connection the sending entity needs to know the address of the receiver. Two dynamic and two static strategies are used for end service identification, described in Section 2.8 in more detail.

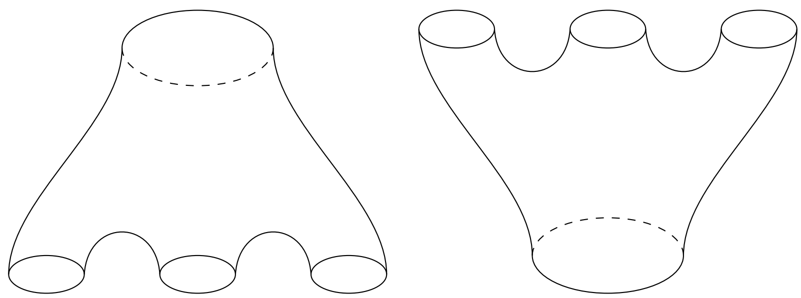
2.7.3 Multiplexing and De-multiplexing of network connections

Multiplexing is a function exercised when multiple layers employ the same connection-oriented protocol [135]. If more than one Service Access Point (SAP) is trying to establish a connection, the transport entity should handle these as separate logical connections. Multiplexing can be used in both directions; in downward multiplexing the N^{th+1} layer establishing connections to multiple N^{th} layer (Figure 2.3(a)) and upward multiplexing when multiple N^{th+1} layers use the same transport entity (Figure 2.3(b)).

2.7.4 Flow Control

Flow control is a complex mechanism at the transport layer involving interactions between transport users, entities and network services [134]. Furthermore, it has to deal with transmission delay between transport-entities, relatively large compared to the delay created by the medium. Further problems arise when the transmission delays vary over time.

The User Datagram Protocol (UDP) [136] is an example of a transport



(a) downward multiplexing (b) upward multiplexing

Figure 2.3: Multiplexing services from an upper layer to a lower layer 2.3(a) and from a lower layer to an upper layer 2.3(b)

layer that does not support flow control, neither providing a congestion control mechanism nor improved reliability. It is used for streaming data, audio or video through a network to provide a level of tolerance against packet loss and jitter. The Transport Control Protocol (TCP) [136] performs congestion control triggered frequently by a high Bit Error Rate (BER) introduced by the wireless channel. Furthermore, the concomitant overheads and end-to-end delays consume additional energy resources. Flow control can be realised through two strategies; first with an open loop controller with no feedback from the system such as Constant Bit Rate (CBR), Variable Bit Rate (VBR) and Unspecified Bit Rate (UBR) [137]; the other option is a closed loop mechanism [138].

2.7.5 Cross Layer Implementation

The trend in designing and implementing robust WSN solutions is to consider multiple layers up to the transport layer [139]. Cross layer implementations are a different design philosophy compared to the strict separation of functionality governed by the OSI layering framework [7]. In order to give more control to the upper layers or re-distribute responsibilities, WSN protocols are implemented considering cross layer operation [140, 141]. This type of implementation aims towards the distribution of the transport layer mechanisms and strategies where the transport functions have direct interaction with routing and medium access control functions [142].

2.8 Services

In order to provide the same level of services in WSNs as in wired networks, very different approaches have to be considered. Whereas in wired net-works the OSI layer framework is implemented in a fairly straightforward manner, protocols in WSNs are being developed as cross layer implementations [143].

2.8.1 TCP/IP

Within the OSI standard, the N^{th} layer provides a defined set of services to $N^{th} + 1$ layer and can only make use of the offered services from the $N^{th} - 1$, in contrast to the TCP/Internet Protocol (IP) [144] where the N^{th} layer can also directly access a layer N - n, N >n >1. In [145] the authors implemented a native TCP on a TDMA based WSN and in order to avoid end-to-end retransmissions, flow control was implemented to control single hop re-transmission.

TCP/IP assumes that all collisions are caused by congestion at the medium, not strictly the case for WSNs, since the major data loss is caused by link failure, low BER, temporarily unavailable nodes, or topology changes. Furthermore, TCP/IP assumes a symmetric up-link and down-link but in WSNs, the delay time between the links can vary considerably [146]. In direct TCP, the WSN uses TCP/IP for communication whereas in proxy TCP and native TCP, an adaption of flow control, IP addressing and packet fragmentation is needed. The authors of [145] used TCP Reno [147], characterised by four phases; slow start, congestion avoidance, fast recovery and fast retransmission. Window-based congestion control algorithms lead to an aggressive reduction of transmission data rate caused by single bit errors.

To overcome the problem, [145] suggests the proxy TCP, using a WSN flow control instead. In the slow start, the Congestion Window [147] is increased exponentially on reception of an ACK until a threshold is reached.

2.9 Cross-Layer Protocols

Two representative cross-layer protocol designs are presented as the foundation to the derivation of the proposed protocol. The Address-Light, Integrated MAC and Routing Protocol (AIMRP) and Prioritised MAC (PMAC) protocols have been implemented on micaZ hardware platform [148] (publication in Appendix B).

2.9.1 The Address-light, integrated MAC and Routing Protocol (AIMRP)

Address-light, integrated MAC and Routing Protocol (AIMRP) [149] establishes zones in the network, organised at least once on network power up. In order to support mobility, the zone set-up can be done repeatedly to maintain awareness of changes in topology; in these studies mobility is not taken into account. The so-called Tier zones are numbered from zero at the Base Station to a maximum of 254, the limit is as a consequence of the type of the variable the Tier zone ID is assigned on viz. an integer value of the size of 8 bit, ID$\in]0 : 254[$. Zero is excluded since it is reserved for the base station, 255 is usually the broadcast address of the network. Since it is assumed that nodes are static, the set up phase is executed on network power up.

The Base Station (BS) transmits an initialising Tier zone set up message, containing its own ID, its Tier ID and other values used to instruct the node of its ID. Each node receiving this message then checks its own Tier ID, set to 254 by default. If the received message contains a smaller value, then the BS is closer than that indicated by the default condition. In that case the node will set its own Tier ID to the value it received incremented by one, since it is one step further away from the BS to the node it received the message from. The node will also generate a Tier zone set up message containing its Tier ID and all the other values. If a node receives a Tier message containing the same or a higher Tier ID, no Tier set up message is required.

In the experimental implementation the nodes establish five zones, achieved by placing them sufficiently far from each other. The placement of the nodes is important in the goal to limit or suppress inter-zone communication, where zone-IDs vary. Each zone consists of five nodes generating traffic. Each node in every zone is responsibly of forwarding messages since the nodes do not have a fixed route to the BS. In order for nodes to forward messages to the next lower Tier zone, each broadcasts a RTS message, a request to transmit. The nodes in the next lower zone detect that message and the node that responds with a broadcast CTS message becomes responsible for relaying the message. All other nodes ready to transmit CTS messages overhear the exchange and desist from the broadcast. The times before sending CTSs are randomly distributed to ensure that all nodes, on average are not responding at the same instant of time. After the network is established, every node is aware of its location relative to the BS. Messages are generated at each node and a range of different experiments are carried out in order to stress the network at different levels. Messages were generated according to Poisson distributed time intervals [150] with a statistical mean value of x seconds in the first experiment, $2 \cdot x$ in the second experiment and so forth.

2.9.2 Prioritised MAC Protocol (PMAC)

The Prioritised MAC (PMAC) protocol has the ability to distinguish high and low prioritised messages [151], the goal being to serve high priority messages first at the expense of all others. This is done by two mechanisms. The first mechanism, implemented at the application layer, is that the priority queue is always checked first. If there are messages queuing in the priority queue they are served first. The second mechanism is implemented at the MAC; if the MAC layer identifies a priority message to be sent, the initial back-off time to compete for the channel is decreased, increasing the likelihood that the node with a prioritised message queued will send its message first during the contention phase.

2.9.3 Derivations for the PTS Scheme

The development of the proposed approach, referred to as the Predictable Transmission Success scheme is based on principles drawn from a combination of AIMRP and PMAC. The requirement of optimising network lifetime is the target goal and a cross layer approach is employed. The control of access times to the channel and the probability of a successful transmission are features that characterise Predictable Transmission Success (PTS).

2.10 Conclusions

The major difference between a WSN and TCP networks is that the former are data oriented whilst the latter are connection oriented [145]. As a consequence of the inherent restricted resources characteristics of a WSN, connection oriented networks would result in significant energy consumption due to for example, the excessive number of re-transmissions and overheads in maintaining connections. Thus single hop data transmission [145, 152] and data oriented networking is the only viable implementation approach; hence the need to consider the transport layer in WSN operation.

The Application Layer is responsible for the creation and classification of messages e.g. it defines if the data is event driven or a routine report, process it if required and passes it to the lower layers of the OSI stack. Dependent on the level of network implementation, the message might be passed to the MAC which then deals with the addressing in order to transmit the message to the intended receiver either to the next hop neighbour or base station. If

Table 2.3: A Summery of the most important Routing Protocols and their Properties [153]

Name	Data Centric	Hierarchical	Location Based	Data Aggregation
SPIN	✓			✓
DD	✓			✓
RR	✓			✓
COUGAR	✓			✓
AQUIRE	✓			
LEACH		✓		✓
TEEN	✓	✓		✓
APTEEN	✓	✓		✓
PEGASIS		✓		✓
GAF		✓	✓	
GEAR			✓	

routing is supported and implemented in the network, the decision on the transmission path is made by the Routing Layer. Several routing strategies that have been reported were presented in this Chapter.

Above the routing decision, the Transport Layer considers a greater view of the network managing congestion and flow control, connection orientation and end-to-end connection control. Each time a message is generated, the Application Layer attaches information about the message for use by other layers. For example, although many layers removed from the Physical Layer that governs access to the channel, this information is important in indicating the message priority level and therefore how it is to be queued. Channel access is abstracted from the Application Layer. The Routing Layer also makes the decision based on information handed over in the header by the Transport Layer. The nested nature of messages means that each layer strips a part of the information from the header provided by the upper layers and attaches information for the next hops. In this way layers influence the actions of other layers, controlling or weighting the decision making process. The manner layers interact and exchange information on messages to be send is illustrated in subsequent Chapters.

In order to enable efficient data transport with low power consumption the development of a cross layer scheme is proposed that builds on the principle of setting appropriate power levels for the connection path between neighbouring nodes in order to deliver the message with a pre-set probability of success. With this scheme, the application can differentiate messages that are sent frequently, such as status reports and event reporting messages that need to be delivered with a higher priority.

The Predictable Transmission Success adopts the principle of adjusting the transmission power level appropriate to the connection which in turn brings benefits in terms of energy savings. The practical embodiment of the prediction of the transmission success relies on harnessing the existing Ready to Send/Clear to Send handshake feature to exchange information on node movement and to retrieve information about the current channel condition. The channel condition is embedded within the RSSI measurement and information on movement requires an extension through additional bytes in the RTS/CTS messages.

The transmission condition is established per each individual packet based on the relative position and channel condition prevailing between the two communicating nodes. No additional overhead is required such as the need to gather global information of the position. Since nodes routinely exchange RTS/CTS handshakes for every message, information on transmission distance between nodes is current and continually refreshed. Thus the scheme can support mobile node environments also. Nodes that continually exchange information about movement can therefore assign a distance to neighbouring nodes and predict the new position. The scheme thus allows the pre-setting of the probability of a successful data recovery, trading the success rate for a lower transmission power.

Chapter 3 lays the foundation for the analysis of the proposed approach by introducing the detail underpinning data traffic patterns that arise commonly in WSNs. Understanding and modelling the manner that traffic is generated permits a realistic assessment of the performance of the protocol and allows comparisons to be drawn with approaches in terms of established network metrics. Chapter 4 introduces the methodology to simulate WSN and the techniques used to validate that evaluation framework, including an introduction of the network environment, an evaluation of the implemented node model and the applied protocols.

Chapter 3

Background of Research

3.1 Traffic Pattern

In the early years of telecommunication no standard existed on maintaining and planning the provision of equipment [154]. As network complexity grew, the need to balance services and cost required a scientific foundation to underpin and analyse network evolution. At its core is an underlying description of the random nature of traffic which then enables the analyses and test of network provisioning scenarios. The analysis framework is based on capturing traffic flows utilising appropriate probability theory.

A number of basic definitions need to be made when describing the interrelationships between network traffic and probabilities. Traffic volume is the amount of traffic carried in a specified period of time T; traffic flow or intensity is defined as the traffic volume divided by the time interval T. The result of the division is dimensionless and in honour of Erlang, a Danish pioneer of traffic theory, it has the dimension Erlang [E] [155]. A single channel, or device, that is busy for the total time t in the period T has a load of $\frac{t}{T}$ and therefore a maximum 1 E [156]. If more than one device is involved and t_x is the time where x out of N element/devices are busy, then;

$$\sum_{x=0}^{N} t_x = T \tag{3.1}$$

The total time that the devices are busy is therefore;

$$\sum_{x=1}^{N} x \cdot t_x \tag{3.2}$$

Furthermore, the intensity of the traffic is;

$$\sum_{x=1}^{N} \frac{x \cdot t_x}{T} \, [E] \tag{3.3}$$

3.1.1 The Poisson Process

Processes follow Poisson statistics if the following statements are true:

- the time intervals of the process do not overlap and are independent from each other
- the number of events in a certain interval only depends on the length of the interval not when it began; the process is also termed stationary [157]
- the probability that exactly one event occurs in any time interval of the length h is $\lambda \cdot h + o(h)$ is [158]

$$P\left[N(h) = 1\right] = \lambda \cdot h + o(h) \tag{3.4}$$

- the probability that more than one event occurs in any time interval of the length h is $o(h)$;

$$P\left[N(h) \geq 2\right] = o(h) \tag{3.5}$$

Some well-modelled processes that follow Poisson statistics are radioactive decay [159], page view requests on the Internet [160] and telephone traffic at switchboards [160]. Relating this behaviour to Wireless Sensor Network scenarios, a phone call is the message to the node. If the node requires to report data in response encapsulating a measurement or event, that message is transmitted to a base station. In case one of many nodes subscribe to the same base station (in a single hop environment) or next hop neighbour, a Poisson distributed inter-arrival time of data results [150].

3.1.2 Poisson Distribution

The Poisson distribution describes the re-occurrence of an unlikely event, the re-occurrence of k events in a defined time interval [150]. If λ is the interval between these events and therefore, the re-occurrence is expected on average, every $\lambda \cdot t$ time units, the Poisson process describes how likely it is for that event to occur within n time units. An illustrative example is the number of customers entering a shopping mall [161]. On average the count is six people per minute, so the expectancy value is 6, which is also the mean. The Poisson distribution gives the probability of how likely it is for only one person to enter within the next minute, or two (1.5%) or 15 (0.2%). If the Probability Density Function (PDF) [162] is accumulated to the Cumulated Density Function (CDF) [163], the latter describes the probability of up to n people entering the mall. The probability for up to ten people entering is about 91%. In Figure 3.1 the PDF and CDF are plotted as a function of the number of occurrences. Using the exemplar scenario above, similar arguments can be developed for the inter-arrival time of messages within Wireless Sensor Networks (WSNs). If a message is expected from multiple nodes generated and transmitted equally likely distributed around 160 ms, the appropriate PDF is shown in Figure 3.1. An algorithm is implemented that generates Poisson traffic patterns under user specified parameters.

The basis of the implementation centres on how often 1, the probability including all possible events $P(\Omega) = 1$, is to be multiplied by a random number between 0 and 1, random $\in (0, 1)$, to reach the threshold th or below.

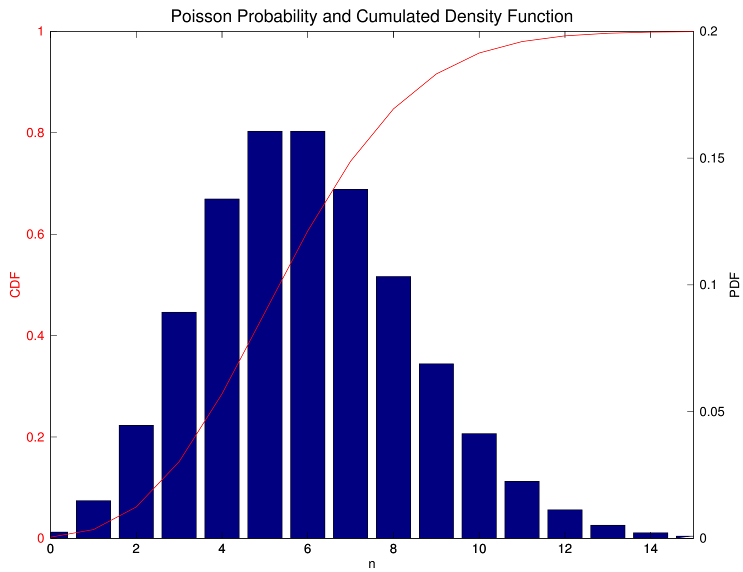


Figure 3.1: The PDF and CDF of a Poisson distribution with $\lambda = 6$

The threshold th describes the time interval in which the event re-occurs. In this particular case the threshold to end the algorithm is calculated by [164];

$$th = e^{-\lambda t} \tag{3.6}$$

where λ is the recurrence rate and t the time in milliseconds. To generate the random number the TinyOS function call `Random.rand();` was used [165], a function that returns an equally distributed 16-bit unsigned integer number. In order to scale the result to the previously defined range, $\in (0, 1)$, it needs to be divided by maximum available value viz. $2^{16} = 65536$, returning the next factor, r, for the multiplication. The number of multiplications progresses until the target value of $e^{-\lambda t}$ is reached. The distribution displayed in Figure 3.2 was generated in this manner on a micaZ mote [166]. A flow chart of the algorithm is given in Figure 3.3.

The blue curve in Figure 3.2 displays every single value that occurred and

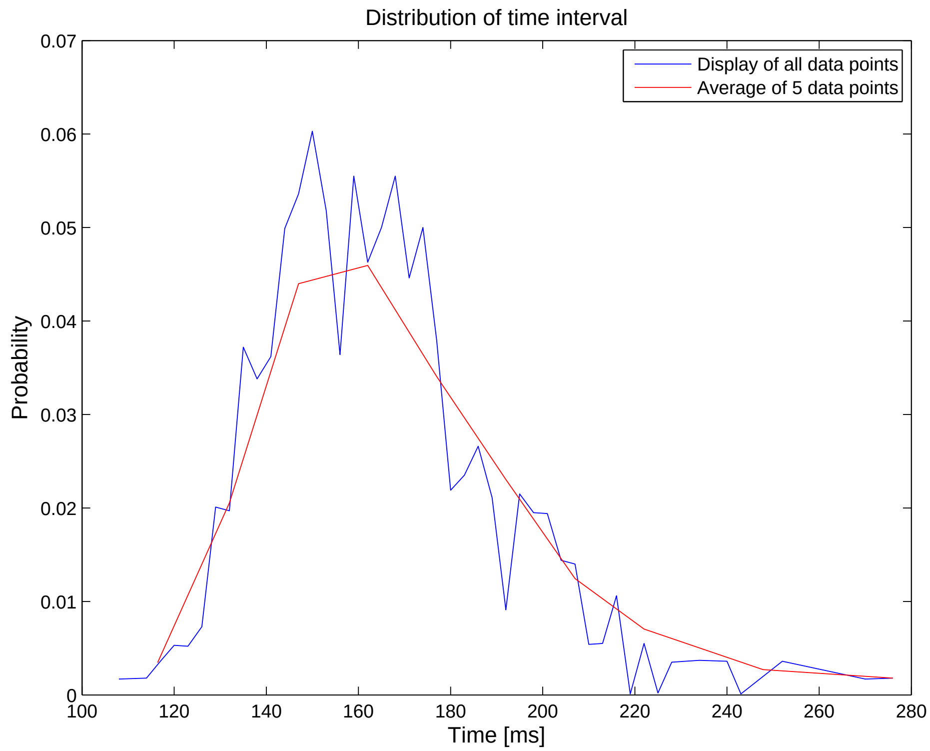


Figure 3.2: Distribution of the Inter-Arrival Time and Averaged Values following the Poisson Distribution

the appropriate probability; it is clear that most values fall between 140 ms and 180 ms. The red curve in Figure 3.2 displays a smoothed curve in which every five values are averaged and displayed in the middle of the interval. Calculating the average of all values gives a value of 164.65 ms, a very close match to the intended 163 ms, following a similar shape and progression as the PDF characteristic displayed in Figure 3.1. ≈160 ms was chosen to show that the algorithm is capable of generating the same result as obtained from experiment using micaZ nodes. The centre of the plots is shifted as a consequence of the different implementations and the maximum probability in Figure 3.2 is much smaller. The latter is as a consequence of the smaller sample size. The CDF of the two models match well [167].

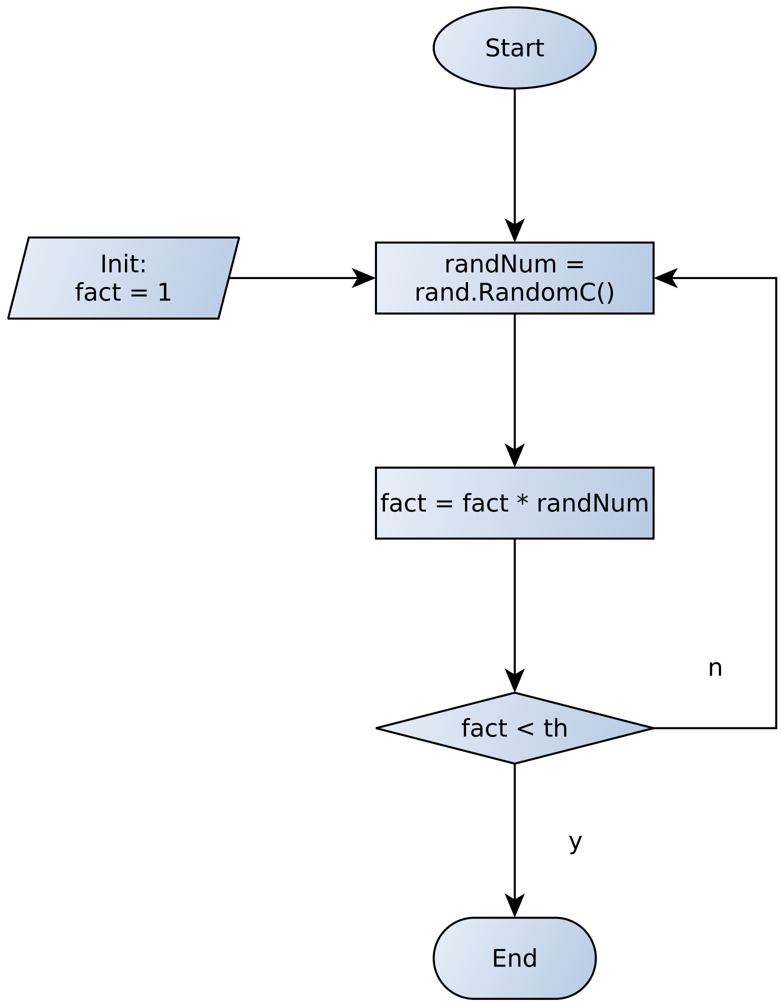


Figure 3.3: Generating Random Number using `rand.RandomC()`

The foundation of a validated network traffic statistic allows the design and analysis of protocols that control and manage the data traffic offered to a Wireless Sensor Network [167].

3.2 Channel Model

The Section provides the characteristics of wireless transmission, especially the behaviour of the channel. Unlike wired networks, WSNs use the air as the transmission medium. Electro-magnetic waves are thus transmitted through a dynamically changing environment which impacts greatly on the design of the WSN [168]. The decay in field strength compared to wired networks is much more severe, resulting in significantly shorter transmission ranges [169]. Further, transmission is much influenced by the environment; outdoor transmission performs better than in an indoor scenario in respect of transmission range [170].

3.2.1 Free Space Model

The signal decay from the free space model is given in Equation (3.9) [171] and depicted in Figure 3.4. In general, the path loss is a function of the power at the transmitter and the path loss coefficient [171]. Therefore, the Received Signal Strength Indication can be expressed as a function of transmit power and distance between sender and receiver as;

$$P_r = \frac{P_t}{PL(u,v)} \tag{3.7}$$

where the received power, P_r, is the ratio of the transmission power, P_t, and the channels path loss, PL. u and v represent the sender and receiver. For Line of Sight (LoS) transmission, the received signal strength can be calculated as [172];

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L} \tag{3.8}$$

P_r = Received Power
P_t = Transmission Power
G_r = Antenna Gain of the Receiver
G_t = Antenna Gain of the Transmitter
L = The System Loss Factor
d = Distance between Sender and Receiver

In order to keep the model as general as possible, no hardware specific gains of the transceivers are taken into consideration; thus in this case G_t and G_r are eliminated from Equation (3.8). Using the normalised power against 1 mW in decibels (dBm), the pass loss is given by, with L as a system loss factor not related to the transmission, as;

$$P_{Rx} = P_{Tx} + 20 \cdot \log_{10}\left(\frac{\lambda}{4\pi d}\right) \tag{3.9}$$

The received power, P_{Rx}, depends on the transmission power, P_{Tx}, the wavelength, λ, and the distance, d, between sender and receiver. The wavelength for this work is assumed to be that of the operating frequency of all channels of the micaZ nodes, which are at 2.4 GHz [173]. A frequency f of 2.4 GHz results in a wavelength of 0.125 metres;

$$\lambda = \frac{c}{f} \tag{3.10}$$

$$\lambda = \frac{3 \text{ e8 ms}^{-1}}{2.4 \text{ e9 Hz}} = 0.125 \text{ m} \tag{3.11}$$

Figure 3.5 records the degradation of the Received Signal Strength Indication (RSSI) as a function of distance with micaZ nodes as transmitter and receiver.

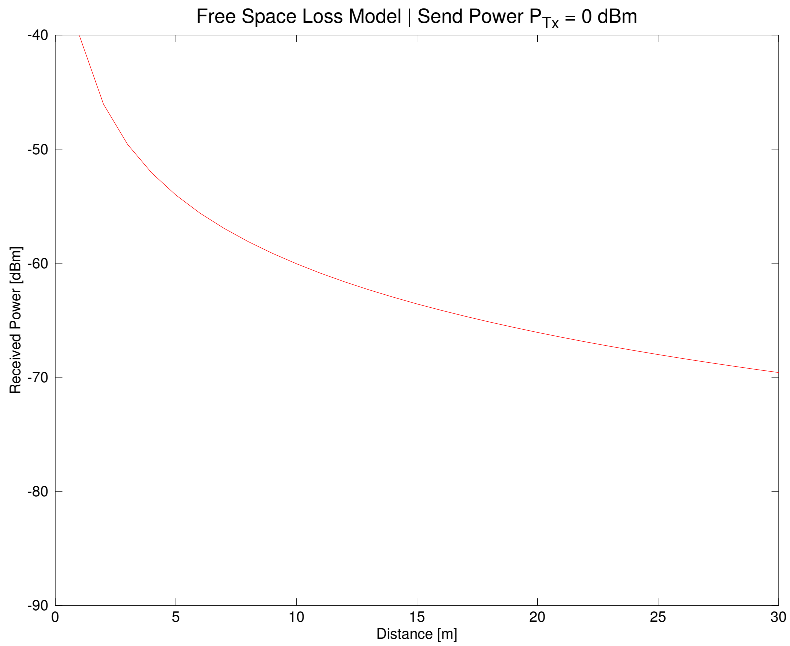


Figure 3.4: Free Space Model Power Decay over Distance

The RSSI measurements are shown from 0 metres (sender and receiver very close) up to 30 metres. A message is generated and transmitted every second at the sender node. The receiving node logged the time the messages are received as well as the signal strength. On reception of 180 messages, the distance is incremented by one metre and the measurement recorded again. On completion of the experiment, all results stored in the database are averaged over the total number of received messages and plotted over distance. A logarithmic trend curve (green) is matched to the measurements showing the same decay as the theoretically derived Free-Space-Model (red) (Equation (3.9)). The offset between the two curves is associated with the indoor (test) environment in which the RSSI level drops by ≈10 dBm.

Understanding the underlying physical properties of the channel is core to the prediction of the transmission probability on which the Predictable

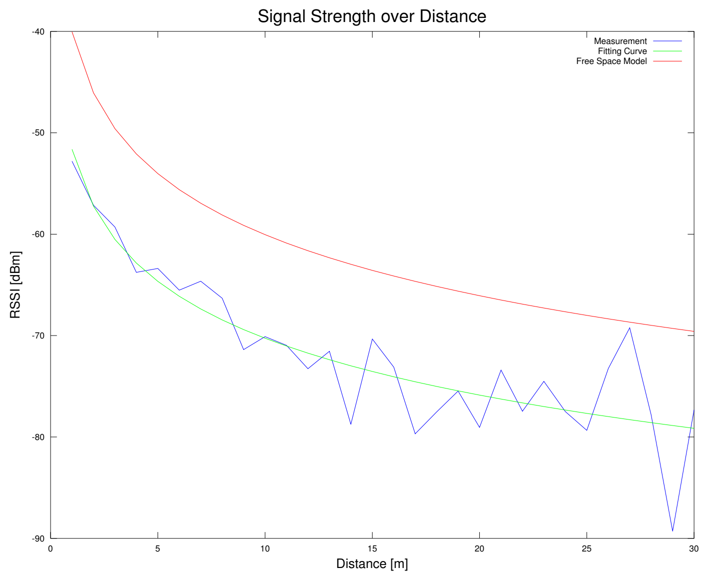


Figure 3.5: Received Signal Strength Indicator Measurement

Transmission Success scheme is based. To derive information on the physical channel enables the nodes to adjust the transmission power level adequately and continue to adapt to changes in the scenario.

3.2.2 Two Ray Ground Model

The channel model presented in Section 3.2.1 is often inaccurate due to the assumption that the transmitted signal follows only direct paths from the sender to the receiver. In most cases the signal is reflected from the ground – and other paths exist also but for the purposes of this research are not considered – and therefore signal interference occurs at the receiver, since the two main paths arrive with different delays (hence phases). The received power in the so-called Two Ray Ground Model is calculated as [171];

$$P_r(d) = P_t G_t G_r \cdot \frac{h_t^2 \cdot h_r^2}{d^4} \tag{3.12}$$

where,

P_t = transmission power
G_t = transmitters' antenna gain
G_r = receivers' antenna gain
h_t = transmitters' antenna height
h_r = receivers' antenna height
d = distance

Assuming that the $d \gg \sqrt{h_t \cdot h_r}$ distance between sender and receiver is much greater than the heights and omitting the hardware specific values for G_t and G_r, Equation (3.12) can be rewritten as;

$$P_r(d) = C_t \cdot \frac{P_t}{d^4} \tag{3.13}$$

combining antenna gains in C_t.

3.2.3 Log-Distance Path Model

The Log-Distance Model is a combination of the two models introduced in Section 3.2.1 and Section 3.2.3. Empirical studies show that radio transmission is proportional to the distance and some path loss coefficient [171];

$$P_r(d) \propto \frac{P_t}{d^\alpha} \tag{3.14}$$

with α, the path loss exponent lying in the range between 1.6 (indoor LoS) to 6 (indoor Non-LoS), $\alpha \in [1.6, 6]$ [174].

As well as fast signal decay there are other impairments which exacerbate the channel performance such as shadowing, fading, Doppler spread, co-channel and adjacent channel interference.

3.2.4 Channel Noise

As in all communication media that transmit electromagnetic signals, the wireless channel is noisy owing to a number of factors. The noise can be calculated according to [175, 30];

$$\mathrm{N[dBm]} = 10\log_{10}(\mathrm{k_B T} f_{\mathrm{BW}}) \tag{3.15}$$

where T is the temperature in Kelvin, $\mathrm{k} = 1.38{\cdot}10^{-23}\frac{\mathrm{W}}{\mathrm{kHz}}$ is the Boltzmann constant and f_{BW} the transmission bandwidth, at a micaZ node frequency of 1 MHz [148]. The noise level of the channel is calculated in relation to the thermal limit, $N_{\mathrm{thermal}} \approx -114$ dBm [176].

3.3 Walk (Mobility) Models

This Section presents reported random walk models used in WSN evaluations. Mobility increases the probability that an existing routing path will fail. Data exchange between two nodes although correctly initialised is compromised as nodes move further apart and during the transmission, the distance increases such that it degrades the Rssi, decreasing as a function of the distance between the two communicating nodes (Section 3.2).

Most commonly used models are; Random Direction [177], Random Waypoint [177], and Gauss Markov Walk [177]. The walk models are used in Chapter 5 to evaluate the performance of the proposed Predictable Transmission Success (PTS) scheme as nodes move. Consideration of the current movement direction of the communication partner enables an informed decision to be drawn on the position (and hence transmission distance) as the message exchange nears completion. For example, for nodes moving further apart, the transmitting node needs to increase the transmission power to maintain the probability of a successful retrieval of the data.

3.3.1 Random Way-Point Walk Model (RWP)

In the Random Waypoint Model (RWP) [178], a node chooses a uniformly distributed random point (the way-point) for its next destination and starts moving towards it on a straight line with a uniformly distributed velocity v, $v \in [v_{\min}, v_{\max}]$. After the node reaches the point, it remains there for a predefined period before choosing a new way-point to move to. The RWP model is a popular walk model often chosen in the evaluation of WSNs since it is implemented in Network Simulator 2 (ns-2) [179] and Global Mobile Information System Simulator (GloMoSim) [180].

The representation of the movement pattern of nodes associated with disaster recovery by RWP is questionable. By its very nature, a first response environment is hectic and random and is not easily modelled by movements in straight lines. Research shows that the RWP model decreases in speed over time [181] and thus has to be used judiciously, given adequate time to reach the steady state before recording outputs. A plot of a node's movement is depicted in Figure 3.6 where the randomly chosen speed between two way-points is represented by the colours green and red representing $v_{\min}$ and $v_{\max}$ respectively.

3.3.2 Random Direction Walk Model (RDWM)

The Random Direction Walk Model (RDWM) avoids the accumulation of way points in the centre of the 'playground' as occurs with RWP [177]. A node chooses a random direction φ [rad] $\in [0, 2\pi]$ any time it reaches the boundary and moves with a randomly chosen speed $v[\mathrm{ms}^{-1}] \in [v_{\min}, v_{\max}]$. An example movement pattern of a node is shown in Figure 3.7. The randomly chosen speed of the node is shown by the colours green and red, representing $v_{\min}$ and $v_{\max}$ respectively.

Figure 3.6: Example plot of Random Way-point Model

3.3.3 Markov-Gauss Random Walk Model (MGRWM)

The Markov-Gauss Random Walk Model (MGRWM) has been used to describe the selection of the next cell a mobile terminal enters within a hexagonal cellular architecture [182]. If the mobile terminal is free to wander, the probability of entering any of the neighbouring cell is described with $p = [1/6, 1/6, 1/6, 1/6, 1/6, 1/6]$, a single row of the probability matrix P. In MGRWM nodes are assigned a random start position, initial speed and direction. From the initial position, movement is initiated with a uniformly distributed direction offset, $\Delta\varphi[^\circ] \in [-20, 20]$, and speed offset $\Delta v[\mathrm{ms}^{-1}] \in [-0.5, 0.5]$ resulting in $v = (v + \Delta v) \oplus 0$. Movement in a backward direction is not considered ($v < 0$), but is treated with the option of turning around. In the case the node moves to a boundary, a new direction is the reflected direction off the boundary at the same entry and exit angles. A node following MGRWM is presented in Figure 3.8, the speed represented as in previous plots.

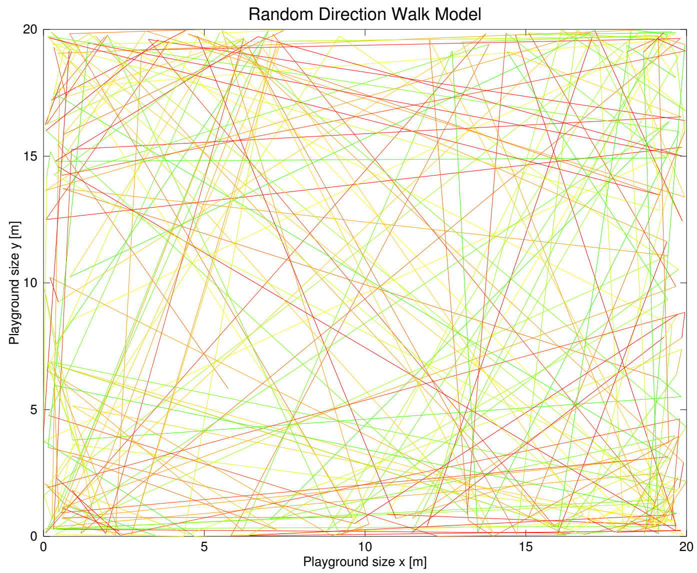


Figure 3.7: Random Direction Walk Model

For the simulation evaluation presented in Chapter 4 and Chapter 6, MGRWM is employed. According to [183] a Markov Gauss Model has been used to describe animal movement. The nodes in the displayed simulation moved a total of ≈101 m at a speed ranging between 0.13 km/hr and 4.7 km/hr.

3.4 Network Lifetime

Network lifetime is a key measure for the evaluation of the proposed scheme. Increasing lifetime offers longer unsupervised network operation. Exchanging batteries is often not an option in WSNs [184]. Authors of [185] define a measure network lifetime as the cumulative active time to the first loss of coverage. Adapting this definition for the purposes of the research, the network is considered viable until the first node is no longer capable of transmitting any information due to any factor e.g. compromised energy resource. Since nodes rarely overlap in their areas of coverage, it is assumed that the network

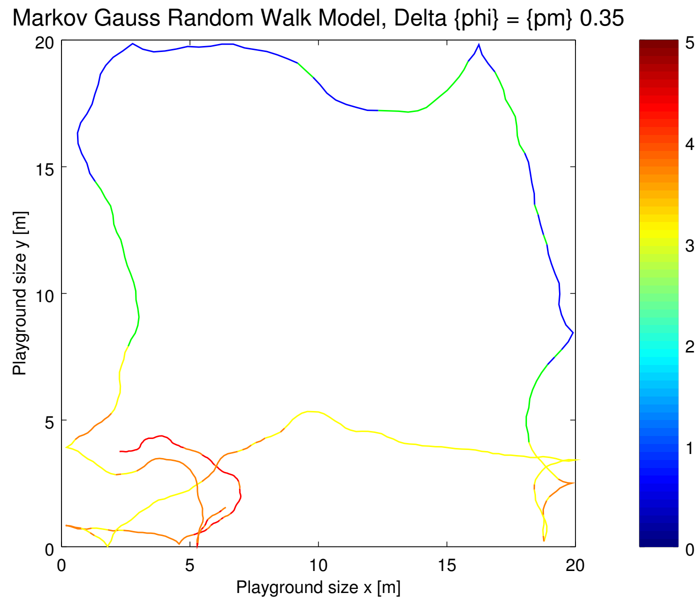


Figure 3.8: Example plot of Markov-Gauss Random Walk Model

loses operational capability the moment the first node fails.

Figure 3.9 shows the ratio of power consumed for the four main tasks executed at each node; transmission consumes the most power compared to receive, listen and computation. All the tasks are shown as a ratio compared to the energy consumed by computation.

3.5 Conclusions Channel Model, Walk Mode, Network Lifetime

The Chapter reviews the traffic statistics that apply to wireless sensor networking. A meaningful evaluation and comparison with other relevant protocols can only be executed if a representative model of data flow is utilised. The statistic must describe as closely as possible the traffic patterns

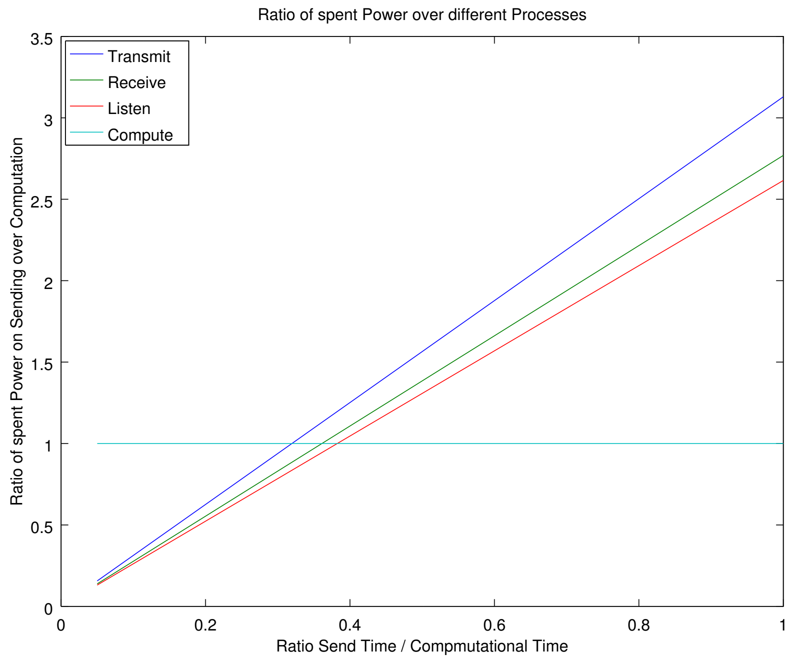


Figure 3.9: Ratio of different tasks compared to Computation

that flow between network nodes to characterise throughput and packet loss (contention). The Poisson Process is a very well-studied statistic and describes well the inter-arrival of messages at a node. Knowledge of the inter-arrival time allows the time between two packets to be determined and lays the foundation for evaluation of the entire network. Measures such as the traffic volume are directly related to the node utilisation and therefore have a strong influence on energy usage.

The Channel is another dynamically changing factor in wireless network performance and treats the impact owing to the environment. Indoor and outdoor scenarios present varying fading and shadowing levels and thus impact differently on the performance. The ability to adapt the solution in response to changing channel conditions offers additional savings in energy consumption.

Mobility of nodes introduces another significant design challenge. The ability to capture or infer information about the communication partners' direction of movement and projected position is a core requirement for deriving the optimum benefit of the proposed protocol. Knowledge of the positions of nodes allows the setting of the transmission power level at the transmitting node required to reach the communication partner with a set probability of a successful data recovery. Thus a number of well know random walk mobility modes are reviewed and the Markov-Gauss Random Walk Model is chosen for the reason that in this model the nodes do not cluster around the centre of the coverage area.

A definition of network lifetime is specified as a core metric on which an assessment of the efficacy of the performance of the protocols rests. A number of definitions have been proposed bringing levels of ambiguity in any comparison of performance. For some networks and applications the failure of a single node is not deemed crucial since the assumption is that other nodes in the network can assume the task. Other applications rely on every node being active and therefore consider a network not viable when the first node fails. The latter definition is adopted in the research.

Chapter 4

Network Simulation

The Chapter introduces the evaluation framework utilised to determine the performance of the proposed protocol through simulation of network scenarios. Experimental characterisation of an extensive network comprising a large number of nodes is challenging and costly and in most cases, simulating the network under the appropriate assumptions is an effective alternative to achieve the same goals. A number of network simulators are available, some free of charge whilst others require the active participation of the user in the evolution of the framework through for example, publishing results through a website or by providing models created by the user during the course of the research [186]. In this work, simulation is adopted as the main means of evaluating the proposed protocol supported by a limited amount of experimental corroboration.

A major consideration in the selection of the network simulation framework lies in the ready availability of a range of validated network, traffic, node and channel models. Existing models for key elements allow the focus of the evaluation to centre on the protocol stack, whilst the network/nodes are known to execute appropriate tasks. The appropriate simulation environment validated for functionality is a viable framework with which to determine performance of more extensive emulations of real applications.

Realistic assignation of parameters for network, node, channel and traffic yields valuable information on network viability before deployment. The list of target goals can be selected e.g. energy consumption ranging from the initial battery charge which decreases over time for example, according to the number of messages transmitted or the robustness of the routing protocol to mobility. Further, the impact of the environment can be evaluated as a function of dynamic changes in the channel characteristics. The environment the nodes operate in exerts a large influence on transmission; an indoor environment presents more significant challenges due to multi-paths as opposed to an outdoor free space, environment.

The Chapter introduces established simulators and develops arguments for the selection of the framework best suited to the research, as well as providing validation of the implementation. The validation is partly experimental allowing the ready variation of parameters as well as random number generator seeds to provide statistically valid outputs. The implemented node models are introduced and also validated by comparing simulation results against theoretically derived values. The validation extends to the simulation of the most basic communication protocol, Aloha [9] and the more advanced, widely used Carrier Sense Multiple Access/Collision Avoidance (Csma/CA) [187]. Conclusions are drawn on the appropriateness of the chosen simulation framework which then informs the selection of the hardware and software elements for the limited experimental phases of the work.

4.1 Simulators

A range of simulators exist which makes the choice of the appropriate evaluation environment challenging. The main objective is to identify the most appropriate simulation framework that meets the needs of the research, the operating system and offers the required degree of flexibility. Different simulators adopt different methodologies usually rooted in different communities that routinely utilise them. The most well-known network simulators are OMNeT++ [188], OPNET [189] and Network Simulator 2 (ns-2)/Network Simulator 3 (ns-3) [190]. Other simulators which have not been considered are the Global Mobile Information System Simulator (GloMoSim) [180] and TOSSIM [191]; indeed the former is no longer under active development. TOSSIM is a simulation environment that simulates entire TinyOS applications [192] with flexibility in the replacement of components.

All reported simulators have been developed to support research in the validation of protocols, network design and architecture. OPNET adopts a different approach compared to OMNeT++; ns-2/ns-3; the latter simulators offer a more complete insight as a consequence of their open-source nature. The following Sections introduce the simulators in more detail and deliver a basis upon which the choice is made.

4.1.1 OPNET

OPNET is a very effective tool for evaluating networks consisting of well-engineered defined and validated elements [193]. A solid library of network components is provided, including a wide range of pre-implemented protocols and network modules. These predefined modules can be easily assembled by the user and a network can be established quickly. The Graphical User Interface (GUI) is one of the best available; modules are provided in libraries available from a side menu and placed on the simulation landscape. Complete with validated functionality, the user specifies from the menu where and what to measure, what the traffic load should be and which protocol(s) are to be used. The main focus of OPNET is in network engineering, operations and planning [189].

For academic use, OPNET provides an evaluation environment for free under strict conditions which must be met by the user [194]. Furthermore, OPNET requires any academic license holder to create a web page containing the content of current research. There are also limitations to the technical support. As a consequence of these conditions and the inability to manipulate models, OPNET is not the framework of choice.

4.1.2 ns-2/ns-3

ns-2 is a discrete event simulator aimed at networking research specifically towards the evaluation of the Transport Control Protocol/Internet Protocol (TCP/IP), routing and multicast protocols over wired and wireless (local and satellite) networks [179, 190]. It is entirely programmed in C++ and began as a variant of REAL [195], its early development enjoying the support of both DARPA [196] and NSF [197].

The simulator requires an object oriented version of the Tool Command Language (Tcl) [198, 199], referred to as oTcl. ns-2/ns-3 can be installed on a variety of Operating System (OS), such as FreeBSD [200], Linux [201], Solaris [202] and Windows [203]. [199] asserts that Tcl can be used for Web applications, desktop GUI applications, testing and automation, database and embedded software development. All these applications stem from the degree of flexibility offered by Tcl, implemented within a small core that is highly adaptable [199]. ns-2 can be used with Network Animator (nam) [204], facilitating comparisons between networks, script automation and investigation of new network concepts [205]. All users are required to validate implemented models, and hence a validation tool is provided which can be executed easily. ns-2/ns-3 is an environment similar to OMNeT++.

4.1.3 OMNeT++

OMNeT++ is also a discrete event simulator, the basic modules programmable in C++ and larger modules assembled through the high-level language, Network Description (NED) [206]. A discrete event simulator records the state of the system at the time it changes triggered by certain, predefined events [207] e.g. changes in state, incoming messages or expiration of the timer. The NED language allows the user to describe the structure of the simulation. Single modules can be connected or can be concatenated to form larger models; the largest compound model that is an executable simulation is the network itself. The NED language provides the following features [206];

- Hierarchical, simple models can be concatenated into larger/more complex models
- Component based, making the code once written, reusable
- Flexible Interface, providing the opportunity to leave placeholders where modules can be established at network set-up time
- Inheritance, a feature of an object oriented C++ implementation providing the possibility to subclass models
- Offers Packages, JAVA [208]-like software to specify inter-dependencies amongst modules
- Inner Types, limiting the scope of modules
- Metadata Annotation, to carry data not relevant for the simulation kernel

Network design can either be edited in text mode, describing the network layout in the specified language, or by using a graphical mode; editing the network can be switched at any time using tabs. OMNeT++ also provides a graphical network inspector and a framework to implement a range of event based simulations mostly generated by the community. The principle enables all users to contribute developments, share with and receive support from others. OMNeT++ v4 is equipped with the ECLIPSE based Integrated Development Environment (IDE) [209], providing the user with ‘drag’n’drop’ functionality from a side menu to establish networking architectures and scenarios. ECLIPSE provides features such as C++ editing and GIT Version Control. This integrates not only the support to easily implement node models but also allows the control of the working software version. Instead of managing different tools, IDE integrates them. OMNeT++ extends the GUI by adding support for `*.ned`-file editing and `*.ini`-files.

The user can also assemble more complex networks and assign functionality through an automatically generated module. An option to program the network topology and its components as an editor of choice is available, executing all compiling commands from a shell environment (like the BASH shell or similar [210]). OMNeT++ is free for academic and for non-profit use without restrictions of access or modification of the code; it is the open-source in which users are allowed full control of the environment. Coupled to the fact that C++ is one of the most commonly used programming languages in engineering and given the existing, widespread experience within the research group, it was a natural choice for the research. The Mobility Framework which provides a basic node mobility model within a layered structure also exists and is relevant. Figure 4.2 shows the components that constitute the node model, illustrating the manner in which elementary modules are linked.

The Network Interface Card (NIC) is a combined module consisting of the Medium Access Control layer (MAC), the Decider and the Signal To Noise Ratio (SNR) evaluation (SNR-Eval [211]); the MAC provides the transmission delay and calculates the SNR, the Decider generates the Bit Error Rate (BER) and packet loss rates. Messages from the upper layer bypass the Decider and messages are stored in a Buffer for the period they are transmitted through the channel. If any other messages are received during that time, the SNR is re-calculated to include collisions. The Decider then inputs messages from the SNR-Eval and translates the SNR to a BER. Error-Correction algorithms can be implemented in the Decider Module. The upper layers follow the Open System Intercommunication (OSI) naming convention [8] and implementation.

4.1.4 OMNeT++; Simulator of Choice

Table 4.1 lists key features that form the basis for the decision to implement the simulation models with OMNeT++. [212] state that the popularity of OMNeT++ is low but according to [213] the number of publications using OMNeT++ is growing as is the number of research groups developing its

Table 4.1: Overview of Simulator Criteria and Rating [212]

	ns-2	GloMoSim	OPNET	OMNeT++
Interface	C++/oTcl	Parsec (C)	C	C++
Parallelism	No	Yes	Yes	Yes
Popularity	High	Moderate	Low	Low
License	Open Source	Open Source	Free for Academic	Free for Academic
Documents and User support	Excellent	Poor	Excellent	Good
Required time to learn	Long	Moderate	Long	Moderate
Scalability	Moderate	High	High	Moderate
Extendibility	Excellent	Excellent	Excellent	Excellent
GUI support	Limited	Limited	Excellent	Good

evaluation frameworks.

The OMNeT++ simulation suite is selected for a number of reasons. The manner in which the network design is established is intuitive and the functionality provided by NED is straightforward, easy to use and clearly structured. The C++ programming language, used to define node functionality is an extensive, commonly used language in engineering with the slight drawback that there is the need to re-compile the simulation in order to run under a different OS or simulator versions; however the code is portable and optimised on each target machine. Furthermore it is available for Linux systems, the preferred OS.

As well as the combination of C++ and NED, the GUI available with the default installation is powerful for network set-up and debugging, as is the Graphical Network Description (GNED) [214]. It is not only used to check the functionality of the node and network configurations but also to assemble the node and the network from basic elements. The textual and graphical representation of the network is equivalent and can be used interchangeably providing functionality such as navigation through the network, manipulation of the parameters and sub-module creation. Furthermore, multiple networks can be managed at a time. The user can drag connections from one module to another in order to establish modules or sub-models.

OMNeT++ is widely used by researchers, developers and academic staff, highly advantageous in respect of online documentation and support. The open source community is very powerful in respect of providing support to others with problems. Thus the selection of OMNeT++ is partly influenced by the experience of fellow researchers forming a local community of practitioners available for troubleshooting.

After simulation set-up, validation for correct functionality is all important, executed and best supported through the Tk Environment (TKENV) [215], a graphical user interface. TKENV also supports tracing and interactive execution of the simulation and is best used during the network development, since the state of the simulation can be inspected of at any time.

It is also possible to turn off all graphical features by switching to a Command Line Environment (CMDENV), a small user interface mainly designed for batch execution. Simulation runs specified in configuration files are executed and even if one run terminates with an error, subsequent runs are nevertheless completed. Simulation runs are identified with an ID and can be selected from the Command Line. When a simulation is initiated, frequent outputs inform the user on the progression of the run by printing values such as Sequence Number, Simulation Time, Elapsed (Real) Time and Operations per Second. A typical output is shown in Listing 4.1.

Listing 4.1: Example Command Line Output during Simulation run – FiFo Sample

```
Starting service of job
** Event #36791  T=927.60506 (15m 27s).  (FFSink) fifonet.sink (id=4)
Received job, queueing time: 2.43302sec
** Event #36792  T=927.6495 (15m 27s).  (FFGenerator) fifonet.gen (id=2)
** Event #36793  T=927.6495 (15m 27s).  (FFBitFifo) fifonet.fifo (id=3)
** Event #36794  T=927.70506 (15m 27s).  (FFBitFifo) fifonet.fifo (id=3)
Completed service of job
```

CMDENV gathers results generated after specifying the parameter to be monitored. Validation of the implemented scheme is partially automated where after the specification of all simulations parameters, the outputs are stored followed by their analysis. The validation methodology is described in Section 4.1.5, introducing an additional open source software package, Octave [216].

4.1.5 Using GNU Octave to automate Simulation Runs

Octave is a high level language designed to facilitate numerical computation [216]. The language is highly compatible with `m`-scripts and can run under MATLAB with little modifications. Although a GUI for Octave is not available, all its advantages stemming from the efficient execution of scripts far outweighs the drawback. The approach relies on the design of a set of scripts that generate `*.ini`-files for different simulation runs, scripts that feed simulations with different sets of independent random number streams, execute the simulations with the appropriate `*.ini`-file and gather all results from the output files of the simulation. The results are plotted for comparison with values derived from theoretical models or to compare with the results of others.

Octave on the one hand has the characteristics of Matlab like syntax but also retains strong similarities to C/C++ as well as executing system commands. All these features make it one of the most useful tools to pre-process the initialisation, to run simulations and post-process the results. The relationship between the implemented components is depicted in Figure 4.1, showing the simulation model, single node implementation in C++ (green), combining node models (red) and enabling automated execution through Octave (blue). A detailed description of the environment is presented in Section 4.2, including an overview of the hardware and software as well as extracts of code that help to explain the complexity of the implementation. Furthermore, the detail underpinning the automation of simulations is given.

4.2 The Node and Network Model

Here, the model and its constituent components are explained together with the implemented functionalities. A node model is presented in Figure 4.2 – designed to include a Mobility Framework – consisting of four elements; Application Layer, Network Layer and the NIC implementing the MAC, Decider and SNR-Eval respectively. Furthermore, the Mobility Framework provides a Blackboard, which enables other modules to issue messages. At the `initialisation()` of a module, it subscribes to the desired variables and during execution-time is notified about changes to those subscribed variables. This concept decreases the amount of send messages throughout the network and enables the notification of multiple modules. Modules can also unsubscribe and re-subscribe from changes.

Another core module is the Mobility Module, used to describe the manner in which nodes move during the simulation; for static nodes movement is

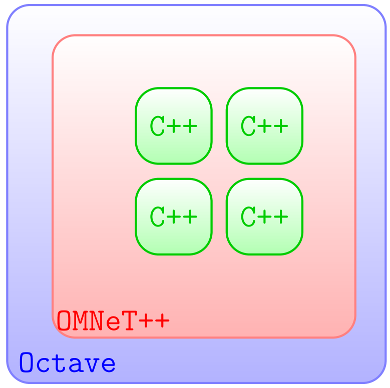


Figure 4.1: Simulation model, single nodes implemented in C++ (green), combining node models in a simulation (red), enable automated execution with octave (blue)

simply set to zero. Within Mobility, the Address Resolution Protocol (ARP) module converts network addresses to MAC addresses implemented by the user in the `getMacAddr()` of the Basic Network Layer [217] or by overloading the function of the current implementation. ARP is an address look-up that translates destination or source addresses of a message from the Network Layer Address to the Physical Address or vice versa. If a look-up pair cannot be found, the ARP raises an error and returns the message. In case of a successful look-up, the address is found in the corresponding address file of the message [218].

The functionality and complexity of the established model was extended over time. Initially the model consisted of the modules displayed in Figure 4.2. After validation of the core through implementation of Aloha and Csma/CA access schemes through the comparison of results to theoretical values, the node model was enhanced through the addition of a routing layer functionality.

4.2.1 Medium Access Control (MAC)

The Section provides the rationale for the selection of the Csma/CA [26] medium access scheme in relation to the research. Csma/CA is one of the most

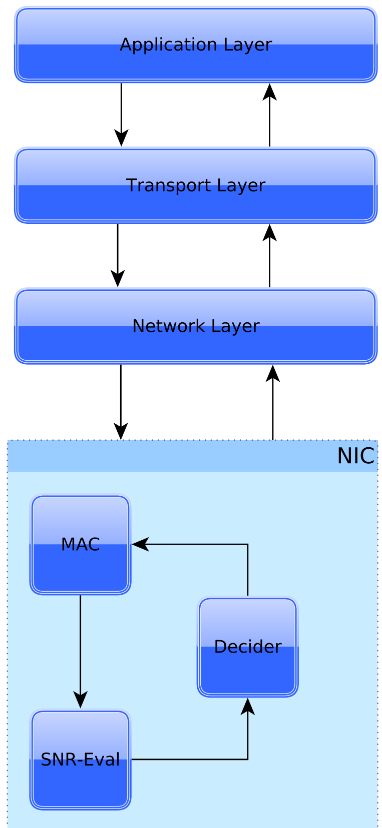


Figure 4.2: The basic node model derived from Mobility Framework

extensively studied MAC schemes with many publications reporting on its performance, its limitations and usage within the IEEE 802.11 standard [37]. Its implementation lends itself to Wireless Sensor Network (WSN) platforms since it is not excessively resource hungry. The MAC with CSMA/CA is explained in Section 2.3.2, its performance evaluated theoretically indicating a possible channel utilisation of >80% [37].

4.3 Simulation Results

The results presented in the Section provide evidence that the implemented layered model functions as intended and that correct results are produced for well-known protocols. All simulations are wrapped in Octave scripts in order to easily re-run simulations with different sets of random number streams. For the ten nodes used, it is desirable to use independent random seeds (Figure 4.3). The random number generator class `cRNG` provides functions producing random integer numbers `intrand( n )` in the range $[0, n-1]$ and `dblrand()` which generates random double integers in the range $[0, 1)$.

Seeding all ten nodes with the same stream generates data packets at exactly the same time, not representative of real scenarios and problematic for a protocol like Aloha, since every node attempts to transmit packets at the same time; therefore every packet is lost due to contention. For different runs, it is vital to invoke non-overlapping streams of random numbers [219] and in doing so also avoid unwanted correlations within results. OMNeT++ provides different seeds in `omnetpp.ini` or any other initialisation of the simulation. In order to avoid correlations, an Octave script is developed that tasks the seed tool to equip every initialisation with a different seed to be used with RNG [219]. The seed tool chooses a distance greater than the number of bits a single node is to transmit within the simulation run.

Octave generates a range of important variables in the `ini`-file; after initialisation, the Octave script `runSimulation` is executed, to initiate all simulations as defined in the `ini`-file. After the execution of all simulations, the `replicate.m` script calls a function that gathers all results from the specified file and stores outputs in a result matrix. This matrix of results and the number of runs executed are then analysed by analysing-script which calculates the theoretical values and plots the simulation results against theory.

The most basic schemes for accessing a communication channel rely on a positive Acknowledgement (ACK) to confirm a successful reception

Figure 4.3: Snapshot of the ALOHA Network. Antenna Tower is representing the Server. PCs are the simulated Nodes offering traffic

without error, with each node with data starts the transmission irrespective of network/channel conditions. For a Poisson characterised message arrival process, the throughput, S, of ALOHA can be calculated using Equation (2.4) giving a theoretical maximum throughput of $S = \frac{1}{2e} \approx 0.184$.

The ALOHA protocol is implemented on a single module, set up to generate traffic at a given rate – Poisson distributed packet inter-arrival times – to transmit to a server. At the outset to maintain simplicity, ACK is not implemented; the transmitting node assuming messages are received successfully. Figure 4.4 shows the theoretical values (Equation (2.4)) and the results from the simulation. The blue curve (simulation) is an average of ten runs each seeded by a set of random numbers. Simulation results are in good agreement with theory.

The Slotted ALOHA [24] is an enhancement to the initial protocol based on

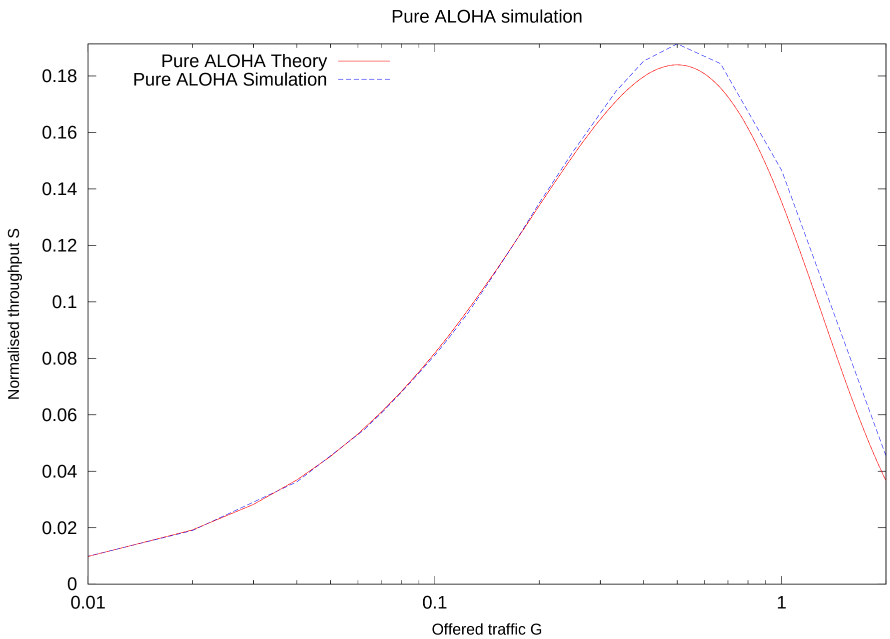


Figure 4.4: Simple node model with multiple runs of pure Aloha protocol versus calculated values

dividing the channel into slots. The node is no longer permitted to transmit when a message is created but must wait until the next slot time begins. This decreases the probability of message collisions and consequently the throughput can be doubled following Equation (2.11). With the slot time set to the transmission time of a single packet calculated to be $t_{slot} = \frac{packetsize}{bitrate}$ the theoretical maximum throughput is calculated to be $S = \frac{1}{e} \approx 0.368$ at an offered traffic rate of $G = 1$ and transmission rate of 9600 bits per second (Figure 4.5).

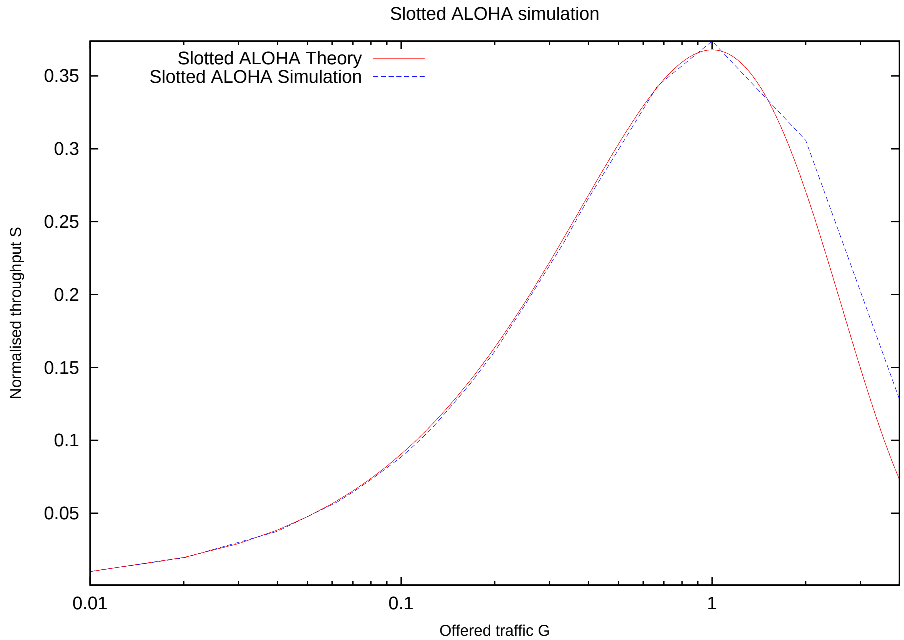


Figure 4.5: Simple node model with multiple runs of slotted ALOHA protocol versus calculated values

$$\sigma = \sqrt{\frac{1}{N}\sum_{k=0}^{N}(val_k - \mu)^2}, \text{where} \tag{4.1}$$

$$\mu = \frac{1}{N}\sum_{k=0}^{N} val_k \tag{4.2}$$

$$\text{acc} > \frac{1}{N^2}\sigma \tag{4.3}$$

For Equation (4.3) to be true, the counter set initially to 3000 decreases by one. Therefore, an accuracy set to 0.01 has to be reached 3000 times before the algorithm returns a true for the function `detected()`. At this point the simulation is considered to be statistically stable [219]. Furthermore, each simulation is repeated 10 times with a different set of random numbers. Each simulation run has to fulfil both requirements. Figure 4.5 depicts the results from the simulation compared to the theoretical values, corroborating good agreement between the two.

Following the basic validation of simulator functionality, the layered node model of Figure 4.2 is simulated supporting the Aloha protocol. The results shown in Figure 4.6 indicate that the throughput remains unchanged when compared to the simple module implementation detailed above and follows Equation (2.4). Again, a good agreement between theory and simulation is evident, validating that the layered node model functionality with Aloha is in line with theory.

The next stage of the validation centres on the implementation of the Carrier Sense Multiple Access (Csma) protocol. This protocol is widely used in commercial products, such as WiFi and Bluetooth. Csma allows multiple devices to access the same transmission medium, in this case a wireless channel. Unlike Aloha, in Csma every device senses the state of the channel and if the channel is idle for a sufficiently long period of time, the device transmits its message. The scheme limits collisions to a certain level, but does not fully prevent them. To further improve channel access, additional schemes have been developed; Carrier Sense Multiple Access/Collision Detection (Csma/CD) and Csma/CA [26].

In Csma/CA, if a node has a packet to transmit most probably generated at a higher layer, it is allocated to the MAC to execute. The MAC then ensures no other transmissions are active at the same time and switches to the Carrier Access mode, indicating the state of the MAC and to sense the channel. If the channel is idle for at least the period of a Distributed Inter-Frame Space (Difs), the node generates a random back-off between 0 and a given maximum. This randomly chosen number is multiplied by the slot time, since Csma is still a slotted protocol and a countdown is initiated until a slot becomes available. If any other transmission is sensed during the period of the countdown, the counter is frozen; if the channel is however idle

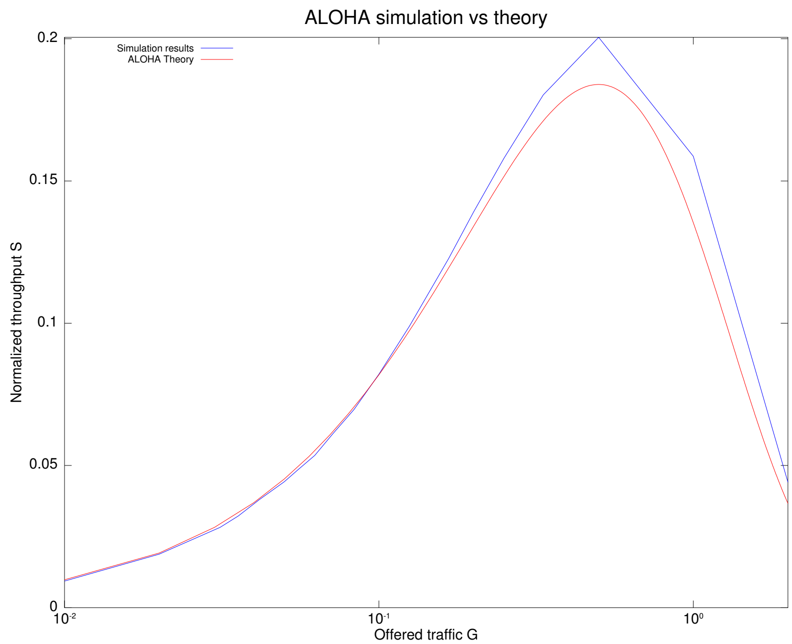


Figure 4.6: Layered node implementation with average of multiple runs of pure Aloha protocol versus theory

the countdown resumes. When the count triggers a transmit cycle, the MAC state switches to transmit (Tx) and the packet is handed to the Physical Layer (PL). On completion of the transmission, an ACK time out is scheduled to indicate if the server has taken too long to respond and that the message is most likely lost. In that case the node uses a stored back up of the original message and re-initiates the protocol.

To validate the layered node model with Csma, the scenario as in [37] is established. The variable parameter is τ, the transmission probability directly related to the Contention Window (CW) [37] as;

$$\tau = \frac{2}{\mathrm{CW}+1} \tag{4.4}$$

where CW is the period of time the channel must be sensed idle before it can be accessed [220]. [37] suggest a range for $\tau \in (0; 0.1]$. Here, the intention is not the duplicate the results of [220] but it is nevertheless important to note that the following assumptions are made;

- constant payload of 8184 bits,
- a maximum number of back-off stages of 6,
- $\mathrm{CW}_{\max} = 2^6$,
- a finite number of nodes,
- an ideal channel,
- a node always has a packet to send,
- a non-empty transmission queue

Taking these assumptions into account, the saturation throughput can be calculated as;

$$S = \frac{P_s P_{tr} E[P]}{(1 - P_{tr})\sigma + P_s P_{tr} T_s + P_{tr}(1 - P_s) T_c} \tag{4.5}$$

and the theoretical values for the CSMA/CA performance are displayed in Figure 4.7 as are the simulations results with the same set of parameters for a network of 10 nodes. Considering the complexity of the implemented layered node model, the results are in good agreement. To overcome the sensitivity owing to the selection of the random number set used to generate the back-off before transmission, the simulation is repeated with different streams of random numbers. The results are the average of the runs. Furthermore, the simulation is carried out for 20 nodes transmitting to a single server. Figure 4.8 shows the simulation results compared to theory, both in good agreement and validate that the implemented model functions according to the CSMA/CA scheme.

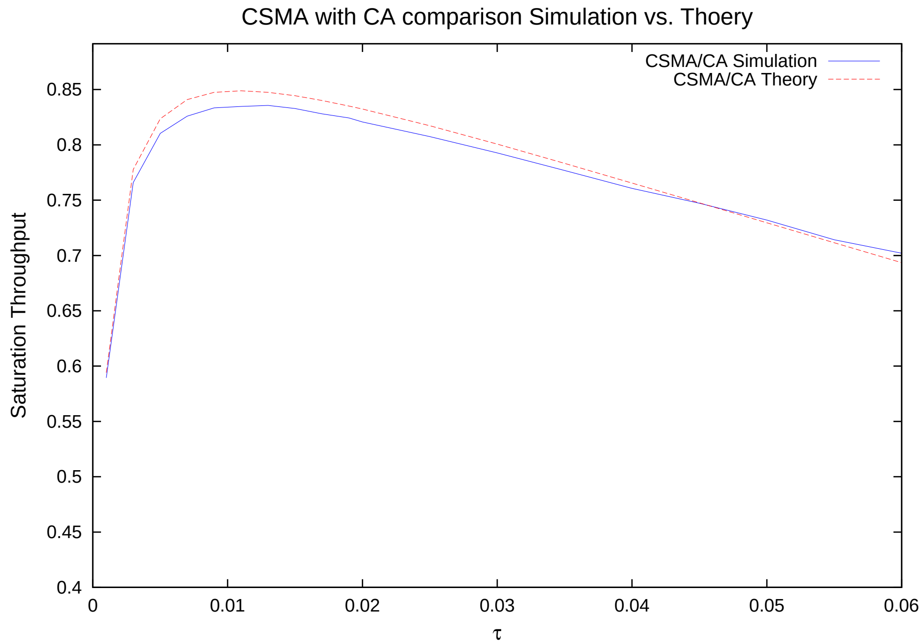


Figure 4.7: Comparing plot of CSMA/CA Theory versus the results gathered from the simulation of 10 nodes

4.4 Validation Commentary

The simplifications made to the simulation model serve to keep the implementation of the theory manageable whilst at the same time possessing sufficient accuracy to provide a robust understanding of the efficacy of the proposed protocol in terms of readily understood performance metrics. It must be noted that the implemented MAC does not fully represent the performance of the Csma/CA scheme only in highly loaded network scenarios, manifest as follows. A node initiates a transmission and is not aware that another transmission has started. Consequently, a loop is entered in which the scanning of the channel for an idle time within Difs occurs over the complete length of the packet transmission $\frac{\text{packetsize}}{\text{bitrate}} = 8.852$ ms. Within that time, the node maintains continual channel monitoring.

This behaviour is not a malfunction in the code, since the node is carrying

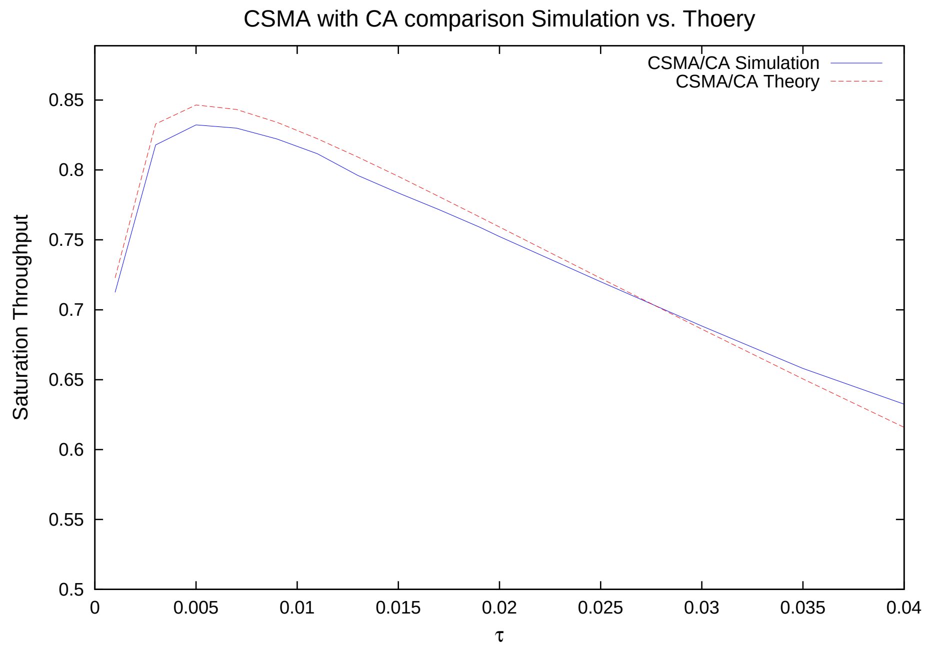


Figure 4.8: Comparing plot of CSMA/CA Theory versus the results gathered from the simulation of 20 nodes

out its designated task, but nevertheless results in a lengthening of the simulation period. On state-of-the-art computers, the performance is not compromised nor does it introduce differences with the simulation results. Thus, the MAC functionality has been implementation correctly, but at the expense of a simulation overhead. All implemented protocols behave according to theoretical expectations.

Statistical correctness is provided by `cADByStddev` [219], a derivative of the base class `cAccuracyDetection()`. An object of the class is created on initialisation of every simulation. During the simulation run the value is collected via the `collect(double val)` method at the end of every successful reception. Testing the mean value as given in Equation (4.2) and the deviation as in Equation (4.1), the accuracy is set to < 0.01 for the 3000 iterations (Figure 4.9).

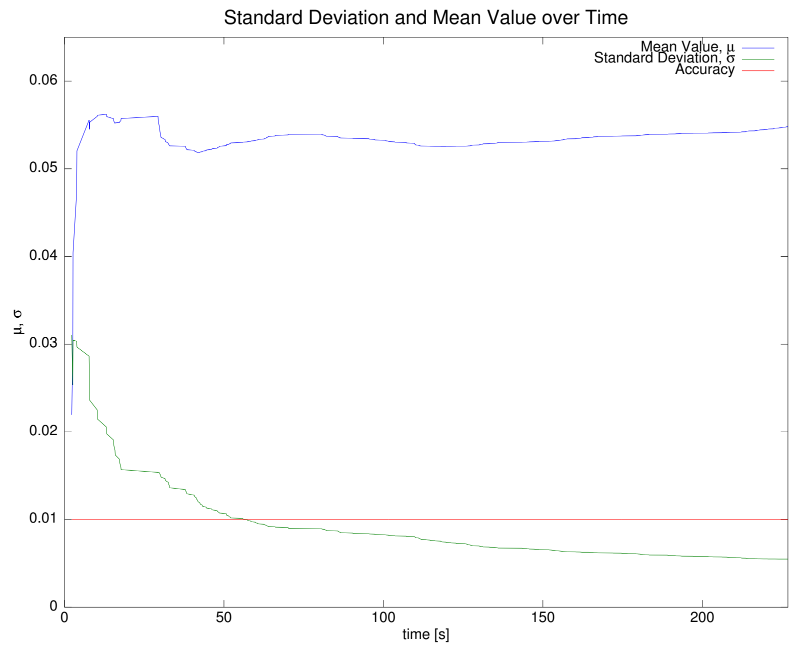


Figure 4.9: μ and σ progressing over Simulation Time

4.5 Automation of Simulations

The Section provides additional detail on the implemented simulation environment. As the complexity of the simulation grew, it proved no longer feasible to execute on a laptop. Thus a set of tools is developed to transfer the simulation environment to a more powerful computing platform with enhanced resources.

4.5.1 Hardware

The primary platform on which simulations are developed is a laptop, equipped with a dual core processor and 1 GByte RAM [221]. The Operating System selected is Linux Desktop, Ubuntu [222, 223] in various versions (release cycle of 0.5 years) (Figure 4.10). A similar system is set up on a computer

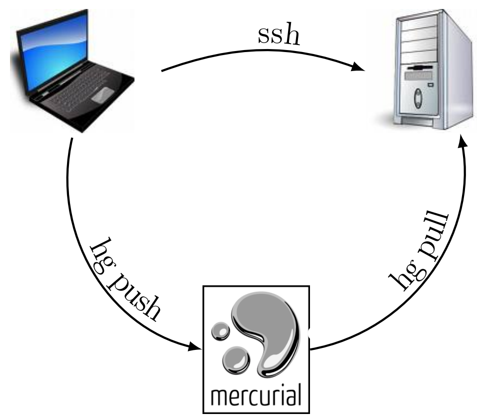


Figure 4.10: Hardware Setup of the Development machine, Simulation Server and Subversion Control Server

carrying out simulations, referred to as ‘CBM’. In addition, CBM hosts a Mercurial [224] sub-version control server. Sub-versioning software is an important strategy to maintaining track of recent and long term changes with the ability to have the freedom to restore old versions without loss of changes. Furthermore, it is important as back up to failures of the primary computer.

4.5.2 Software Setup

An efficient process for the execution of the extensive range of simulations is needed to generate a thorough performance analysis of the proposed technique. The execution of a multitude of simulations at for example, different traffic loads and random generator seeds introduce with every additional parameter loops encapsulating all previous parameters, represents a massive challenge. Thus the automation of the process is paramount to generating the scope of results.

The foundation of establishing an automation environment is the initialisation file `*.ini`-file holding the simulation parameters. Table 4.2 shows a

Table 4.2: Simulation environment and general parameters

General		Environment	
Network name	'PTS'	PlayGroundSize [m^2]	50×50
Simulation time [s]	3600	Num Subscribers	`i`
Output file	'PTS_`i`'	Max TX Power [dBm]	0
Environment	'TK'∨'CMD'	Path Loss Coeff.	4
		Saturation [dBm]	-120

sample set of parameters; a complete set of parameters, including the module parameters can be found in Appendix C.

A Perl [225] script extracts the required parameters from the `ini`-file, encapsulating them in a C-style command that can be executed by Octave and written to a file. Listing 4.2 is an example of the script.

Listing 4.2: Perl::A script to convert the used `ini`-file to be used with Octave

```
if($line =~ /^;/)
{
    print perlIni "    fprintf(fidout, \"";
    print perlIni $line;
    print perlIni "\\n\");\n";
}
```

The parameters are encapsulated in loops controlling the number of simulation runs to be executed. A nested inner loop controls the random number generator seeds and the traffic load to the network using the Poisson distribution. Once the initialisation draft is created, the simulation modules and wrapping script are committed locally and pushed to the Mercurial server.

On logging onto the simulation server, the current change set is updated using `hg pull` and updated `hg update` to initiate a working copy. The fact that the sub-version control and simulation servers are running on the same machine does not influence the process; the version control code and the code to be executed are placed in different directories. Having transferred the code and all parameters are wrapped into the loops, pre-processing of the directories and files are required. All steps are controlled and executed by the main Octave script. Clean-up of all target directories from previous result files occurs to mitigate ambiguities in post-processing the simulation results.

After clean-up, the simulation environment changes to Command Line, improving the simulation speed markedly compared with operation through the GUI. Most importantly during the simulation process transmission through the *ssh*, connection of the GUI output is prohibited. To change the execution requires a change through `changeToCMD.sh`, executing a *Regular Expression* [226] on the Makefile, a template used to parse text or source code file.

If the content of the file matches the pattern it is accepted, otherwise it is rejected. The Makefile controls the environment the simulation is compiled to and takes care of the Include directories [227], Compiler [228] and Linker [229] versions. The compiler translates the written source code into a language that the target machine can execute. In order to translate the source code, the compiler parses the code multiple times and applies several optimisations such as substitution of code declared as `inline`, the linker then binds the more abstract names to more concrete names pointing to absolute locations in the executable code. It also relocates and searches code in other libraries and makes it available to the program.

Relationships between source code files, intermediate files and executable programs are described within the language. Most importantly it controls which files are inter-dependent, removing the necessity to re-compile and link all files, only the ones that change to execute on the simulation. GNU's Not Unix (GNU) Make keeps track of the changes and compiles the required elements. To circumvent the role of the GUI, comments printed to the TKENV used to develop and debug the simulation can be blocked.

In the debugging phase, the number of outputs can grow quite rapidly and in order to reduce debugging time, another script is developed. All comments starting with the keywords: `EV`, `ev`, `cout` or `coreEV` and lines containing one of these keywords are replaced with a leading `//perl` followed by the original line. The keyword *perl* identifies the lines for subsequent printing the debugging information through Tkenv. Once all files are prepared, unnecessary code is removed and relevant parameters changed, the simulation is compiled for the target computing platform and environment. A command string is generated including the clean-up of old executables, re-generating the Makefile and recompiling ready for execution.

The main Octave script generates the necessary initialisation file and assigns the name of the simulation and corresponding `ini`-file. In order not to overwrite result files, the script also manages the naming of these files. Depending on a number of factors, such as number of nodes to be simulated, simulation time and traffic load, the simulation process can consume a significant amount of time. Octave calls on a Perl script when all runs finish and results are acquired. The script then generates an email informing that the simulation is finished and results are available for download (Listing 4.3).

Listing 4.3: Perl::Send Mail script to notify the author of the simulation

```
#!/usr/bin/perl

use MIME::Lite;

# Here may be a arg-in check to get the actual file name from the calling octave script

my $msg = MIME::Lite->new(
    From        => 'CBMSimulationServer@strathclyde.ac.uk',
    To          => 'author@strath.ac.uk',
    Subject     => 'Simulation results',
    Type        => 'multipart/mixed',
);

$msg->attach(
    Type    => 'TEXT',
    Data    => 'Simulation results',
);

$msg->attach(
    Type     => 'text/plain',
    Path     => '/Path/To/Plain/Text/ResultFile.txt',
    Filename => 'ResultFile.txt',
);

$msg->send;
```

4.5.3 Evaluation of Results

The evaluation of results is semi-automated. The acquisition of the data relies heavily on the syntax embedded within the result-vector-files. If this key was standardised, the evaluation could have been speeded up further; the evaluation script is currently modified by hand and implemented in Octave using C-style scan commands to read the content of the result vectors. The length of the seed array and the name of the simulation run are required as input, since results are averaged over a number of iterations. On calculation of the average, the resulting curve is plotted through a *gnuplot* window allowing comparisons to be made between simulation results and theoretical behaviour. After labelling, the characteristic is printed to a file using the default output devices available, usually `-depsc2`, a virtual colour encapsulated post script (eps) printer.

4.6 Conclusions

In the absence of extensive experimental resources, simulation provides an excellent alternative for the design, analysis, and validation of new network architectures and protocol concepts. In the Chapter a range of simulation environments have been introduced, highlighting their advantages and disadvantages and the basis for the selection of the framework with which an evaluation of the performance of the Predictable Transmission Success scheme is developed. Furthermore the validation of the foundation the scheme has been established including the validation of the node model and the analysis of the results.

Options for the simulation environments are introduced, the three major candidates being Opnet, ns-2/ns-3 and OMNeT++ and their features summarised (Table 4.1). OMNeT++ is selected due to a number of factors; ease of emulated network establishment, execution and output analysis being the main drivers. Given the scope and complexity of the analysis inherent in the research, an environment with freedom in respect of manipulating key parameters is a central requirement, albeit at the expense of a longer evaluation times; results can be generated simply by altering parameters, replacing single modules and storage of results. Speed up is achieved by segmenting operations; executing resource demanding simulations on one PC whilst implementation, debugging and analysis of results are carried out on another. The software version control tool Mercurial is used to distribute the implemented models across different machines. The level of access to the underlying models, random number generators, accessibility and controllability through automation, GUI support for network design and code debugging capability are a set of features which outperform other simulation environments. Octave is used to control simulation runs and analyse results.

Validation of the functionality of the modules through comparison with other reported research and mathematical models has been carried out. Well characterised channel access schemes are implemented to provide further evidence of the correct functioning of the node model. Implementations of Aloha, Slotted-Aloha and Csma/CA show that the outputs of the simulations are in good agreement with theoretically derived results.

In the research, the proposed cross-layer protocol Predictable Transmission Success is firstly modelled in C++. The resulting node model and transmission scheme are implemented along with the Physical Layer and MAC layers within OMNeT++. PTS is based on the Received Signal Strength Indication implemented in the Decider module.

The experience gained both with the protocol and hardware has highlighted that significant enhancements to the simulation can be made. As an example, a GUI with a well implemented dash-board comprising default values set for the deeper, hidden system variables is undoubtedly a great extension of OMNeT++, bringing it closer in look and feel to commercially available tools. In addition, it is clear that there is potential to further automate the process, easing the simulation burden for end-users.

Chapter 5

Power Saving Scheme; Predictive Transmission Success (PTS)

5.1 Introduction

The Chapter provides the principles underpinning a power saving approach that optimises energy consumption in Wireless Sensor Networks (WSNs). The Predictable Transmission Success scheme is rooted in a cross-layered implementation, tasking the routing layer to execute the best forwarding decision in terms of the next hop neighbour and the Medium Access Control layer (MAC) to decide upon the type of message and the appropriate adjustment to the transmission power. The practical implementation relies on a Ready to Send/Clear to Send (RTS/CTS) handshake informing on the transmission distance to the next hop neighbour derived from the Received Signal Strength Indication (RSSI).

The distance between communicating nodes is derived from the RSSI, a strong function of the prevailing channel properties. The derived distance is referred to as a 'virtual distance' since it does not represent the physical distance but rather the transmission distance between two nodes in free space. Determination of the working distance to the next hop neighbour is the key input to Predictable Transmission Success that then assigns the transmission power level for successful packet reception.

The proposed Predictable Transmission Success (PTS) scheme can be applied in a range of environments and applications. Existing protocols can be extended through the energy management inherent in PTS without compromising the application. The root lies in the nature of the scheme which follows the principle of adjusting the required transmission power level for successful data recovery.

The use of a 'virtual distance' compensates for the environment and channel conditions. PTS establishes a connection with a known, predictable probability of success and selects the optimum next hop neighbour through allocation of the required power at the transmitter.

A major advantage of PTS is the flexibility to enhance already implemented schemes with nominal changes. Prioritisation is supported not only through the selection of the optimum channel but also by further adjustment of the transmission power in order to increase the probability of successful data recovery. The prioritisation dimension is detailed in Section 6.6.

The properties of PTS are developed, supported and validated by theory and operation in terms of the transmission range, transmit power and Bit Error Rate (BER) are described.

Before the full implementation and evaluation of the scheme in OMNeT++, a simulation in C++ is implemented to generate results illustrating that the scheme is effective in its goal to increase energy efficiency on transmissions (Section 5.4). These results are best viewed as preliminary because the impact of node hardware is not considered.

Major savings in energy consumption are derived on consideration of the overall performance, although noteworthy improvements result on the proper control of the transmitted power alone. Mobility behaviour is described by the Markov-Gauss Random Walk Model (Figure 5.1).

5.2 Predictive Transmissions Success (PTS)

The approach presented is a cross-layer implementation that optimises power consumption as a function of a predicted probability of successful data recovery of scheduled transmissions.

A transmission is scheduled by the application layer when data is available to transmit or relay, most often forwarding a request from a neighbour node. The main principle of the scheme centres on a prediction of the likelihood of a successful transmission. The basic strategy is based on appropriate predictions that in turn preserve node energy because unnecessary re-transmissions are avoided. Figure 5.2 shows a typical network scenario in which Node `A` wants to send data to the Base Station. As is often the case, no direct connection is available and Node `A` needs to relay data through one of the other available nodes {`B,C,D`}. In this case Node `A` has a better probability of reaching Node `B` and Node `C` because they are located closer to Node `A`. However if Node `B` and Node `C` move away from Node `A` before the transmission is completed, the connection is broken owing to an increase in transmission distance. A solution would be to increase transmission power, counter intuitive in respect of energy consumption, but managed in the appropriate fashion in the long term yields significant savings in energy. Mobility increases the complexity of the challenge in provisioning reliable data transport. In static networks, an established route enjoys a relative long life time; in networks comprising mobile nodes, the path is highly likely to be broken due to increased distances between sender and receiver.

PTS reduces the number of unnecessary transmission attempts. The root is equipping nodes with an accurate indication of the transmission distance when the last communication finished. In this respect, RSSI measurements are recorded during the RTS/CTS handshake, initial spent power is logged

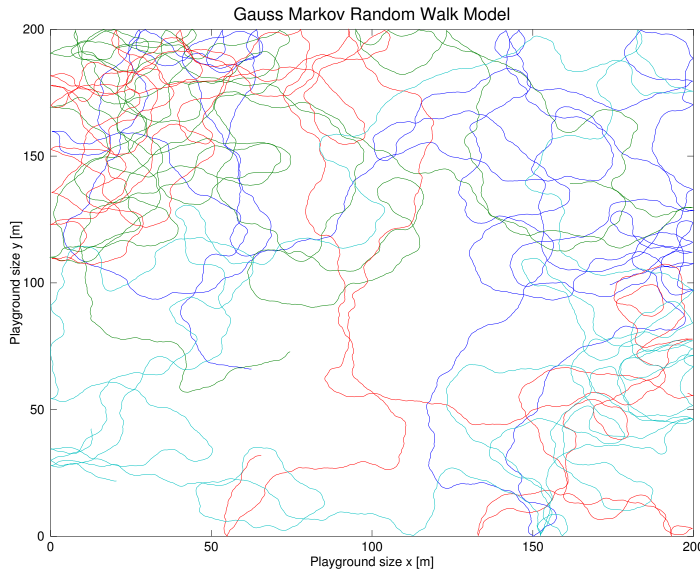


Figure 5.1: Example traces of four nodes doing a Markov Gauss Random Walk

and the direction of the current movement of any potential communication partner is included in the Clear to Send (CTS) message. Given the above information, the best communication partner can be determined and the transmission power of the sending node can be adjusted to the minimum level required to transmit data with a certain probability of success based on the estimated distance. The scheme minimises the impact on other ongoing transmissions (nodes do not interfere with each other while exchanging data), promoting spatial reuse where other nodes execute on scheduled transmissions. Every node is able to estimate the likelihood of a successful transmission.

The information on the transmission distance is calculated every time a packet is ready to be sent. Thus nodes do not set the power level based on outdated estimates of the transmission distance since re-calculation of the required transmission power is dynamic.

A packet is just as likely to be received incorrectly as its weakest bit; if

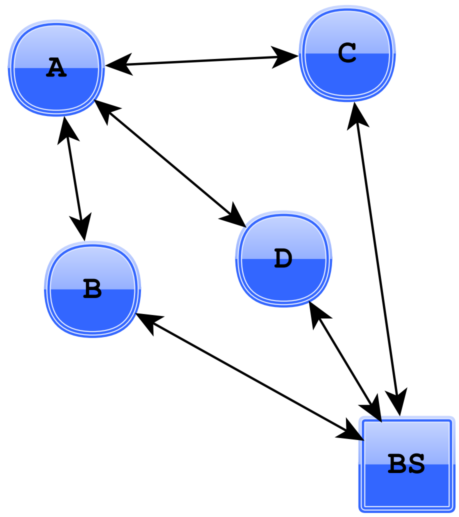


Figure 5.2: Typical WSN scenario

one bit in the data is corrupted, all content is lost for that transmission. The receiving node then needs to report that the message is corrupted and request a re-transmission. Therefore, the Bit Error Rate of the weakest bit is assumed to be representative for all the bits in the frame. The probability of a bit being successfully received depends on the distance between sender/receiver and the transmission format. The micaZ nodes in use in the research are based on a CC2420 radio interface [230] using Offset Quadrature Phase-Shift Keying (OQPSK). OQPSK receivers and transmitters operate at half the bit rate of Quadrature Phase-Shift Keying (QPSK); two QPSK transmitters/receivers are arranged in phase (I) and quadrature (Q) [231]. The BER is obtained by solving the following integral [232];

$$Q\left(\sqrt{\frac{2E_b}{N_0}}\right) = Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty \exp\left(-\frac{u^2}{2}\right) \mathrm{d}u \tag{5.1}$$

leading to the probability of an error of;

$$P_e = \frac{1}{2}\left[1 + \text{erf}\left(\frac{u}{\sqrt{2}}\right)\right]\Bigg|_x^{\infty} \tag{5.2}$$

and finally to;

$$P_e = \frac{1}{2}\left[1 - \text{erf}\underbrace{\left(\sqrt{\frac{E_b}{N_0}}\right)}_{\Xi}\right] \tag{5.3}$$

where E_b the energy spent on the transmission and N_0 is the noise on reception. E_b can be obtained from the Free Space Path Loss model [232] where the power decays logarithmically following;

$$E_b = P_{Tx} + 20\log_{10}\left(\frac{\lambda}{4\pi d}\right) \tag{5.4}$$

where P_{Tx} is the transmitter power, λ is the wavelength of the carrier frequency and d is the distance between sender and receiver. Using Equation (5.3) with Equation (5.4), the probability of a successfully transmitted bit can be calculated which for the purposes of the research, is assumed to be the probability for a successfully transmitted packet (1 - P_e)^n, where n is the number of bits in the packet [231]. The probability of a bit being lost decreases with increasing transmission power (Figure 5.3) and the probability of a frame being received successfully is displayed in Figure 5.4.

Thus the principle upon which the proposed protocol is based is straightforward; packets are more likely to be lost as the distance between sender and receiver increases. If nodes use a measure of the transmission distance, then an estimate of the likelihood of a frame being successfully transmitted to the next hop neighbour can be made. Then the scheduling of transmissions becomes more predictable and the likelihood of the loss of packets decreases and can be dynamically controlled. The decision making dimension increases

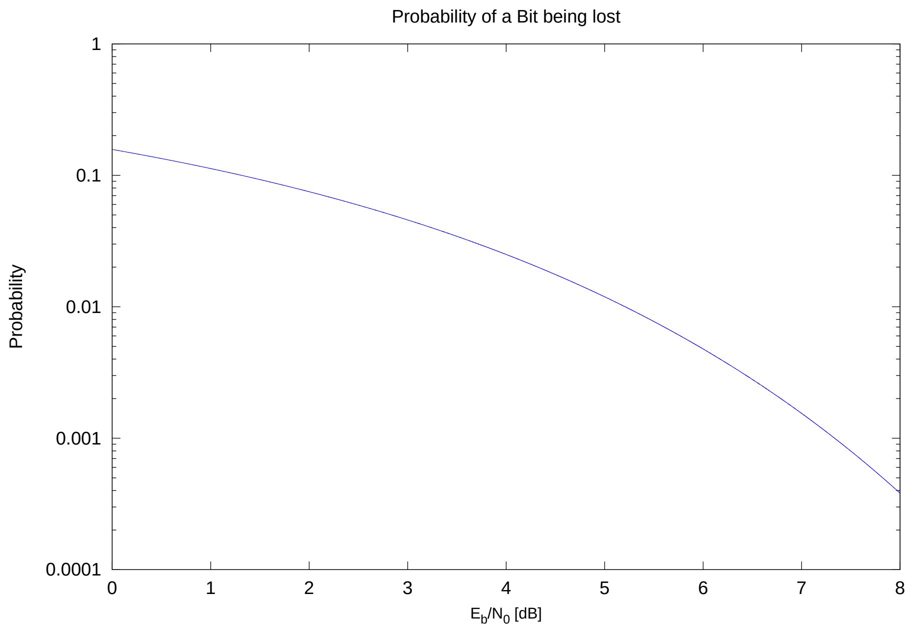


Figure 5.3: Probability of a bit getting lost

the reliability of per hop connections, which in turn optimises the power spent on multiple transmissions.

An estimation of the attainable distance can be obtained directly through the proper use of the radio interface – in this case the CC2420 – most readily through the measurement of the RSSI; authors in [233] show that since the hardware capability is continually evolving, that the RSSI can also be used as an indicator of communication distance. Therefore, the nodes can acquire awareness in relation to their relative positions and in turn can make informed decisions with energy consumption as the cost function. The proper selection of the most appropriate neighbouring node for relaying data reduces the probability of a frame being lost during on-going transmissions. Furthermore, an application can set the required success level for each packet depended on the priority of the message.

Adopting the premise that the probability of a successful transmission is

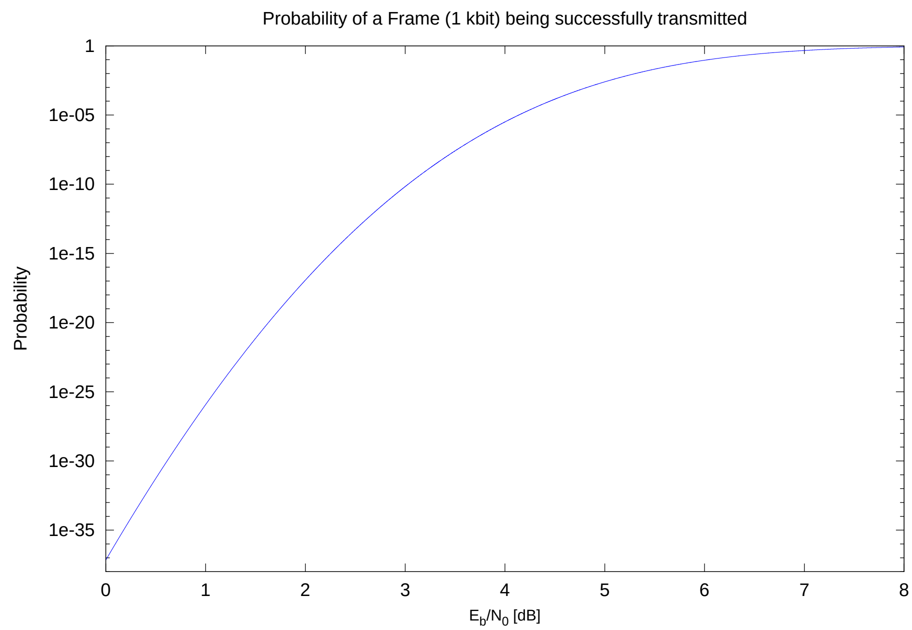


Figure 5.4: Successful frame probability

dependent on distance, the transmission power, P_e can be expressed as;

$$P_e = f\left(d, P_{Tx}\right) \tag{5.5}$$

P_e is evaluated and displayed in Figure 5.5 for a single bit and Figure 5.6 for the transmission of a packet of length 1000 bits. The x-axis represents the transmission power in dBm and the y-axis is the distance in metres between receiver and transmitter. The characteristic in Figure 5.6 decays much faster for the packet than for a single bit owing to the fact that the probability of a successful packet transmission is n^{th} power of the probability of a successful bit transmission, n being the number of bits to be transmitted. The surface in Figure 5.6 follows the shape of Figure 3.4 along the y-axis and the shape of Figure 5.5 along the x-axis.

Figure 5.6 and Figure 5.7 show the transmission success rates for a single bit and a complete data packet respectively. The success of a transmission

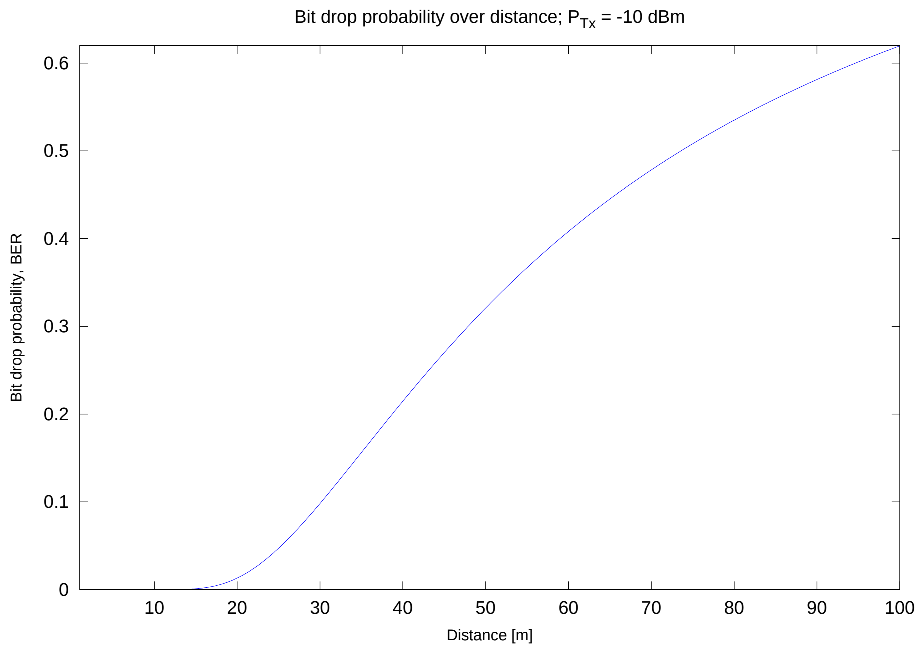


Figure 5.5: Bit drop probability over distance, Transmission power at −10 dBm

depends on firstly the distance between two nodes; the further the nodes move apart the more unlikely is a successful transmission (Chapter 3, Figure 3.4). That progression is evident in Figure 5.6 and Figure 5.7 along the distance axis. The second factor is the transmission power. Closer inspection of Figure 5.6 at a specified distance shows clearly that the decrease in the success of a transmission is as a consequence of a lower transmission power.

Embedding appropriate knowledge within the nodes enables more intelligent decision making through an informed prediction on the success rate for a planned transmission and the subsequent selection of the appropriate node taking into consideration transmission distance. Figure 5.6 and Figure 5.7 are informative but it is nevertheless difficult to derive data transfer success rates as a function of distance and transmission power. Thus Figure 5.8 presents a contour plot capturing the relationship between successful packet

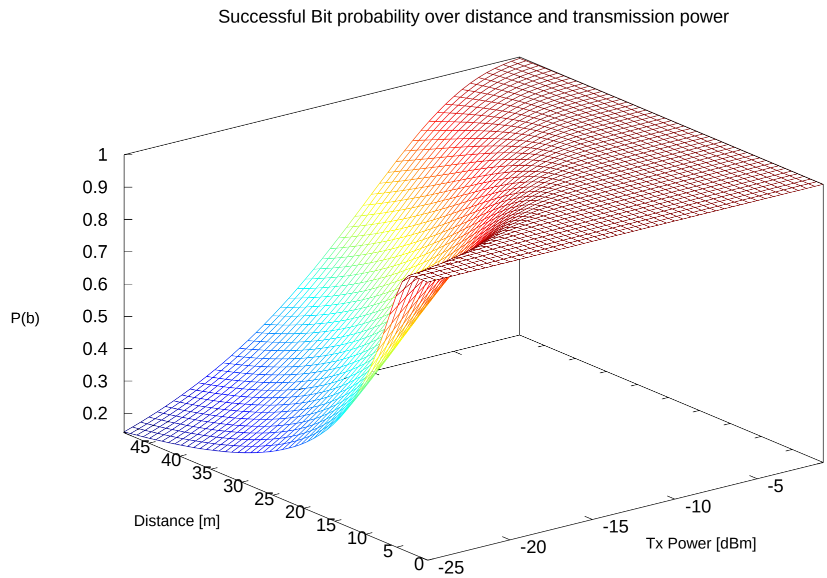


Figure 5.6: Successful Bit Probability over distance and transmission power

transmissions as a function of these two key parameters, distance and transmission power. The coloured lines represent the probabilities of a frame being successful received, the more pronounced the line, the more probable the success. Since the behaviours are relatively close, an option is to adjust transmission power to increase the likelihood of a successful frame reception; for example an appropriate threshold could be set at;

$$P_{cf\_min} \geq 0.85$$

The trade-off between transmission power and on-going transmissions can be captured through P_{Tx};

$$(1 - P_e) = P_s \tag{5.6}$$

where P_e as the probability that a bit/symbol is transmitted with an error.

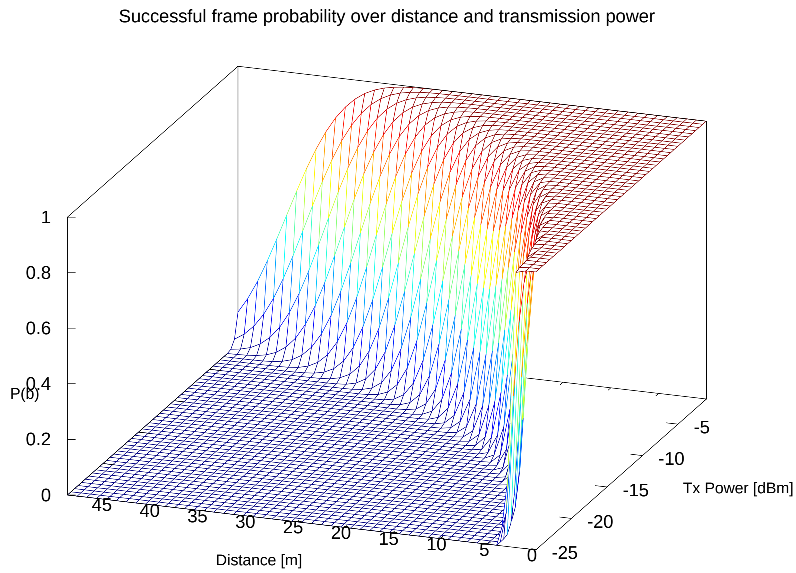


Figure 5.7: Successful Frame Probability over distance and transmission power

Thus the probability that the transmission occurred with no error;

$$P_f = (1 - 2P_e + P_e^2)^n \tag{5.7}$$

Since in OQPSK every symbol is two-bit encrypted, therefore 1000 symbols need to be transmitted with P_e from Equation (5.3);

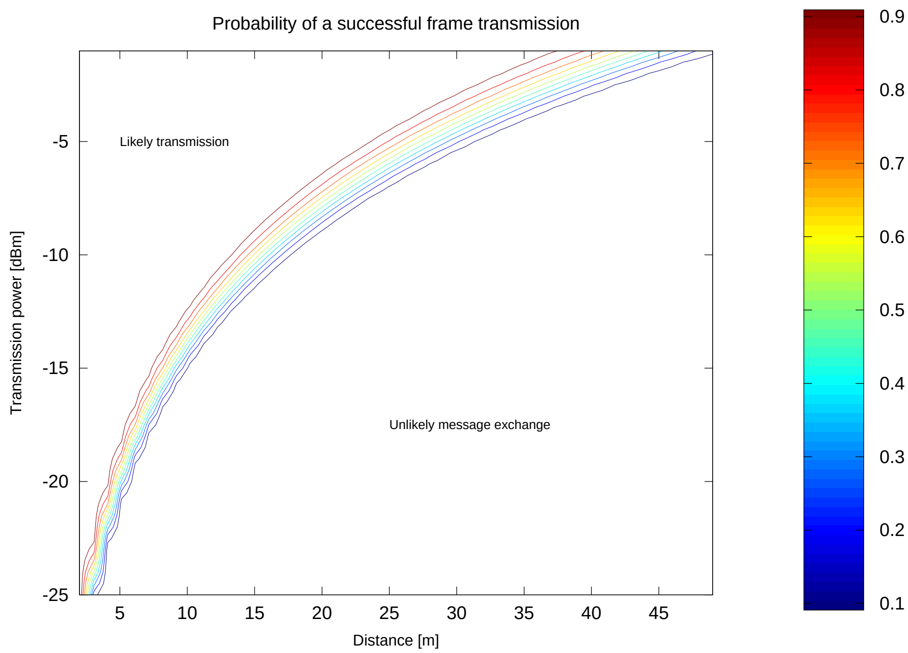


Figure 5.8: Probability of a successful frame transmission over distance and transmission power

$$
\begin{aligned}
P_f &= \left(1 - (1 - \operatorname{erf}(\Xi)) + \left(\frac{1}{2}(1 - \operatorname{erf}(\Xi))\right)^2\right)^n \qquad (5.8)\\
&= \left(\operatorname{erf}(\Xi) + \frac{1}{4}(1 - \operatorname{erf}(\Xi))^2\right)^n \\
&= \left(\operatorname{erf}(\Xi) + \frac{1}{4}\left(1 - 2 \cdot \operatorname{erf}(\Xi) + \operatorname{erf}(\Xi)^2\right)\right)^n \\
&= \left(\operatorname{erf}(\Xi) + \frac{1}{4} - \frac{1}{2}\operatorname{erf}(\Xi) + \frac{1}{4}\operatorname{erf}(\Xi)^2\right)^n \\
&= \left(\frac{1}{4}\left(\operatorname{erf}(\Xi)^2 + 2 \cdot \operatorname{erf}(\Xi) + 1\right)\right)^n = \left(\frac{1}{4}(\operatorname{erf}(\Xi) + 1)^2\right)^n
\end{aligned}
$$

$$
4^n \cdot P_f = (\operatorname{erf}(\Xi) + 1)^{2n} = 2^{2n} \cdot P_f \qquad (5.9)
$$

The required transmission success, P_f, is obtained by solving Equation (5.8) for P_{Tx};

$$
\begin{aligned}
\operatorname{erf}(\Xi) + 1 &= 2 \cdot P_{cf\_min}^{\frac{1}{2n}} \\
\operatorname{erf}(\Xi) &= 2 \cdot P_{cf\_min}^{\frac{1}{2n}} - 1 \\
\left(\frac{1}{N_0} \cdot 10^{\left(P_{Tx} 20 \log_{10}\left(\frac{\lambda}{4\pi d}\right)/10\right)} \right)^{\frac{1}{2}} &= \operatorname{norminv} \underbrace{\left(2 \cdot P_{cf\_min}^{\frac{1}{2n}} - 1 \right)}_{\Psi} \\
\frac{1}{N_0} \cdot 10^{\left(P_{Tx} 20 \log_{10}\left(\frac{\lambda}{4\pi d}\right)/10\right)} &= \left(\operatorname{norminv}(\Psi)\right)^2 \\
10^{\left(\left(P_{Tx} 20 \log_{10}\left(\frac{\lambda}{4\pi d}\right)\right)/10\right)}/1000 &= N_0 \cdot \left(\operatorname{norminv}(\Psi)\right)^2 \\
10^{\left(\left(P_{Tx} 20 \log_{10}\left(\frac{\lambda}{4\pi d}\right)\right)/10\right)} &= 1000 \cdot N_0 \cdot \left(\operatorname{norminv}(\Psi)\right)^2 \\
\left(P_{Tx} + 20 \cdot \log_{10}\left(\frac{\lambda}{4\pi d}\right) \right)/10 &= \log_{10}\left(1000 \cdot N_0 \cdot \left(\operatorname{norminv}(\Psi)\right)^2\right) \\
P_{Tx} + 20 \log_{10}\left(\frac{\lambda}{4\pi d}\right) &= 10 \cdot \left(3 + \log_{10}\left(N_0 \cdot \left(\operatorname{norminv}(\Psi)\right)^2\right)\right)
\end{aligned}
$$

where

P_{cf_min} = minimal required transmission power
P_{Tx} = transmission power used to determine 'virtual distance'
N_0 = channel noise
n = number of bits transmitted
d = 'virtual distance'
λ = carrier frequency

The required transmission power level P_{cf_min} is;

$$
P_{Tx} = 30 + 10 \log_{10}\left(N_0 \cdot \operatorname{norminv}\left(2 P_{cf\_min}^{\frac{1}{2n}} - 1\right)^2 \right) - 20 \log_{10}\left(\frac{\lambda}{4\pi d}\right) \tag{5.10}
$$

Derivation of the distance from the RSSI means that the node can evaluate the transmission power for the packet to be received with probability P_{cf_min}. The threshold value P_{cf_min} can be pre-set or adjusted as a function of the application requirement.

At the receiving node the RSSI measure is nested in the CTS reply at an overhead of an additional eight bits. Furthermore, radio interface sensitivity sets the minimum received power level and hence the dynamic range of the implementation. For the physical nodes used in the experimentation, taking into consideration the impact of the channel (Chapter 3.2) for the distances selected, RSSI signal levels of between −40 dBm and −96 dBm have been measured. In this case a dynamic range of 56 dBm needs to be represented; 56 with an 8-bit unsigned integer value gives an accuracy of $\frac{56}{256} \approx 0.2187$ dBm.

5.3 Eliminating the Channel

Initial development of the proposed protocol suggests that the transmission probability is heavily dependent on the prevailing channel characteristics. Assuming RTS/CTS handshaking, the Ready to Send receiver measures the RSSI of the packet and transmits the value to the sender encapsulated in the CTS. The sender of the Ready to Send message records the value from the received CTS message providing the information to determine how much power is required to transmit data, using simple calculations comprising additions and subtractions.

In fact on closer inspection, as is demonstrated in the following, the physical channel characteristics can be eliminated from Equation (5.10), as long as the channel characteristics are symmetric. This assumption is exploited by PTS to relax the requirement in the goal of optimising the transmission power for each message; the allocation is no longer a function of the channel.

$$P_{Rx\_RTS} = P_{Tx\_RTS} + 20\log_{10}\left(\frac{\lambda}{4\pi d}\right) \tag{5.11}$$

where

P_{Rx_RTS} = transmission power RTS message
P_{Tx_RTS} = received RSSI measurement
d = distance
λ = carrier frequency

The receiver of the Ready to Send (RTS) message can determine the distance to the RTS sender, taking into consideration the impact on the channel through the path loss viz.;

$$d_{RTS} = \frac{\lambda}{4\pi} \cdot 10^{-\frac{P_{Rx\_RTS} - P_{Tx\_RTS}}{20}} \tag{5.12}$$

Upon receiving a RTS message the node replies with a CTS message. Insertion of the measured RSSI in the RTS into the CTS does not increase the length of the packet significantly. The RTS sender potentially receives several CTS message(s) and on that basis determines the transmission distance to each sender. Due to channel asymmetry this distance, d_{CTS} , might differ from d_{RTS} but can be determined using Equation (5.12) modified to;

$$d_{CTS} = \frac{\lambda}{4\pi} \cdot 10^{-\frac{P_{Rx\_CTS} - P_{Tx\_CTS}}{20}} \tag{5.13}$$

To avoid transmission failure due to rapid yet marginal variations in the channel conditions, the CTS receiver chooses the longer distance to determine the required transmission power level. The higher distance is selected in the case when RSSI measurements differ (using the lower RSSI value) by using;

$$P_{Tx\_req} = \underbrace{30 + 10\log_{10}\left[N_0\left(\text{normsinv}\left(2 \cdot P_{cf\_min}^{\frac{1}{2n}} - 1\right)\right)^2\right]}_{=\Theta} - 20\log_{10}\left(\frac{\lambda}{4\pi d}\right) \tag{5.14}$$

Instead of using the distance derived from Equation (5.13), the CTS receiver can use Equation (5.12) which leads to;

$$
\begin{aligned}
P_{Tx\_req} &= \Theta - 20\log_{10}\left(\frac{\cancel{\chi}}{\cancel{4\pi}\frac{\cancel{\chi}}{\cancel{4\pi}}10^{-\frac{P_{Rx}-P_{Tx}}{20}}}\right) = \Theta - 20\log_{10}\left(10^{\frac{P_{Rx}-P_{Tx}}{20}}\right) \\
P_{Tx\_req} &= \Theta - P_{Rx} + P_{Tx} \\
&= 30 + \underbrace{10\log_{10}\left[N_0\left(\text{normsinv}\left(2\cdot P_{cf\_min}^{\frac{1}{2n}} - 1\right)\right)^2\right]}_{\Gamma} - P_{Rx} + P_{Tx}
\end{aligned}
\tag{5.15}
$$

Equation (5.15) simplifies to;

$$P_{Tx\_req} = 30 + \Gamma - P_{Rx} + P_{Tx} \tag{5.16}$$

where Γ is a static look-up table. Using $d_{RTS} \oplus d_{CTS}$, the maximum distance in tandem with the look-up table confines the implementation on the node to basic additions and subtractions only.

Figure 5.9 shows P_{Tx_req}, the required transmission power to deliver the packet as a function of transmission certainty. The example is for the case where transmission power is set to -5 dBm and the RSSI is measured to be -85 dBm. Embedding that information into a look-up table lowers the operational requirement of the nodes, the RSSI look up removes the need for calculations and facilitates the match between transmission power and probability of connection set by the application.

The look-up table is established using a Standard Template Library (STL) `map` [234], usually consisting of two values, the key and the mapped value and sorting is executed by the key value accessed through the '`[]`'-braces operator (`sample_map[<element_no>]`). The methodology adopts a self-adaptive approach to controlling the transmission power irrespective of channel conditions. Despite the fact that the transmission characteristics of channel are dynamic, the measure of the RSSI represents the channel condition prevailing at that instant in time. Routine RTS/CTS handshaking

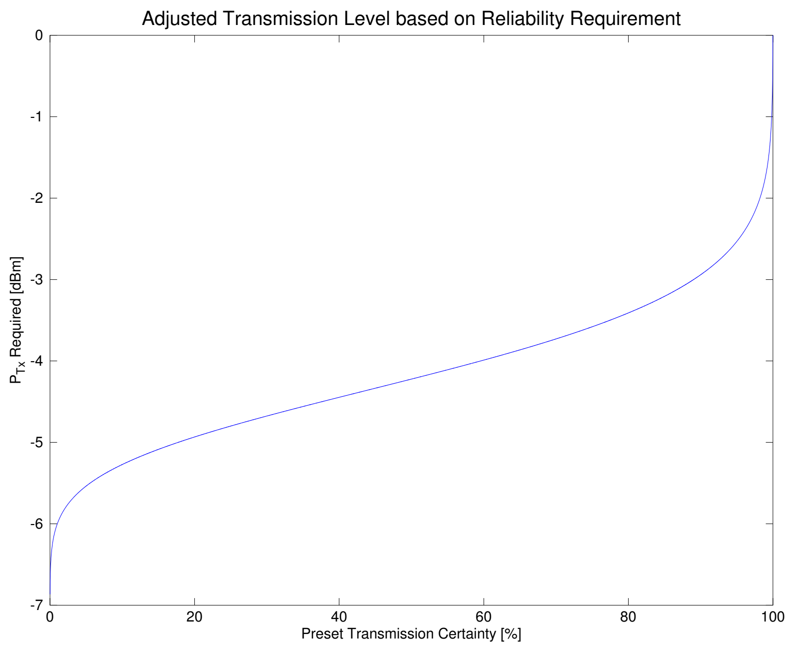


Figure 5.9: Transmission Power Requirement for pre-set Transmission Certainty

provides dynamic updates on channel conditions. For example, (say) an obstacle introduced in the connection path increases the attenuation and hence the concomitant fall in the RSSI is best viewed as an increase in the virtual distance. Virtual distance is the parameter that treats the impact of dynamic changes in path attenuation.

Since the transmitter allocates power according to the lowest RSSI – indicative of a longer virtual distance – any increase in attenuation is managed successfully. The scheme generates frequent updates on the transmission power level needed in order to meet a pre-set condition and dynamically modulates the transmission power. The same is true if path attenuation decreases. Figure 5.10 illustrates this behaviour where the initial transmission power level is set to -5 dBm and the received signal strength is measured to be

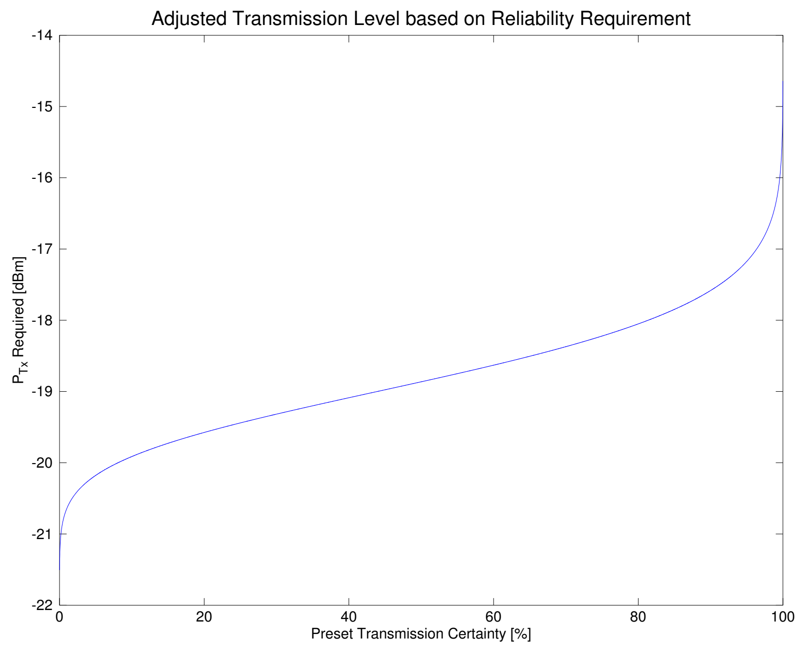


Figure 5.10: Transmission Power Requirement for pre-set Transmission Certainty scaled down

−65 dBm. The required transmission power level is selected at −17 dBm to achieve a 90% certainty of reception.

5.4 PTS Evaluation; Preliminary Implementation

The preliminary assessment of the scheme is carried out through a C++ implementation. The evaluation of the approach includes random mobility and associated hardware related node characteristics. Transmissions are handled by the channel class and paths are characterised according to the free space loss model (Figure 5.12).

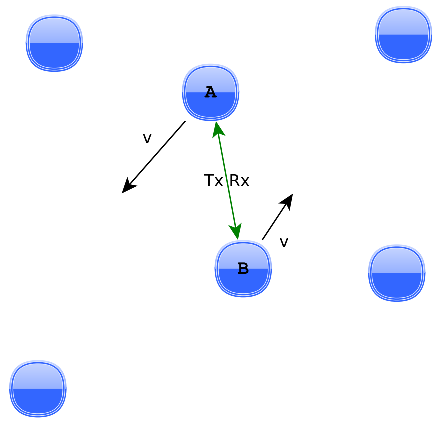
The position of the nodes is updated every 0.1 second. One node is chosen to be the transmitter determining the RSSI to all other nodes. Three scenarios are possible;

i) The message is received successfully; the RSSI is above the receivers' sensitivity (Figure 5.11(a)).
ii) The link fades during an on-going transmission. The intended receiver replies to the RTS but subsequently during the transmission, moves further away resulting in a loss of frame (Figure 5.11(b)).
iii) No communication is possible; the node falls outwith the receiving range of the RTS and therefore no CTS is generated (Figure 5.11(c)).

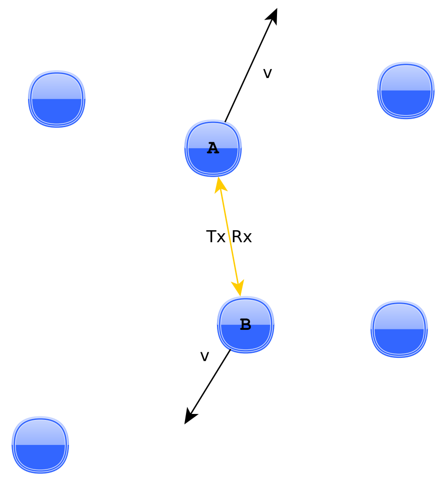
The proposed scheme aims to handle the scenarios ini) and ii). In the first case no major adjustment requires to be executed but since the application might require a certain level of reliability that the packet is received, optimisation of the transmission power by the MAC might still need to be carried out. In the second case, although, very few message fall into class ii), the probability of a successful transmission of less than 85% for more than half of the transmissions, 53.42%, have a certainty of less 90%. The proposed scheme aims to tackle this problem by embedding knowledge in the nodes in so doing empowering them to adjust the transmission power and select the appropriate routing candidate, without increasing transmission power. These results are generated through simulations presented in Chapter 6.

5.5 Examplar Application of PTS

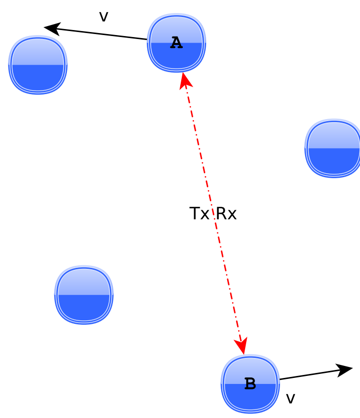
In order to further corroborate the efficacy of the proposed scheme, a real world scenario is considered, in this case the monitoring of the activity of cattle within a free roaming environment. Cows equipped with sensors measuring activity patterns aid the farmer by providing information on the health conditions of individual animals thus improving on-farm operational efficiencies. For example, this information made available within the proper time frame increases the milk productivity of a dairy farm [235, 236].



(a) Successful Message Exchange



(b) Fading Channel during Message Exchange



(c) Impossible Message Exchange

Figure 5.11: Communication Scenarios while Nodes move

In order to provide relevant information real time, individual cows are equipped with small, cheap sensor devices capable of measuring, processing and forwarding activity information, a practical implementation of a WSN.

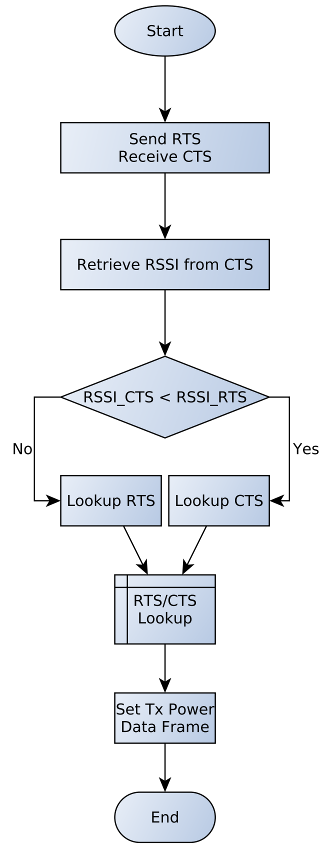


Figure 5.12: Transmission Power Decision Process

The data from each cow is transported through the network and collected centrally to be displayed to the end-user.

Network (sensor) lifetime is a core consideration; any solution in this dynamic free roaming environment must remove the practical problems associated with replacing batteries. Further the application necessitates around the clock monitoring over the lifetime of the cow – which is usually 5 years – since cattle may exhibit heat or indeed calf during the night.

Thus any scheme which extends the lifetime of the network is much sought after and indeed the case study is ideal to test out the efficacy of the proposed scheme. The sensor nodes on cattle perform the measurement, processing and download task as is but in addition PTS enhances the lifetime of restricted energy resources mostly governed by the transmission power. Cattle in close proximity to the data collection base station need not transmit at the same power level as those further away. Furthermore, since the data messages are update reports, all messages need not be received successfully at the first attempt. Counter-intuitively all messages are set to the minimum required transmission power with the resultant increase in lost packets saves more energy than sending all at a constant maximum power level. The trade-off in this particular application is battery life against successful packet reception; PTS increases network lifetime and spatial re-use.

Chapter 6 evaluates the proposed methodology, presenting simulation results for the case with no PTS and develops a comparison when using the proposed scheme.

5.6 Conclusions

The Chapter provides the principles underpinning the Predictable Transmission Success scheme. The practical implementation relies on RTS/CTS handshaking as the basis to estimate the distance to the next hop neighbour derived from the Received Signal Strength Indication. The properties of PTS are developed supported and validated in part by theory.

It is shown that the probability of a packet being successfully recovered is dependent on the distance between the two communicating nodes and the transmit power at the sender. The term virtual distance is introduced to show that the parameter captures channel conditions. The virtual distance at the time of the transmission is input to the allocation of the required transmission power that in turn fulfils a pre-set probability of successful data reception for that message. The required information, attached to the data packet, consists of the current distance, the signal strength on reception

(RSSI) and the predicated final distance. The MAC strips the information from the upper layer message header and adjusts the transmission power level, following a channel access scheme such as Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) to transmit the message. Adjusting the transmission power can be executed through a look-up table that the nodes are equipped with at deployment e.g. `std::map`. In case of multiple CTS messages the routing layer decides which node is the most appropriate candidate to forward the message. Further enhancement of these principles is introduced in Chapter 7.

Adjusting channel access times within Carrier Sense Multiple Access/-Collision Avoidance for the prioritised transmission of messages is proposed by [237]; scaling transmission power is introduced in but is not optimised to transmit messages with an application defined success rate whilst maintaining the data exchange coverage to a minimum. PTS is complementary to these schemes, most effective in mobile node WSNs, and optimising energy consumption according to applications/user requirements.

Preliminary results generated by implementing the scheme in C++ show that the proposed scheme yields a 46% increase in communications.

Chapter 6

PTS Performance Evaluation

The simulation framework described in Chapter 4 is utilised to evaluate the performance of the proposed power saving scheme. The investigation of the scheme and its behaviour when applied to a network builds on the preliminary analysis established in a C++ implementation, mainly used to validate the theoretical improvement the scheme can achieve. Furthermore, the C++ analysis treats the movement to show that the nodes, dependent on the measured Received Signal Strength Indication (Rssi) are able to adjust transmission power levels to compensate for changing path lengths.

The second analysis phase harnesses the OMNeT++ platform, the foundation for a stable and reliable simulation framework. The simulation makes use of a proven wireless mobility model and implements a layered node structure including a Network Interface Card (NIC) and Medium Access Control layer (MAC) layer. Further layers can be implemented easily, layered on top of the protocol stack. Validation results corroborate the proper functionality of the structural implementation that the MAC follows the intended behaviour and the routing scheme is as designed. These core components form the foundation with which the Predictable Transmission Success (PTS) is implemented for evaluation.

The Chapter provides an analysis of the power saving owing to and the reliability of the proposed cross layer implementation. Not only is the energy consumption of individual nodes investigated but the impact on the network lifetime is estimated to show the impact that the scheme can make in respect of extending operational integrity.

6.1 C++ Evaluation of PTS

The preliminary results of the power saving scheme implemented in C++ are presented in order to illustrate its fundamental potential to improve energy consumption. The implementation simplifies the scenario to the extent that the hardware and the related data transport delays are not considered; since the focus is on used energy, the delay of a message in the NIC is not critical. Message generation and processing does not significantly impact on the amount of energy the system consumes when compared to the power spent on transmissions. This implementation is aimed at proving that the scheme can indeed achieve savings in power.

The analysis considers 20 mobile nodes within 50×50 m^2 area. The movement pattern follows the Gauss-Markov-Walk model (Section 3.3.3). The assumed bandwidth is the maximum available to micaZ nodes viz. 256 kbit/s and the packet size is set to 6144 bits. Each node (`A`) in the network transmits messages at random times, equally distributed to a randomly chosen communication partner (`B`). The measurement of the distance between nodes, which in turn sets the required transmission power level, is executed using the method described in Section 5.3; each transmitting node adjusts its transmission power to a minimum. The successful reception rate is then verified after transmission through the channel. To capture the degree of power saving owing to the proposed scheme, two simulations are carried out at a pre-set certainty of 80% and 90% respectively. The pre-set certainty is a value the application or user selects which sets the power used on every transmission.

Figure 6.1 and Figure 6.2 show that the nodes are able to save a notable amount of energy ranging from 15.5% to 9.7% for a pre-set certainty of 80% or 90% respectively.

Having demonstrated that the proposed scheme is able to conserve energy, further evaluation is undertaken utilising the network simulator.

6.1.1 Network Lifetime

For the purposes of the research here, network lifetime is defined as the period during which it is functional, up to the point that the first node fails to transmit irrespective of the cause of failure (Section 3.4 [184]). However, since it is difficult to simulate the range of triggers for this state, the evaluation of network lifetime focuses on the failure due to node power shortage only.

Nodes are equipped with a battery providing 5 mJ of energy, a relatively modest source but sufficient to determine the behaviour of the PTS scheme in comparison to other approaches. Furthermore, the evaluation centres on considering the power used to transceive only; the power consumption due to data transmission and the power level at which the messages are sent has the greatest impact on power consumption. The assumption is that the processing at the node has a relatively modest impact on the overall power consumption [238]. Figure 6.3 shows a comparison of two network scenarios; one utilising PTS, whilst in the other no energy saving scheme is applied.

As expected, energy consumption is faster for nodes not equipped with PTS. The evaluation is programmed to stop as soon as one node is fully drained of power. At the end of the simulation, all nodes are un-initialised and their battery charge level are re-set. The power level remaining at the node at this time is recorded. Although other nodes may still have a small amount of energy remaining, nevertheless the evaluation is ended on the basis of the first node to fail. Following the definition of network lifetime given in [184], the network is considered operational until the first node fails due to energy shortage.

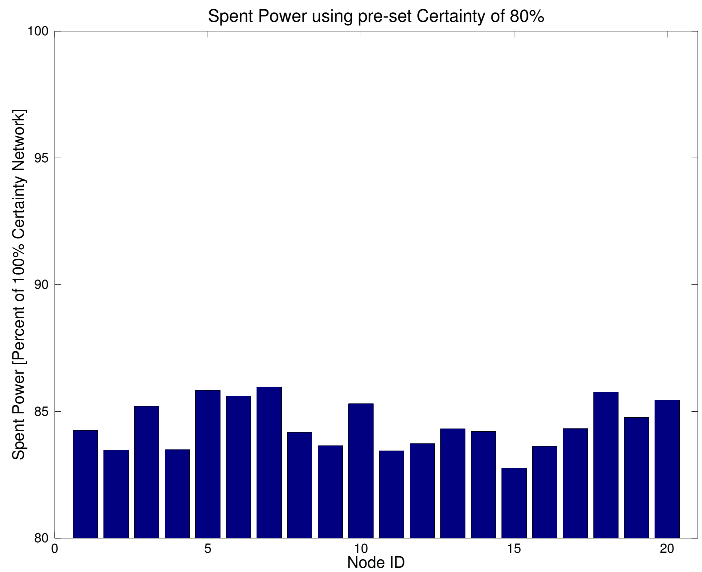


Figure 6.1: Achieved Power Savings with 80% pre-set transmission certainty. The amount of power used compared to a 100% certainty network is shown

Figure 6.3 also shows the energy remaining in a node. The inter-arrival time of messages is equally distributed and each node transmits a message once on the assumption that no re-transmissions are necessary. Limiting message re-transmissions is an advantage of PTS over other schemes. Scenarios in which re-transmissions are not necessarily required are environmental monitoring where the value to be monitored does not changing frequently or it can be easily predicted. Therefore, the base station can manage missing packets by either ignoring a measurement point or upon reception by interpolation of the next message. When a phenomenon is well studied, the base station could predict the value of the missing packet and adjust to the correct one on receiving the next message.

As illustrated in Figure 6.3, the node without PTS drains its battery faster that nodes applying the scheme. Figure 6.4 is a similar plot but does not consider the remaining energy of the sender but instead shows the remaining

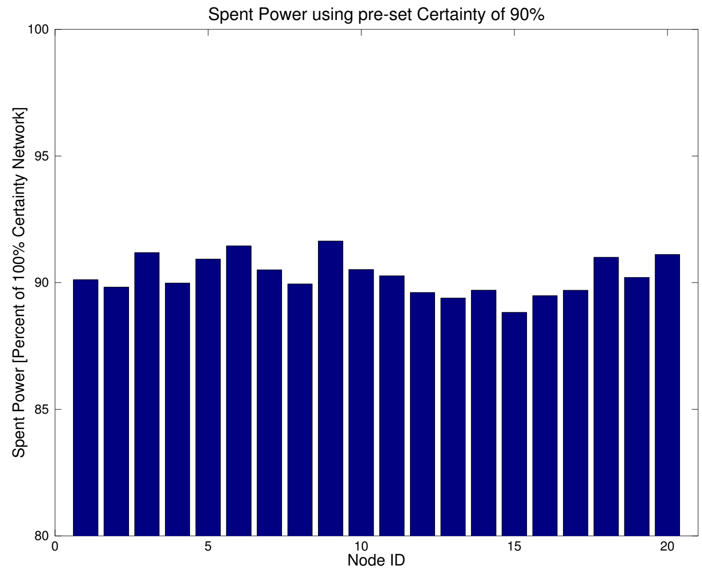


Figure 6.2: Achieved Power Savings with 90% pre-set transmission certainty. The amount of power used compared to a 100% certainty network is shown

energy of the recipients.

Figure 6.5 presents a comparison of the node power consumption for pre-set data recovery success rates of 80% and 90%. As expected raising the reception probability to 90% results in increased energy consumption. This trade-off is the basis for the proposed priority scheme presented in Table 6.2, the flexibility of PTS able to provision pre-set Quality of Service (QoS) levels to any deployment. The user or the application is able to overlay a classification on the data generated and transport it through the network at pre-defined energy consumption level determined by the selected QoS ('blue' line of Figure 6.3).

All nodes are equipped with equal power, two double A-batteries providing an initial charge of approximately 2500 mAh at a voltage of 2.8 V [239]. Transmission consumes the most significant level of energy – sending and receiving data – and for the purposes of the evaluation, this is considered

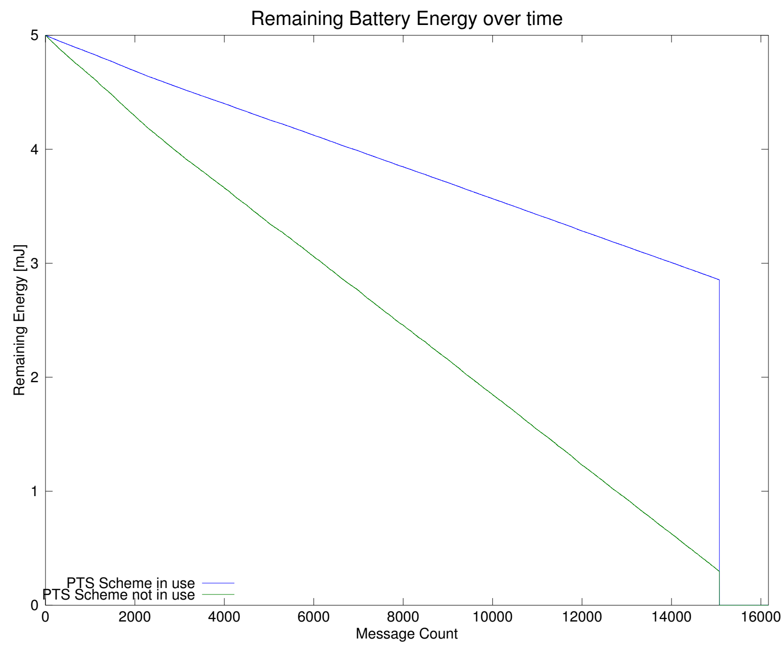


Figure 6.3: Comparison of Battery Draining of a node using PTS and not using PTS

the only source of energy consumption [2]. According to [240] these batteries can provide only 31.25% of their total energy above a voltage of 2.8 Volts, restricting their useful energy to less than a third of specified rating, i.e. from the rating of 27 kJ, only 8.437 kJ are available at a voltage above the required 2.8 Volts minimum for node operation, a consequence of the almost linear voltage drop when current is drawn [241].

The power consumed on transmission is calculated as;

$$P_{Tx}[\text{mW}] \cdot t[\text{s}] \tag{6.1}$$

where t is the time the message is on active transmission, obtained by dividing the message length by the transmission rate (t = 0.024 s or 24 ms). The

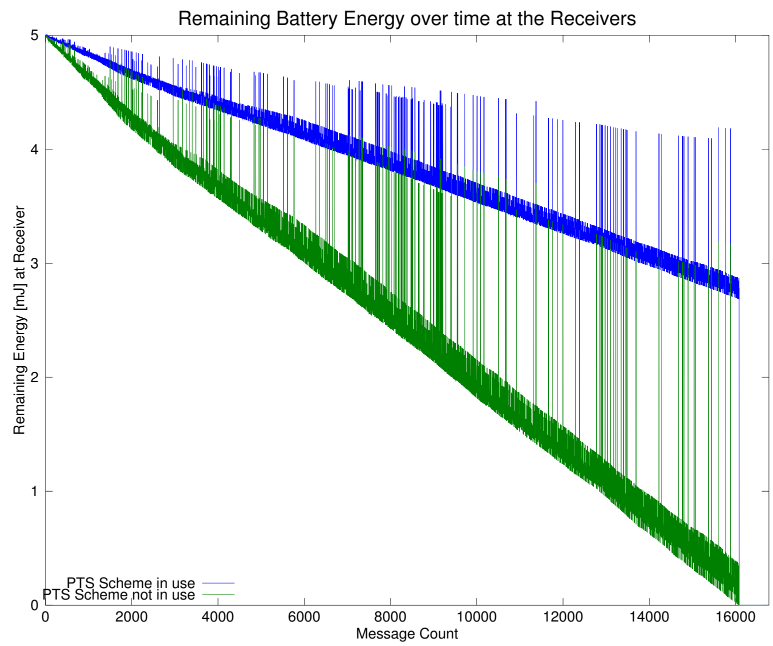


Figure 6.4: Remaining energy at the receivers side

transmission power in mW is;

$$P_{Tx}[\text{mW}] = 10^{\frac{P_{Tx}[\text{dBm}]}{10}} \tag{6.2}$$

The rudimentary evaluation serves to corroborate the impact on a relatively narrow segment of a more extensive Wireless Sensor Network (WSN) implementation; the behaviour of a more realistic network scenario is more meaningful.

6.2 Omnet++ Evaluation of PTS

Following is the detail of the evaluation of a representative WSN applying PTS through OMNeT++, presenting the methodology of the implementation as well as the behaviour trends of network relevant parameters. The parameters

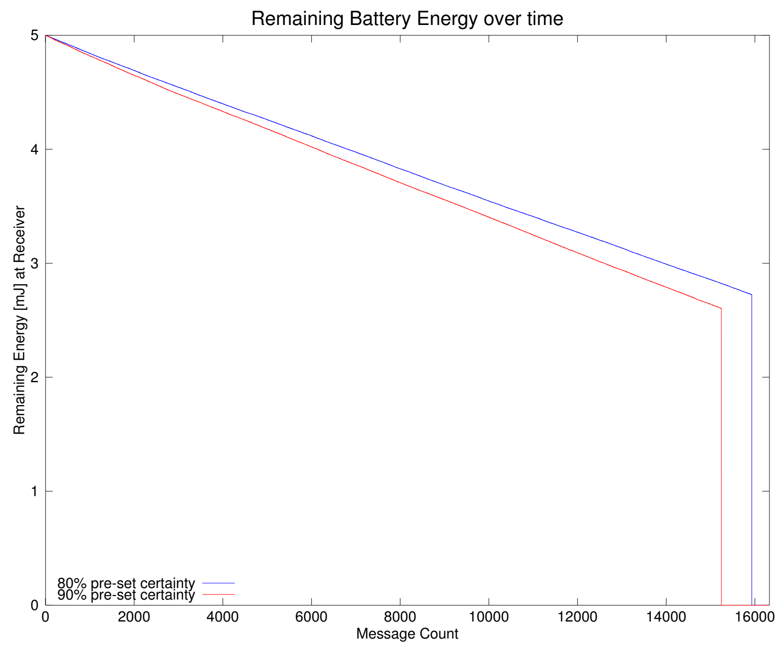


Figure 6.5: Increasing the transmit power to support 90% reception rate

used in the evaluation are listed in Table 6.1.

6.2.1 PTS Implementation

Section 5.2 introduced the cross layer approach to implementing PTS, here implemented in OMNeT++ at both the node and network levels.

The first stage evaluation consists of a network of 20 nodes generating data at random times following Poisson statistics. Two relevant modes of network operation representing real deployed scenarios are appropriate to the evaluation. A node with data so send, can execute either on the receipt of a request (so-called Polling mode [242]) or on a trigger of a locally stored threshold (so-called Push mode [242]). In the Polling mode, the routing layer is aware of the origin of the message and has applied its methodology to route to a designated destination. In the Push mode the routing layer

Table 6.1: PTS Simulation Parameter Set

Parameter	Unit	Value
Simulated Time	[s]	3600
Playground Size	[m^2]	2500
Subscribers		20
Path Loss Coefficient α		4
MAC Queue Length		5
DIFS	[s]	0.006
Slot Duration	[s]	0.01
Contention Window		31
Bit Rate	[b/s]	15360
Busy RSSI	[dBm]	-97
Thermal Noise	[dBm]	-110
Carrier Frequency	[GHz]	2.4
Sensitivity	[dBm]	-96

needs to compute a path first, after which the message is stamped with the next hop node and handed down to the MAC which performs the Ready to Send/Clear to Send (RTS/CTS) handshake. In the Clear to Send (CTS) message is the RSSI within the Ready to Send (RTS) message at the receiver. Additionally the RTS sender – also the CTS receiver – measures the RSSI of the CTS message. These two RSSI values and the pre-set routing of the message to the next hop neighbour are the basis of the PTS implementation (Equation (5.16)).

As stated in [79] energy balancing can be achieved based on routing decisions. Balancing the load either intra-cluster or inter-cluster [104, 243] modulates energy consumption across the network, resulting in a more equally distributed energy consumption which in turn prolongs the node and network lifetime (assuming the definition of [184]).

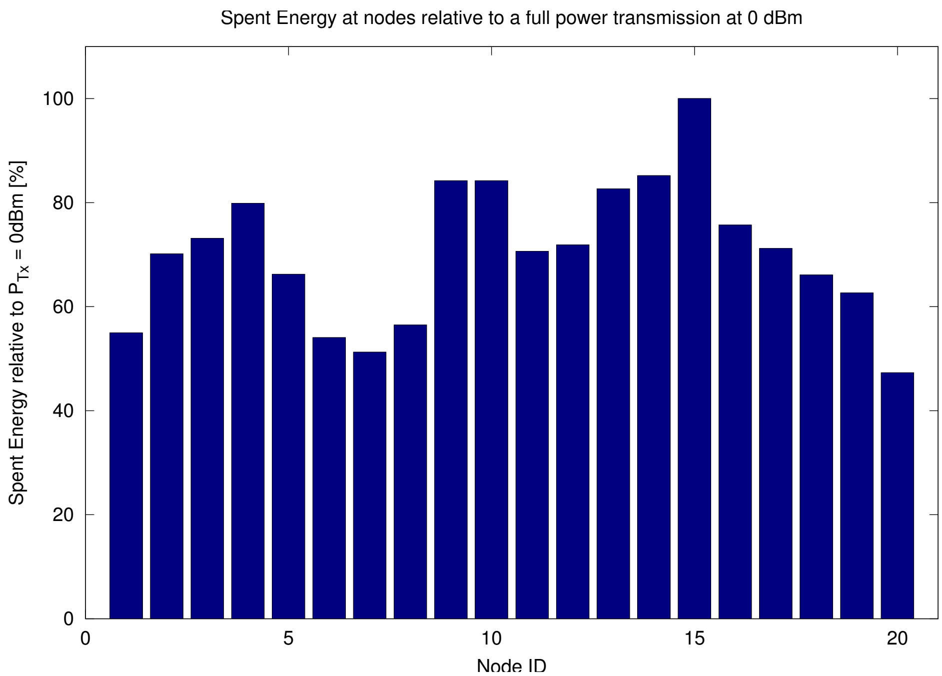


Figure 6.6: Spent Energy with PTS scheme as percentage of transmission made at 0 dBm

6.2.2 PTS Enabled Priority Schemes

[244] state that the transmission power can be and should be controlled for every packet transmission, a power adjustment approach similar to the proposed scheme. However their focus is to enhance spatial re-use only, in so doing increasing network throughput. Furthermore, the scheme is developed for WiFi and has not been applied to WSNs.

The proposed scheme predicts the transmission rate depending on the measured RSSI and node movement. The RSSI considers channel conditions and distance between the two nodes and owing to movement the required transmission power is adjusted further, not taken into consideration in the work of [244]. The dimension of overlaying different QoS levels for data delivery was not considered; the differentiation is a valuable extension with priority levels set for every message.

Table 6.2: Scheme of different priority levels to differentiate messages

	Name	Delay [s]	P_{Tx} [dBm]	Priority
Type 1	DATA MESSAGE	+0.0	−0	0
Type 2	STREAM MESSAGE	+0.0	−1.3	1

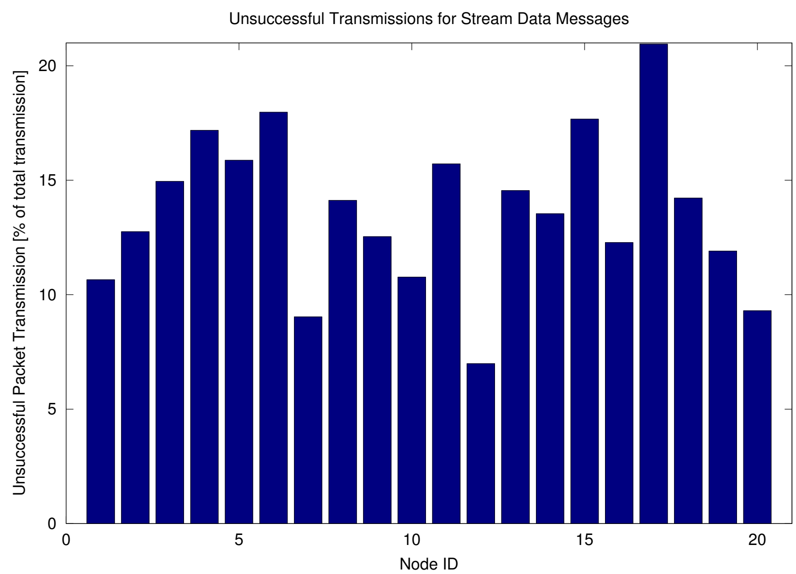


Figure 6.7: Relative number of lost packet throughout the simulation run

In the following evaluation two different classes of data are considered; streaming data, where a packet is generated every second – referred to as a STREAM MESSAGE – and a more randomly distributed message type – referred to as a DATA MESSAGE – generated according to `uniform(8, 12)`, a packet every 8 to 12 seconds. Nodes move following a direction update and an update of speed every 0.1 sec according to a Markov-Gauss Random Walk Model. Updates in speed and direction are incremental.

Data are distinguished by either their `msg->type()`, or by `msg->name()` property, whichever is more appropriate. If the data is deemed to be a DATA MESSAGE, the power is then adjusted according to PTS with the transmission probability pre-set to 100% to ensure successful packet delivery. If the message is a STREAM MESSAGE, the probability of a successful reception can be relaxed to 80%; the concomitant reduction in the transmission power is 1.3 dBm, a 26% energy reduction (Figure 6.7).

However the saving in energy cannot be considered in isolation and the impact on packet loss is crucial to the application. Figure 6.7 shows the relative number of lost packets compared to the overall number sent for type STREAM MESSAGE.

In applications where packet loss is unacceptable, it is most readily managed through re-transmissions which translate into additional energy consumption. Thus the net effect on node energy consumption is the same as the energy conserved through adjustment of the transmission power is used for re-transmissions. Nevertheless in this scenario the PTS scheme is still beneficial in improving the spatial re-use, enabling more communication paths per square metre (see Figure 6.12).

Figure 6.8 shows the power saved with and without PTS with respect to network lifetime, which increases to 134%, assuming that the saved energy is due to reduced transmission power of individual packets only. The energy saving must however be related to the context of the application. If the application relies on the reporting of regular message updates e.g. the humidity of a monitored area or the activity/mobility pattern of animals, the impact of a lost packet can be managed. Schemes can be implemented at the base station to militate against lost packets e.g. missing data can be delivered in the next packet received [245] or an interpolation can be executed to recover the missing data [246]. A scheme or application relying on acknowledging the reception of the data packet implies the same cost penalty at the node whether PTS is implemented or not.

On average, the energy saved at every node is slightly less than 20% with PTS compared to the original transmission power level (Figure 6.8).

The principle of adjusting transmission power to levels based on a pre-

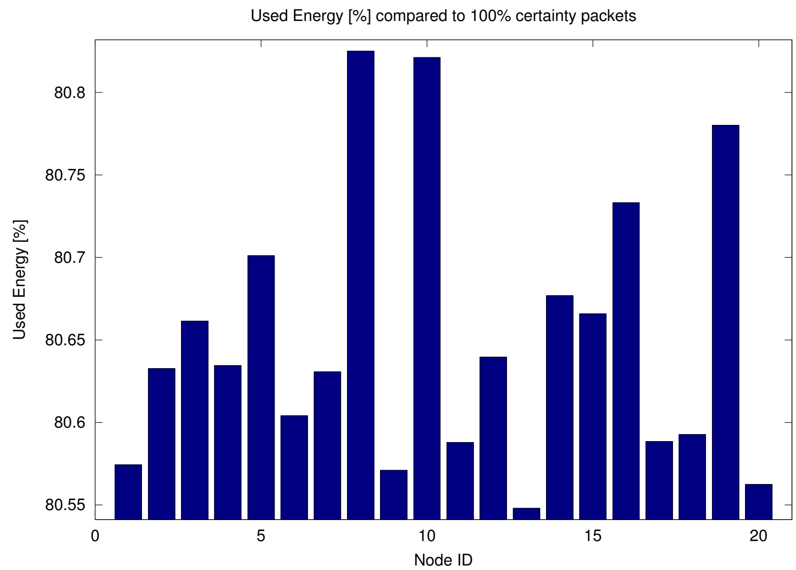


Figure 6.8: Saved Energy achieved by the proposed scheme considering all packets are send

defined probability of a successful data recovery introduces flexibility in the implementation of applications. A range of trade-offs between transmission power and the probability of data recovery result e.g. excessive reductions in the power transmitted results in significant packet loss which offers selection options for the user e.g. the applications can tolerate a certain level of packet loss or the saved energy is spend on re-transmissions. Depending on the application/service re-transmission may not be necessary as the application may rely on since aggregated information over a certain period of time. In these scenarios, the result of data that is transmitted successfully is significant energy savings, especially if re-transmissions are kept to a minimum.

In some cases, even if an application is sensitive to any packet loss (100% probability of successful reception), some level of energy saving can be secured (Figure 6.9). The level of saving is highly dependent depend on the transmission power set for an application (not using PTS). To ease implementation complexity, often all nodes in a deployment operate at maximum output

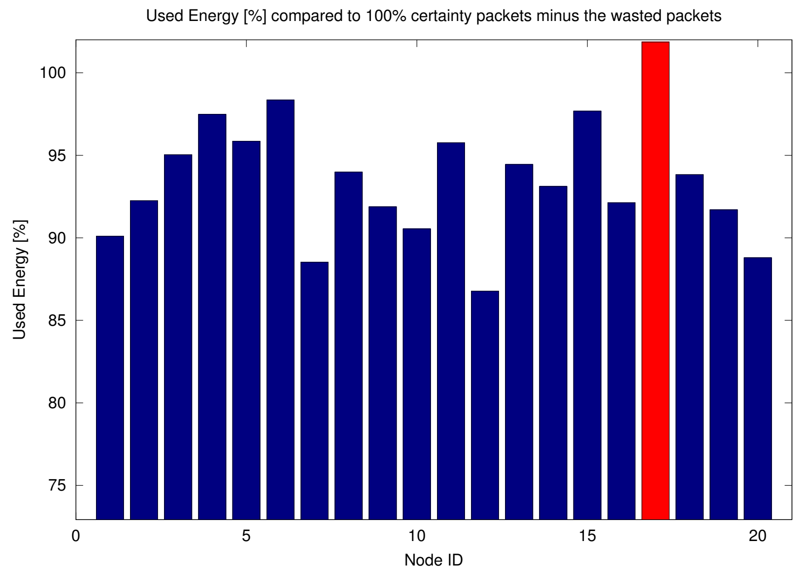


Figure 6.9: Spent power on transmission of packets with pre-set reliability of 80% instead of 100%

power, even for nodes in close proximity to next hop neighbours. Thus in some segments of the overall network, the transmission power levels far exceed those required for successful data (some up to 20 dBm). Any intelligence in the node in respect of adjusting the transmission power for optimum reception will yield savings in energy consumption.

Note that Node 17 is experiencing more lost packets since it is situated at the edge of the network and consequently needs to increase its transmission power to a higher level. The node is also connected to 6 other nodes, resulting in a high traffic load (Figure 6.10).

6.3 Network Lifetime

WSNs implementations are designed to operate autonomously without human intervention for as long as possible. Thus network lifetime is a major challenge and the proposed scheme sits squarely in the role of increasing the network

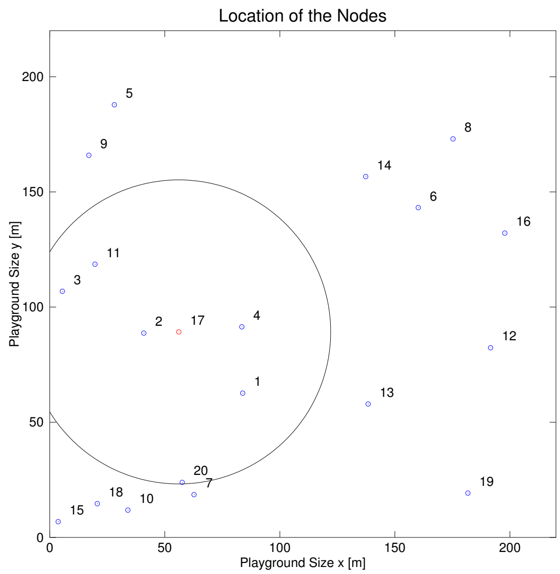


Figure 6.10: Node Distribution on Playground

lifetime whilst at the same time minimising the impact on data delivery. Furthermore, there are a number of additional network benefits to PTS such assigning priority as well as increases in spatial re-use [244].

For the simulation of network lifetime, a script is used to monitor the remaining power in the nodes' battery for the node distribution depicted in Figure 6.10. The nodes displayed in Figure 6.10 select a communication neighbour randomly. The MAC adjusts the transmission power irrespective of the next hop selection process and focuses on establishing successful message exchange only. PTS is responsible for setting the lowest application acceptable transmission power. Initially the batteries hold a charge of 2500 mAh i.e. a current of 1 mA for 2500 h and multiplying the charge with the voltage of the two batteries connected in series, the available energy is calculated to be 27 kJ. Recall that according to [247], since the usable energy in a pair of AA batteries is ≈31.25% of rating delivered at a voltage above 2.8 V, the usable energy to 8.4375 kJ. [240] reports on the current drawn for every

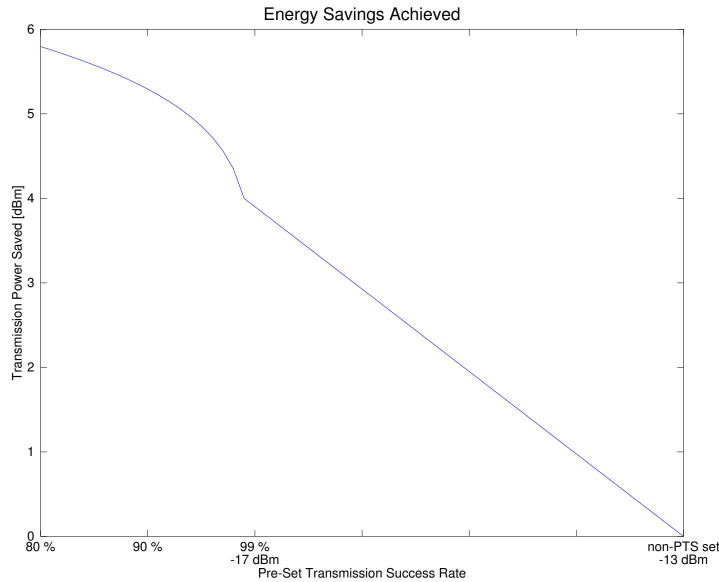


Figure 6.11: Power Savings achieved due to Transmission Power Adjustment of PTS

transmission, reception and measurement executed by a node. The current drawn multiplied by the time taken by each task is subtracted from the initial battery charge.

The network is considered viable until the first node fails and confining the fail mode to battery drain only, the PTS scheme is evaluated for a transmission power of −5 dBm; the increase of the lifetime is ≈6%. A further 5% - 11% in total increase in lifetime is attained for a network where the transmission power is 0 dBm (Figure 6.11).

In a WSN application transmitting data every second, the available energy is consumed in less than 9 months. The implementation of PTS increases network lifetime by 22% to ≈11 months, at the expense of packet loss which increases by ≈11%. Nodes without PTS successfully transmit 25,846,338 packets whilst 28,708,386 are successfully transmitted with PTS.

6.4 Geographical Assisted Fidelity Comparison (GAF)

A performance comparison of PTS with the Geographical Assisted Fidelity [124] technique, viewed as one of the most effective in terms of energy savings in WSNs is presented.

Geographical Assisted Fidelity is a location supported, geographically-informed, energy conservation scheme which aims to prolong network lifetime. The principle is to turn off the radio at nodes not integral to the routing of packets. Consequently Geographical Assisted Fidelity (GAF) creates clusters, so called virtual grids, established by nodes in close geographical proximity. Nodes situated in the same grid negotiate which one is going to 'sleep' (turns off its radio) for an agreed period. After a sleep phase, nodes re-activate radios to re-negotiate based on application and system information. In order to do so, the nodes need to determine a Global Positioning System (GPS) location which in turn governs connections between nodes. [248] argue that network lifetime can be decreased significantly, a strong function of the GPS duty cycle. GAF assumes a 33 mW penalty for GPS communication and only one node per cluster is required to forward messages to the next cluster.

In a negotiation interval, nodes decide which one is to be powered to forward packets; all other nodes in the cluster can turn off the radios for a negotiated period of time and therefore preserve energy, a decision mainly based on the remaining power of a node during in so doing guaranteeing a degree of load balancing. The definition of network lifetime in [124] differs from that of [184]; the network is assumed to be functional as long as data can be routed through the network. The definition also differs from the present work which assumes network is compromised when one node fails (Section 3.4). If the main consideration is that a message can be routed through the network, the nature of WSN inherently allows operation with a certain number of node failures. The drawback of such a definition is compromised coverage e.g. a failure of one node in animal conditions monitoring applications is unacceptable. Therefore the definition of network lifetime is adopted from [184].

The authors argue further, that energy usage for 'idle:receive:transmit' has a ratio of 1:1.2:1.6 [249], only true for a fixed transmission power. GAF is also not suitable for a range of applications as the operational environment has an impact on the protocol performance; PTS is able to adapting to the channel condition (Section 5.3).

The GPS information central to the approach represents the most significant energy consumption at nodes. The assumption that nodes that generate traffic are not subject to failure due to energy consumption also restricts the application domains. In PTS, every node is assumed to generate traffic, and all nodes are responsible for forwarding packets if required. Also, GAF implements a 2 Mbit/s data link, a marked difference to most WSN link rates. Nevertheless a comparison of PTS with GAF is still worthwhile as long as parameter such as data rate and available power source are constant.

The advantage that accrues from PTS scheme is achieved not by sleep modes but through informed adjustment of the transmission power level driven by the application or the priority of the packet. All nodes fully participate as sources of information and in the delivery of data, reducing the link lengths between communicating nodes. GAF uses GPS to determine the absolute position of the nodes; PTS uses the relative position of two nodes at the time they are about to exchange data. The latter can also be used to treat mobile node scenarios. RTS/CTS handshaking generates information about the direction of node movement allowing the dynamic adjustment of transmitted power to compensate for the increased distance.

6.4.1 PTS Evaluation with GAF Parameters

In order to compare the energy saving performance of PTS with GAF, the GAF parameters of [124] are used in the simulation (Table 6.3). The GAF

Table 6.3: Simulation Parameters

Parameter	GAF	PTS
Packet Size	4096 bit	4096 bit
Packets per s	30	30
Bit Rate	121 kbit s^{-1}	121 kbit s^{-1}
Initial Charge	2.5 Ah	2.5 Ah
Send Power	0 dBm	scaled as required
Receive Power	$0.75 \cdot P_{Tx}$	$0.75 \cdot P_{Tx}$
Idle Power	$0.625 \cdot P_{Tx}$	$0.625 \cdot P_{Tx}$
GPS Communication	0.033 Ws^{-1}	—

data rate is not readily achievable in WSNs; de Meulenaer et al. achieved an effective data rate of 121 kbit/s for a MicaZ node [250]. The GAF transmission data rate is therefore set to the level that of the MicaZ and its transmission power is set to the same level as GAF in transmit mode. The power required receiving data and the power consumed in the 'sleep mode' are as given in [124]; nodes not participating in communication tasks consume 0.025 times the energy.

6.4.2 Performance Comparison

Although, the reported GAF evaluation assumes a different definition, in order to make the comparison valid the definition of network lifetime is as adopted by the current work. The same level of available energy resource per node is assumed. Further, the consumption owing to transmission is assumed to be the 33 mW attributed to the need to determine node position from GPS.

The network lifetime for GAF is evaluated to be 1.13 days whilst under the same conditions, for PTS it is 1.02 days. Under low loads, GAF conserves a significant amount of energy by placing nodes into sleep mode. However as the load increases, the number of nodes in sleep mode decreases or else the throughput of the network is compromised.

Due to the limits in GPS accuracy, the forming of clusters is difficult in mobile scenarios as in some cases many nodes occupy the same cluster. Scaling down the cluster size is only viable if GPS accuracy increases. [251] assumes an accuracy of 10 metres, which translates into a minimum cluster size of ≈100 m^2. In the worst case scenario, two cluster heads are more than 28 m apart; due to the limited resources available within cost effective WSN implementations, this transmission distance is problematic and in the extreme, requires nodes to transmit at full power, at the expense of all other transmissions within the grid.

Figure 6.12 shows the relationship between transmission power and coverage of the transmitting node; coverage evolves proportionally to r^2. In relation to the positional measurement accuracy, a latency of up to one second is assumed which further increases the error on the exact position. A node may be associated with one cluster at any given instant which might change within the latency period to another neighbouring cluster. In the extreme, the node may not even join neighbouring cluster.

Another meaningful metric of performance is the number of bits per square metre that are communicated. GAF transmits at constant power. For a transmission power of 0 dBm, the number of bits transported as a function of the coverage area with PTS is shown in Figure 6.13; the y-axis shows the number of additional bits transported and the x-axis shows the fraction of area covered.

Overall, GAF provides an advantage at low traffic loads, recognising the overhead of energy consumption owing to GPS localisation. The advantage erodes at modest to high traffic loads. On closer inspection of the characteristic, reading the figure from the right, it shows that the ratio is 100 i.e. both protocols cover the same area. PTS adjusts the transmission power for each packet dependent on the distance between two nodes and in cases of

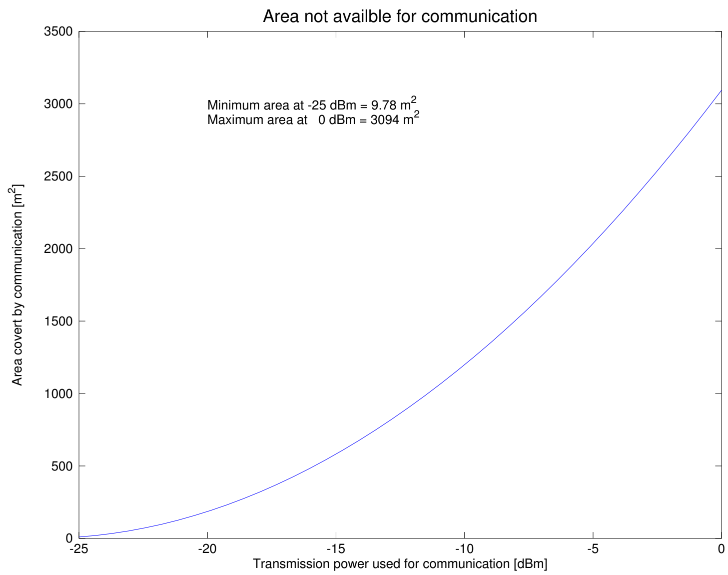


Figure 6.12: Spatial re-use that enables additional parallel communications

close next hop neighbours, can lower the transmission power level. As a consequence an increase in the number of nodes that can exchange data packets within the same coverage area results, effectively increasing the number of bits successfully exchanged.

Figure 6.14 shows the network lifetime for both GAF and PTS as a function of increasing traffic load. At a load of 1 E the channel is fully occupied and no benefit accrues from GAF sleep modes whereas PTS continues to adjust transmission power to the required level and therefore optimises energy consumption.

The additional power usage overhead for GPS derived location information and cluster-head negotiations, enable PTS to outperform GAF at modest to high traffic loads. Adaptive power setting offers many advantages with neighbouring nodes able to communicate at limited interference. Furthermore, nodes are not allocating resource on cluster head negotiation nor on localisation. The sleep mode is also an option for PTS; turning off radios at nodes

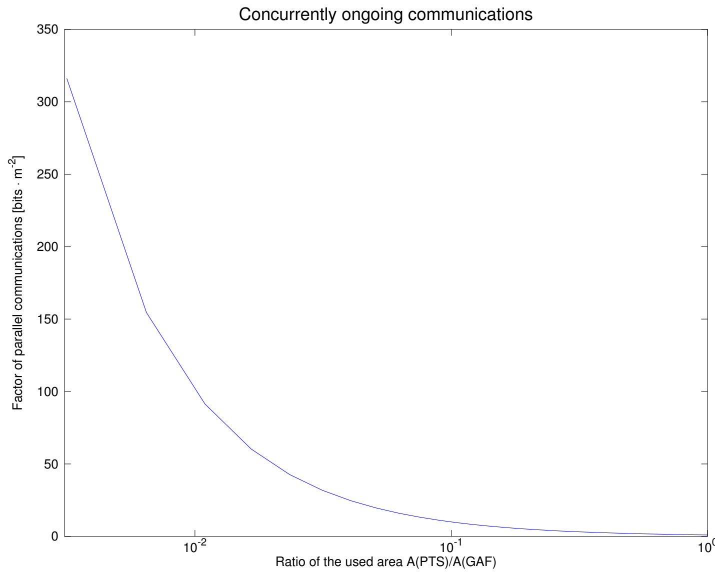


Figure 6.13: Fraction of Bits per m^2 that can be communicated using PTS

with no data to communicate is compelling and a fruitful subject for future studies, bearing in mind the impact on network topologies and routing paths. Furthermore, if all nodes change state at the same time, a peak-load surge is presented to the network that is difficult to manage. Many messages may not be received at the first attempt and re-transmissions consume significant energy.

6.5 GAF Extensions

[252] reports on an enhancement to the operation of the initial GAF protocol by subdividing the coverage area in terms of hexagons, rather than squares. Furthermore, cluster-head selection considers not only residual energy but also the position within the group. To balance energy consumption, the head is located the middle of the cluster, further refined on consideration of the residual energy. Furthermore, movement of the centre location of

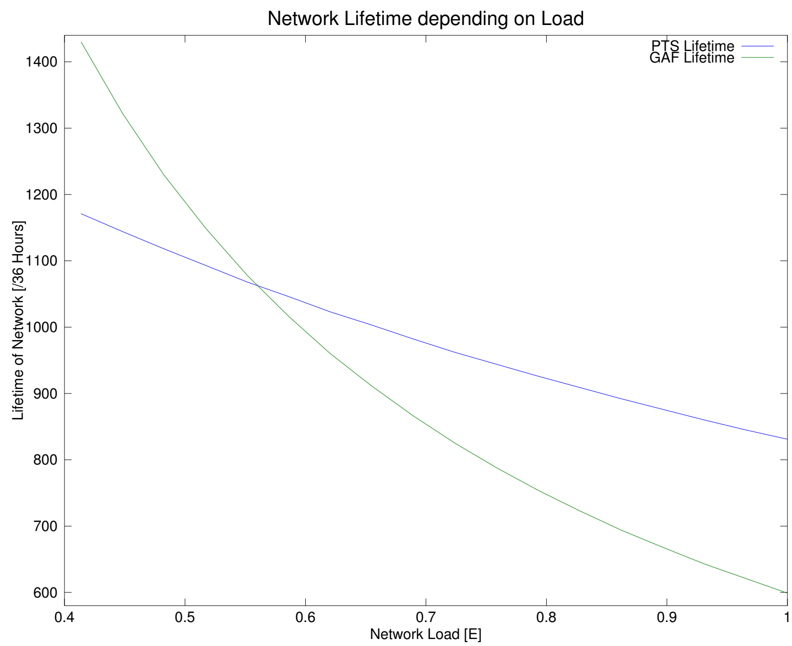


Figure 6.14: Lifetime Comparison of GAF and PTS based on the load offered to the Network

the grid is proposed periodically so nodes located at the cluster border and therefore not likely to be considered in the selection of the cluster head, are included. The shift of cells requires synchronisation of all nodes, not considered within the methodology. The enhancement increases network lifetime but does not address the need for determining node location. Also although nodes can decrease their transmission power, nodes near the cell border still interfere with neighbouring cell transmissions. In the extension, nodes still adopt the sleep mode which decreases connectivity and changes the network topology. After a cluster head selection phase, all elected cluster heads need to handshake to enable forwarding.

To overcome crosstalk/interference, reduction of the grid size of the cells is suggested despite the core location information provided by GPS. Nodes are also assumed to be stationary, limiting the applicability of the protocol. The assumption that nodes in the same grid have redundant measurements is also not true for a range of applications e.g. a node attached to one animal does not acquire any information about another within the same grid. PTS fulfils the need for the transport of data from individual, mobile nodes whilst reducing the energy consumption through intelligent control of the power transmission levels. Also, PTS has the inherent flexibility to increase the transmission power if high priority data alerts from any node need to be sent to the base station or end user.

6.6 Conclusions

In this Chapter a performance evaluation of PTS is executed using the OMNeT++ simulation environment. The metric adopted to evaluate the scheme and demonstrate its advantages are energy saving per message through adjustment of the transmission power per node, the energy savings achieved throughout the network and the network lifetime.

The performance of PTS is compared to the Geographical Assisted Fidelity scheme; both aim to reduce the power consumption of mobile nodes. Although both schemes rely on location information, GAF uses Global Positioning System whilst PTS on the other hand does not require an exact position but the relative node positions derived from the Received Signal Strength Indication measurement at no additional cost. GAF assigns nodes to clusters based on position and places all nodes not elected cluster head into sleep mode. Turning off the radio saves significant levels of energy but is only viable when the nodes are not required to participate in neither the process or measurement task nor maintaining active communication paths during sleep time. In addition the nodes might move during the period of the sleep time and may re-activate their radio in a different grid cell. In order to synchronise to the negotiated transmission plan peculiar to that certain cell requires a

Table 6.4: Power Consumption over offered Network Load

Load [E]	Power [mW]
0.42	105.56
0.5	120.68
0.57	138.35
0.64	158.56
0.71	181.31
0.78	206.61
0.85	234.44
0.92	264.83
1	297.75

communication overhead.

In contrast PTS derives a measurement of the relative position each time a RTS/CTS handshake is carried out. Embedding additional information about the direction of nodes in the RTS message and returned in the CTS, provides the MAC with the information to adjust the transmission power level. Depending on the grid cell size, nodes operating GAF utilise the maximum transmission power level. Based on square grids, node links up the distance d of $d \leq R\sqrt{5}$ with R being the squares side are established.

Reliance on GPS information for location makes GAF unsuitable for indoor scenarios; even in an urban environment GPS precision is compromised due to multi-paths. This is a notable PTS advantage as the channel model is embedded in the transmission power adjustment calculation.

The PTS comparison to the GAF scheme is based on [124], ensuring that the network environment both scheme are simulated under the same conditions. The simulation parameters are listed in Table 6.3, the major highlight is the energy penalty for GAF's GPS communication overhead. All nodes are equipped with the same energy source and the simulation is ended according to the lifetime definition in Chapter 3.4.

As traffic load increases, the number of GAF nodes permitted to enter

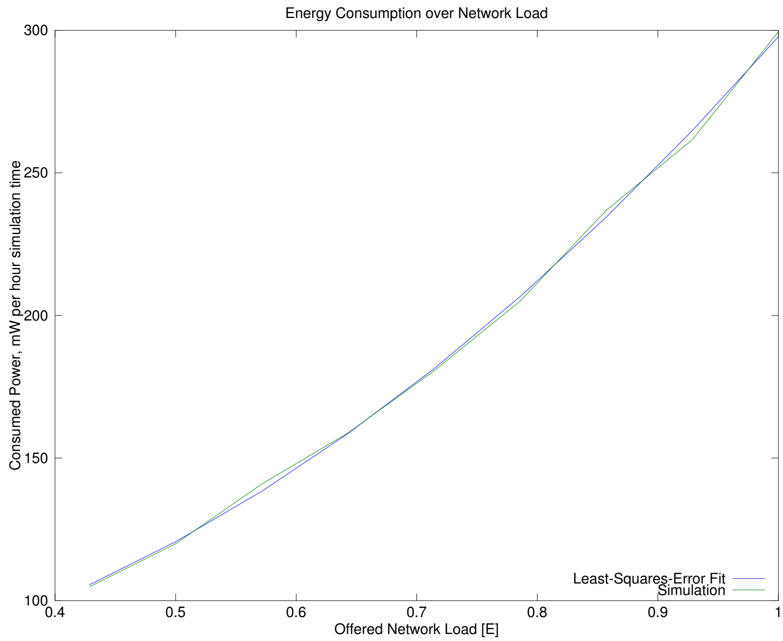


Figure 6.15: Increase of Power Consumption over Offered Network Load

Table 6.5: Advantages of PTS over GAF

	PTS	GAF
Mobility	Included	Using Extensions
Scalable T_{Tx}	Per Packet	Pre-Set
On the side Implementation	Yes	No
Positioning	Locally	Globally
Lifetime over Load	Decreasing	Strongly Decreasing

the sleep mode decreases and consequently the massive advantages in terms of energy consumption owing to turning off the radio diminishes whilst PTS maintains the improvement. Table 6.4 shows values of power derived from simulation results; the power consumption is logged as network load is increased. Furthermore, GAF nodes are set to the maximum transmission power irrespective of the distance between communicating nodes. Figure 6.14 shows the erosion of the advantage owing to the sleep mode as a function of increasing network load. The increased power consumption is displayed in Figure 6.15. Other advantage and disadvantages such as mobility support and scalability of transmission power are summarised in Table 6.5.

A second measure, spatial re-use which describes the amount of data that can be communicated within a certain area $\left(\frac{\text{Bytes}}{\text{Metre}^2}\right)$ is highlighted as an additional advantage of PTS. The metric indicates the amount of data that can be communicated per square metre on top of energy savings achieved. Table 6.6 summarises the trade-off between the coverage area and transmission power (Figure 6.12).

Another transmission scheme that relies on transmission power control is the Adaptive Transmission Power Control (ATPC) [20], using beacon messages to maintain link state models. After an initial setup, links are maintained through frequent update messaging. Unlike PTS, ATPC's underlying model is based on experimental results; PTS is based on the transmission probability of Bits and Symbols for the selected modulation scheme. Whereas ATPC does not adjust the transmission power viz. when the RSSI remains within the boundaries of the upper and lower link quality, PTS adjusts the power for every transmitted packet. ATPC uses a slowly changing model of the link environment to set the transmission power required that, unlike PTS, cannot adapt to rapid changes in virtual or communication distance. Furthermore, PTS exploits the shorter message format inherent to RTS/CTS handshaking to determine the link condition whereas ATPC transmits full data frames at the last recorded transmission power level, only adapting if the link quality is poor or the message is lost.

Table 6.6: Area occupied due to Transmission, PTS, GAF

Transmission Power [dBm]	PTS Occupied Area [m^2]	Transmission Power [dBm]	GAF Occupied Area [m^2]
−25	9.78	0	3093.78
−24	27.32	0	3093.78
−23	53.69	0	3093.78
−22	88.87	0	3093.78
−21	132.86	0	3093.78
−20	185.68	0	3093.78
−19	247.31	0	3093.78
−18	317.77	0	3093.78
−17	397.04	0	3093.78
−16	485.13	0	3093.78
−15	582.03	0	3093.78
−14	687.76	0	3093.78
−13	802.30	0	3093.78
−12	925.66	0	3093.78
−11	1057.84	0	3093.78
−10	1198.83	0	3093.78
−9	1348.65	0	3093.78
−8	1507.28	0	3093.78
−7	1674.73	0	3093.78
−6	1851.00	0	3093.78
−5	2036.08	0	3093.78
−4	2229.99	0	3093.78
−3	2432.71	0	3093.78
−2	2644.25	0	3093.78
−1	2864.60	0	3093.78
0	3093.78	0	3093.78

Chapter 7

Conclusions and Future Work

The research centres on the development of a power saving scheme that enables nodes within an extensive Wireless Sensor Network (WSN) to preserve energy and increase network lifetime. The foundation of the strategy adopts the principle that since WSN nodes consume the most significant amount of energy on the transmission of data, a potentially viable approach is to develop a mechanism that preserves scarce energy resources by adjusting the level of power transmitted to an optimum for connections to next hop neighbouring nodes.

The proposed Predictable Transmission Success (PTS) offers noticeable potential for saving energy within WSN application scenarios. The main advantage of the proposed scheme is that it can be applied to almost every Medium Access Control layer (MAC) schema employing a Ready to Send/Clear to Send (RTS/CTS) handshake. The overhead owing to PTS extensions within the RTS/CTS message exchange is negligible since only eight additional Bytes are required to designate the Received Signal Strength Indication (RSSI) information. The RSSI is embedded in most WSN implementations, executing the measurement routinely at the hardware level. This is the major advantage of the proposed scheme since minimum of complexity is required to deploy the solution in new and existing platforms. The scheme is also agnostic to the hardware since it can be overlaid into any MAC implementation.

The Thesis starts with a review of the state-of-the-art protocols at different layers of the Open System Intercommunication (OSI) stack. Informed by reported schemes and implementation strategies, PTS is developed as a cross-layer power adjustment scheme capable of predicating the communication link distance and designed in a manner that can be easily overlaid onto existing/implemented schemes. The principle centres on a per packet dynamic adjustment of the transmitted power based on an evaluation of the RSSI at the node and the direction of movement of the relevant receiving node. The link distance is derived from the RSSI and in tandem with the setting of a probability of successful data recovery, PTS adjusts the transmission power level so that trade-offs can be established between the packet loss, coverage area and network lifetime.

Current protocol implementations often suffer from competing goals but the proposed PTS scheme offers a number of important advantages and functionalities;

- adjustment of transmission power per packet
- message prioritisation
- manages node movement
- can be retro-fitted to existing deployments
- adapts to prevailing channel conditions

PTS is designed to be compatible with existing WSNs, not compromising their operation and deployment. Nodes deployed randomly are free to establish a network structure in an ad-hoc manner, determining the next-hop neighbour to forward messages in the most efficient way that meets the needs of the application. Indeed the core RSSI information is routinely generated during any set-up phase, and subsequently empowers each individual node with the capability to adjust the transmission power according to the coverage distance and/or the priority assigned to the message by the user or application. As well as adjusting the transmission power to the optimum for the link distance, packet priority, connection quality, the nodes can continually and dynamically adjust the transmission power in response to a changing environment and therefore channel condition.

The development is inspired by activity monitoring of individual animals application viz. free roaming cows in typical agricultural environments. Timely – quasi-real time – information about the cow conditions provides the farmer with valuable knowledge on the condition of the herd, yielding increased revenues from more directed operational interventions. The application has demanding requirements which have driven the development of PTS; however it must be noted that these requirements are not solely relevant to this application but are also characteristic of a range of other applications.

WSN applications demand high levels of autonomous behaviour for as long as possible. In addition the solution has to treat many nodes – thousands per base station – that move randomly in and out of the coverage area with a lifetime as long as the application without material manual interventions beyond the initial deployment e.g. no battery replacement. PTS promotes these features, extending network lifetime, unattended.

PTS manages two scenarios in an effective manner; networks comprising mobile nodes and dynamically varying channel environments. In respect of the first scenario, nodes exploit the RSSI derived during the RTS/CTS handshake to determine their relative mobility e.g. a decrease in the RSSI in the Clear to Send (CTS) message indicates that nodes are moving apart. Utilising that knowledge, the transmitting node can predict the distance between itself and its communication partner when the transmission is finished. This, maximum, link distance is used in turn to set the transmission power, ensuring a successful connection and removing the need for the re-transmission of data. In the extreme, the combined period of Acknowledgement (ACK), scheduling of the re-transmission and sensing the channel idle, may result a new link distance between the sender and intended receiver that cannot be bridged.

In this case an intermediate node is required to forward the message.

In the second scenario, the distance between two nodes is assumed to change at a timescale during which the channel characteristic does not alert appreciably during an on-going transmission. The term 'virtual distance' is introduced to describe the scenario where an obstacle impairs the established path. It encapsulates the effective increase in distance between two communicating nodes owing to the obstruction in relation to the unimpaired line-of-sight distance, manifest as a decrease in the RSSI. The nodes are agnostic to these impairments since a decrease in the RSSI can be corrected through the adjustment of transmission power to bridge the effective link distance.

In both cases a small overhead of one byte results, required to inform the communication partner about the change in RSSI. The transmitting node is not obligated to gather information on the operating environment, as the core is the RSSI which has embedded in its value, the prevailing channel condition.

Furthermore, the implementation is a straightforward software upgrade and relies on the establishment of a look-up-table. A range of transport tasks are managed by the scheme, since priority levels can be assigned to each class of data by the application; the MAC implements the priority set by the application layer. Recurring DATA STREAMs are assigned a lower priority than an alarm/alert that needs a response quickly. Furthermore, at the data sink such as a computer, the system allows for further data processing for example interpolating missing measurements and extrapolation of future measurements [246, 245].

Improvements in energy savings are evaluated through mathematical models and further corroborated through extensive analysis of the impact of the approach utilising simulations. The simulation environment is validated through comparison with published work and results are produced within representative network/application scenarios.

7.1 Future Work

In the proposed PTS implementation, the MAC manages a table in which a match between the transmission power obtained from the RSSIs during RTS/CTS handshake and link distance is maintained. This table provides the MAC with a power level for transmission with minimum perturbation to on-going transmissions.

The Geographical Assisted Fidelity (GAF) principle of utilising the sleep mode results in significant energy savings. The problem is however the significant and frequent changes to network topology. Nodes need to negotiate when to enter the sleep mode, and when to re-activate for the next time to re-negotiate. These periods of negotiation represent a significant maintenance overhead and precise synchronisation. However a relative mode position between two communicating nodes based on RSSI measurement might be a sufficiently precise positioning to apply a sleep mode dimension to PTS.

Routing decisions can also be based on energy considerations upon the CTS relay between forwarding nodes. If the routing decision is based not only the optimal route but also on the remaining power at the forwarding partner, energy consumption can be balanced more equally throughout the network.

For mobile nodes, a generalised coordinate system can be introduced so that movement can not only be predicted by a Constant Turn Rate and Velocity (CTRV) model but rather be tracked with more advanced techniques such a Kalman-Filters. The required overhead in computation and communication is to be held against the potential additional savings.

7.2 Evaluation Framework

The automation of simulation runs can benefit greatly from the development of a Graphical User Interface (GUI) that allows the easy setting of all parameters, especially when configuring the `replicate.m` script that speeds up the process of establishing simulation runs. Scripts to switch through simulation environments (Command Line (CMD), TK [253]) and to disable/enable debugging make the interaction more efficient. Furthermore, the collection and evaluation of results can be automated using Octave. With knowledge of the format of the generated output file, it is possible to attach more scripts that read the result files and generate graphs. These plots could then be attached to an email, removing the need for the user to log on in order to retrieve graphs.

Appendix A

Acronyms

BS	Base Station
WSN	Wireless Sensor Network
IDE	Integrated Development Environment
ISO	International Organisation for Standardisation
OSI	Open System Intercommunication
QoS	Quality of Service
NIC	Network Interface Card
MAC	Medium Access Control layer
RL	Routing Layer
ACK	Acknowledgement
Rssi	Received Signal Strength Indication
CS	Carrier Sense
MA	Multiple Access
Csma	Carrier Sense Multiple Access
Csma/CD	Carrier Sense Multiple Access/Collision Detection
Csma/CA	Carrier Sense Multiple Access/Collision Avoidance
Cdma	Code Division Multiple Access
Ctrv	Constant Turn Rate and Velocity
CW	Contention Window

IFS	Interframe Space
Difs	Distributed Inter-Frame Space
Sifs	Short Inter-Frame Space
CTS	Clear to transmit
OS	Operating System
GUI	Graphical User Interface
GNU	GNU's Not Unix
VC	Virtual Circuit
BER	Bit Error Rate
RTS	Ready to Send
CTS	Clear to Send
RTS/CTS	Ready to Send/Clear to Send
ISM	Industrial, Scientific and Medical
PDF	Probability Density Function
CDF	Cumulated Density Function
LoS	Line of Sight
RWP	Random Waypoint Model
PTS	Predictable Transmission Success
GPS	Global Positioning System
GAF	Geographical Assisted Fidelity
CBR	Constant Bit Rate
VBR	Variable Bit Rate
UBR	Unspecified Bit Rate
TCP	Transport Control Protocol
TCP/IP	Transport Control Protocol/Internet Protocol
IP	Internet Protocol
UDP	User Datagram Protocol
Tdma	Time Division Multiple Access
Cwnd	Congestion Window
SNR	Signal to Noise Ratio

PL	Physical Layer
IR	Infra Red
ADV	Advertisement message
REQ	Request message
Data	Data message
LA	Local Aggregator
MA	Master Aggregations
SNR	Signal To Noise Ratio
QPSK	Quadrature Phase-Shift Keying
OQPSK	Offset Quadrature Phase-Shift Keying
Tcl	Tool Command Language
STL	Standard Template Library
nam	Network Animator
Tkenv	Tk Environment
GloMoSim	Global Mobile Information System Simulator
NED	Network Description
GNED	Graphical Network Description
NIC	Network Interface Card
CMD	Command Line
Cmdenv	Command Line Environment
ns-2	Network Simulator 2
ns-3	Network Simulator 3
GloMoSim	Global Mobile Information System Simulator
Cbah	Clustering Based Aggregation Heuristic
Sap	Service Access Point
Nsap	Network Service Access Point
Tsap	Transport Service Access Point
MGRWM	Markov-Gauss Random Walk Model
Rdwm	Random Direction Walk Model
ARP	Address Resolution Protocol

Udp	User Datagram Protocol
Aimrp	Address-light, integrated MAC and Routing Protocol
Pmac	Prioritised MAC
Spin	Sensor Protocol for Information via Negotiation
DD	Direct Diffusion
RR	Rumour Routing
Mcfa	Minimum cost forwarding algorithm
Gbr	Gradient Based Routing
Idsq	Information Driven Sensor Querying
cadr	Constrained Anisotropic Diffusion Routing
Aquire	Active QUery in sensoR nEtworks
Ear	Energy Aware Routing
Leach	Low Level Adaptive Clustering Hierarchy
Pegasis	Power-Efficient Gathering in Sensor Information System
Teen	Threshold-Sensitive Energy Efficient Sensor Network
Apteen	Adaptive Periodic Threshold Sensitive Energy Efficient Sensor Network
Mecn	Minimum Energy Communication Network
Smecn	Small Minimum Energy Communication Network
SOP	Self-Organising Protocol
Sar	Sensor Aggregate Routing
Dam	Distributed Aggregate Management
Ebam	Energy Based Activity Monitoring
Emlam	The Expectation-Maximisation Like Activity Monitoring algorithm
Vga	Virtual Grid Architecture
Hpar	Hierarchical Power-Aware Routing
Ttdd	Two-Tier Data Dissemination
Gear	Geographic and Energy Aware Routing
Mfr	Most Forward within Radius

Dir	Compass Routing Method
Gedir	Geographic Distance Routing
Goafr	Greedy Other Adaptive Face Routing
FR	Face Routing
Afr	Adaptive Face Routing (FR)
Dsr	Dynamic Source Routing
Atpc	Adaptive Transmission Power Control

Appendix B

Publications

A Priority Based Routing Protocol for Wireless Sensor Networks

Christian Andrä, Kae Hsiang Kwong, Craig Michie and Ivan Andonovic
Department of Electronic and Electrical Engineering, University of Strathclyde,
204 George Street, G1 1XW Glasgow, United Kingdom
email: { christian.andrae, kwong, c.michie, i.andonovic} @eee.strath.ac.uk

***Abstract*—Recently, the demands on wireless sensor networks have switched from low traffic rate and static topology to more challenging requirements in order to meet the rapid expansion of WSN into various domain applications. This paper proposes a seamless cross layer solution that integrates network layer and medium access control to accommodate some of the new challenges. This new solution allows routing paths being generated dynamically to meet the requirement of potential mobile nodes. Higher data throughput and flow control are part of the new demands required to be addressed urgently. The proposed solution integrates a priority based MAC to handle congestion and packet loss problems which commonly happened in WSN when an occurrence of event spread into wide area.**

I. Introduction

A wireless sensor network (WSN) is a network system that is formed autonomously by a group of sensor nodes which are commonly fabricated by using low cost/specification hardware. The development of wireless sensor networks has evolved from its original initiative and found its way into broader applications. Some of the new applications pose higher demands than conventional WSNs and exceed its original design spaces. For instance, in [1] the design principle has moved from the conventional low data volume to information-rich data applications in which data volume is dramatically distinguished from conventional WSNs. In [2], the WSN technology has been applied into farming industry, where individual animals are mounted with a sensor node that is fabricated into a collar. The location of the sensor node is mobile and it will change whenever the animal moves. Therefore, this new application has broken one of the common assumptions of WSN, where the location of sensor nodes is said to be static.

Cross layer implementation is a common way of efficient protocol implementation, where different layers of the protocol stack exchange information. In the case under consideration, network layer and medium access layer (MAC) are implemented in such a way. MAC and routing layer exchanging information about the nodes status and the message type that is going to be send. This way of implementation looses the strict way of the layered structure of the OSI definition. It makes it easier for the applications to gather information directly from the physical layer (PHY), [3]. As a result, a light-weight protocol referred to as PriBaR is developed. In order to allow this protocol to run and survive in challenging environment during the development process a few principles are followed rigorously:

- i Small in size. The code size has to be in the region of less than 100 kBytes. That was achieved, the total size of the transfered code image of the proposed protocol was 43.6 kBytes.
- ii Low memory usage. The hardware restriction of the MICAz nodes just allow 512K byte of data, [4]. Therefore, a routing table that records all neighbouring nodes might not be advisable.
- iii Low power consumption. Hand-shaking, neighbour discovery, network congestion and packet retransmission all cost power. All these should be kept to minimum.
- iv Autonomous and independent operation. Data transmission tasks (i.e. packet relay, link establishment, routing) should run independently and should not impact the sensor actuation. Each sensor node should be able to operate autonomously as well as part of a WSN. Protocols should not be over-complex and require frequent hand-shaking to establish new links.

The paper is structured as follows. Section II gives an overview of related work. In section III, the PriBaR cross layer protocol is introduced and the experimental setup is described. Following this, section V presents the results of this work. Finally, section VI summarizes the findings of this paper and gives an outlook to future work.

II. Related Work

A depth survey in [5] has outlined that WSNs can effectively conduct some of the networking tasks that require co-operative action from neighbouring nodes (i.e. resource sharing and autonomously networking). WSNs are still lacking capability to perform an integrate network management such as network resource allocation and data loss recovery in which will adapt to current network condition. For example, in [6], [7], [8] novels resource management and channel access schemes have been studied. Whereas, issues related to congestion avoidance and packet loss recovery have been discussed in [9], [10]. Generally, researchers look into these problems in isolation. With a single layer of routing and networking scheme it might not work well when the environment changes. A WSN is deployed commonly in challenging and complex environments. Such environments might be unmanned, impossible to retrieve for manual reconfiguration or simply hazardous.

To the best knowledge of the authors there was no such work, combining the prioritised medium access with a autonomous zone set up. The used AIMRP routing protocol has been studied extensively in [11]. The P-MAC medium access was implemented before at Strathclyde University. The expertise following from this can be used to implement the combination of these both protocols. The expected advantages of a lightweight implementation and prioritised medium access will be explained in more detail in the following section III.

III. Priority Based Routing Protocol, PriBaR

The proposed PriBaR protocol is the first step to developing a fully integrated protocol that equips WSN with traffic engineering module that allows it to perform dynamic routing and network congestion management. This PriBaR protocol is comprised with two parts:

i. the dynamic routing module and
ii. the congestion management module

A unique network topological structure, which dissects the sensor nodes into a number of zones according to their position is required by the PriBaR protocol. The zones are extended, starting from the base station outwards to the farther remote sensor nodes. These zones are organised and identified using the (hop-)distance between sensor node and base station. This information is recorded by each individual sensor node in a unique field - TIER ID. The base station is set to zone 1. If the sensor node is one hop away from the base station it is said that this particular sensor node is located in zone 2 and for all the sensor nodes that are one hop away from the base station the value for their TIER ID is equal to 2. Whereas, if a node needed to forward a packet to the base station and this packet requires assistance of 3 sensor nodes performing multi-hop relays then this sensor node is said to be in zone 5 (TIER ID=5). The number of zones is limited by the variable type it was assigned to, `uint8_t` which allows 256 values starting from zero.

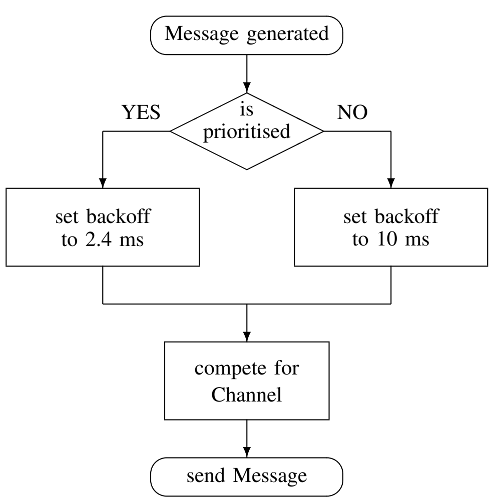


Figure 1. Flow chart; setting initial backoff when competing for the channel

To establish the network topology configuration, a simple passing through mechanism is used. The base station broadcasts a network configuration packet which contains BS ID

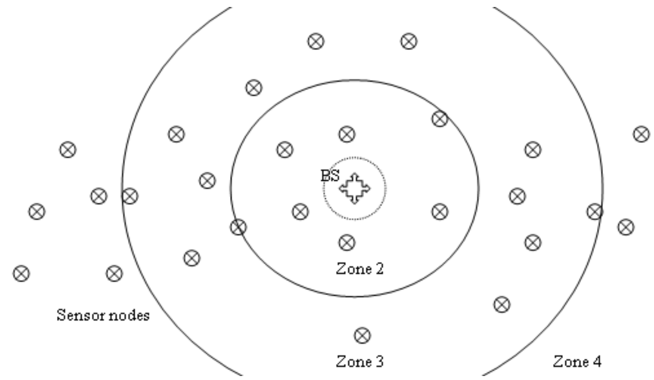


Figure 2. A typical network topology configuration by PriBaR protocol

= 0 and TIER ID = 1. When this packet is received by a sensor node located in vicinity to BS it will know that it is one hop away from BS and set its TIER ID = 2. All other nodes that received the message will do the same. The nodes that received the first broadcast will continue the network configuration process by issuing a configuration packet with TIER ID = 2. This message will be received by other nodes which will set their TIER ID to 3 accordingly. This process then continues until all the nodes are reached and assigned with a corresponding TIER ID. As soon as the configuration packet has been successfully passed through the network topology, the PriBaR protocol can start to route packets back to the BS.

The second part of PriBaR protocol is network congestion management module. In TCP/IP network, congestion management is based on:

i comparison of historical traffic volume [12],
ii response time [12], and
iii packet loss [13]

These approaches not only require large amount of memory and computing power but also reside on top of TCP/IP layer, which associate with an overhead that is not possible to apply in WSN. To mitigate the network congestion, the PriBaR protocol relies on the CSMA/CA medium access control with priority packet support. This module works like this:

i when the channel is too busy random delay kicks in, the nodes have to wait before re-transmission
ii prioritised packets are associated with important or critical data, therefore they will access the channel with higher priority.

Nodes can distinguish between three types of messages. First message type is TIER message. A node that did not receive a TIER message does not know its position in the network, and therefore is not going to take over any forwarding jobs and does not generate traffic itself. This type of message is recognised by the MAC layer since it has a different message type ID and therefore triggers a different function. The flow chart of the triggered function is displayed in Fig. 3. Second type of message is the data message. Initially white-spaced with zeros, the only information in this message is the origin node ID and TIER zone ID, the time of departure from the origin node, the buffer utilisation and dropped messages count at the origin node and whether it is a prioritised message or not. Each intermediate node is filling its information about ID, TIER zone ID, buffer utilisation and number of dropped packets at the first white space it finds next to the information from the previous node. A data message usually is acknowledged on reception. Therefore, the MAC acknowledgement (ACK) is turned on and indicates that the sent message is received from the relay node or base station. On reception of an ACK the transmission is finished. It is prompting the buffer pointer to be increased and therefore sending the next message, if there is one. Furthermore, the resend-counter is reset. The reset counter is used to limit the retries before a messages is going to be dropped. The flow chart of PriBaR's medium access is illustrated in Fig. 1.

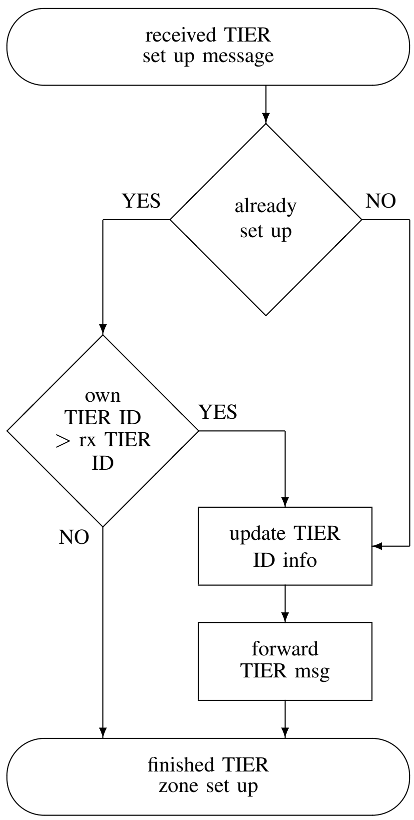


Figure 3. TIER set up flow chart

IV. Experiment Configuration

Main goal of the experimental set up was to create 4 zones each hosting 5 nodes. In order to set up a non-overlapped multi-tier environment, it was impossible within the given environment to set up more than 4 zones, where each zone got a LoS (line of sight) to the next upper and lower zone and every node can receive the time base signal. The experiment required a careful configuration so that the nodes in x^{th}-TIER zone could only communicate to nodes in $(x+1)^{th}$ and $(x-1)^{th}$ tiers but no others. This was done by separating the nodes in different screened areas within the lab; concrete walls are utilised to provide necessary partitioning allowing a number of TIERs being setup in a confined area. Fig. 4 illustrates the experiment configuration. With this set up there was no zone x that had a line of sight (LoS) with neither $x-2$ nor $x+2$.

In the experiment, 20 nodes, always 5 node in one of the 4 zones, were used and one base station node, all based on MICAz [xbow] hardware.

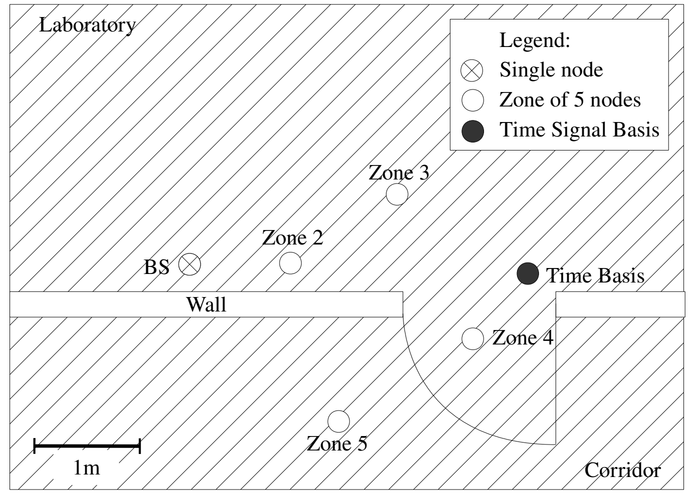


Figure 4. Dimensions of the experimental set up

After the zone configuration was completed, the nodes only performed forwarding task to those nodes that were located in the previous zone, which have a higher TIER ID. This means that, for example, nodes in zone two would not accept a message from nodes in zone four. In the experiment, the sensor nodes did not only act as a relay node, but also as traffic sources. Each node is programmed with an exponential traffic generator, which will generate $\frac{2}{3}$ normal packets per second and 1 priority packet every 60 seconds.

The measurement with this set up will be presented in the following section. The presented figures will all be organised in the same way that they x-axis represents the zone IDs, the y-axes the measured physical quantity, and the different grey scales represent a different node in the corresponding zone. If the value drops to zero in a certain zone and rises above zero in the following zone, this simply means that the physical quantity could not be measured at this specific node.

V. Results

The measurements that were carried out within the experiment include the total delay of both, normal and prioritised messages, the amount of dropped messages at each node, again for both kinds of messages. Furthermore, each node that forwarded a message was stamping its buffer utilisation of normal messages, if the message was a non-prioritised message, or the prioritised buffer utilisation was stamped, if the forwarded message was a prioritised message. Additionally, each node was putting its ID and its TIER zone ID into the message. Based on that information it can be shown that no message that arrived at the base station was taking an unnecessary turn or was looping between zones.

The first graph that is presented in Fig. 5 shows the amount of generated messages at each node that arrived at the base station, which does not mean that every message arrived at

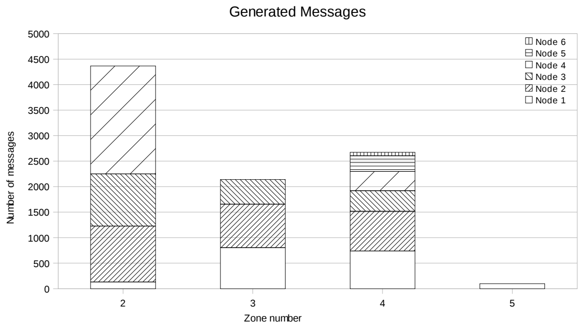


Figure 5. Generated packets at nodes over the network

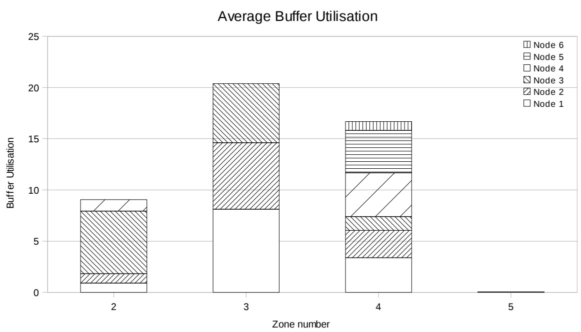


Figure 6. Buffer utilisation over the network

the base station. This graph shows that messages from TIER zones further away from the base station suffer from higher likelihood of packet loss and therefore fewer packets arrived from there. In an ideal network, including an ideal channel with zero loss, unlimited bandwidth and unlimited buffer capacity, one would expect a uniform distribution of generated messages. The message counter at each node, which includes also the dropped messages, suggest such a distribution. The reason for this is increased buffer utilisation towards the base station and hence the resulting higher contention for the channel. Fig. 6 shows the corresponding buffer utilisation.

One important issue of the experiment was to show that prioritised messages are travelling through the network faster than others. As it can be seen from Tab. I, prioritised messages travelled faster through the network than not prioritised ones.

Table I
DELAY TIMES

Delay [ms]	zone 2	zone 3	zone 4	zone 5
p messages	1753.2	3627.8	4630.9	6859.9
non-p messages	3475.1	6255.0	8365.1	5493.7

VI. CONCLUSION AND FUTURE WORK

In this paper, a cross layer solution has been proposed to fulfil the latest evolution trends of WSN in which data load increases from its conventional low volume to a moderate load. Instead of using static topology, here, the network topology is dynamic which changes frequently. This evolution has offered new challenges in WSN research and development, contradicting current WSN design spaces.

From the values in Tab. I the prioritised message handling was proven. Decreasing the initial back off time and always handling prioritised messages first, increases the likelihood to win the channel for the transmission and hence faster forwarding is achieved. Also the routing scheme of PriBaR is proven, since not a single message that arrived at the base station took an unecessary hop or looped between zones. This scheme of straight forward routing will be adjusted in the future to guarantee delivery even if there is no next hop directly available. Therefore, the experimental set up needs to be reconfigured and the implementation of further functionality is required.

REFERENCES

[1] I. F. Akyildiz, T. Melodia, and K. R. Chowdury, "Wireless multimedia sensor networks: A survey," *IEEE Wireless Communications*, 2007.
[2] M. Radenkovic and B. Wietrzyk, "Wireless mobile ad-hoc sensor network for very large scale cattle monitoring," in *Proc. IEEE Sixth International Conference on Networking (ICE'07)*, 2007.
[3] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*. The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England: John Wiley & Sons Ltd., 2005.
[4] C. Technology, *MICAz Datasheet*. [Online]. Available: http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICAz_Datasheet.pdf
[5] C. Wang, K. Sohraby, B. Li, M. Daneshmand, and Y. Hu, "A survey of transport protocols for wireless sensor networks," *Networking, IEEE*, vol. 20, no. 3, 2006.
[6] K. H. Kwong, H. G. Goh, C. Michie, and I. Andonovic, "Priority medium access control for wireless sensor network," *Communication Systems and Networks*, 2008.
[7] C. Suh and Y.-B. Ko, "A traffic aware, energy efficient mac protocol for wireless sensor networks," *Circuits and Systems, 2005. ISCAS 2005*, vol. 3, 2005.
[8] L. van Hoesel and P. Havinga, "A lightweight medium access protocol (lmac) for wireless sensor networks," *1st Int. Workshop on Networked Sensing Systems (INSS 2004)*, 2004.
[9] M. Zawodniok and S. Jagannathan, "Predictive congestion control protocol for wireless sensor networks," *IEEE Wireless Communications*, vol. 6, no. 11, 2007.
[10] C.-Y. Wan, A. T. Campbell, and L. Krishnamurthy, "Pump-slowly, fetch-quickly (psfq): A reliable transport protocol for sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, 2005.
[11] S. Kulkarni, A. Iyer, and C. Rosenberg, "An address-light, integrated mac and routing protocol for wireless sensor networks," *Networking, IEEE*, vol. 14, no. 4, 2006.
[12] S. Feit, *TCP/IP Architecture, Protocols, and Implementation with IPv6 and IP Security*. New York: McGraw-Hill, 1996.
[13] S. Hashim, H. T. Mouftah, and N. D. Georganas, "A new transport layer sensor network protocol," in *Proc. IEEE Canadian Conference on Electrical and Computer Engineering 2006. (CCECE'06)*, 2006.

Appendix C

A Complete Set of Simulation Parameters

Listing C.1: A full Set of Simulation Parameters

```
clear;

simTitle = "PTS";
model = "./PTS";
A = 95;

subscribers = 20;
pdfType = 1;
mlength = 952;
rate = 2e6;
T = 7200;
R = 10;              # Number of runs to be executed
startSeed = 137;

runs = 1:.5:10;

#Set up a vector with random number generator seeds
```

```
intervalSeeds = createSeeds(R, 2*rate*T, startSeed);
retrySeeds = createSeeds(R, T/mlength, intervalSeeds(R));
lengthSeeds= createSeeds(R,2*rate*T*pdfType,retrySeeds(R));

# Remove all old result vector files
rmOldResultVec = sprintf("cd ../../omnetpp/omnetpp3/%s/ &&
    rm -f spatialReuse.vec", simTitle);
system(rmOldResultVec);

# Disable all debugging information
commentOutEV = sprintf("cd ../../omnetpp/omnetpp3/%s/ && ./
    disableEV.sh", simTitle);
system(commentOutEV);

# Re-build simulations' make file
makeSimulation = sprintf("cd ../../omnetpp/omnetpp3/%s/ &&
    make clean && make -f Makefile.gen && make", simTitle);
system(makeSimulation);

# Make sure to wark in CMD--No graphical output
runChangeMake = sprintf("cd ../../omnetpp/omnetpp3/%s/ &&
    ./changeToCMD.sh", simTitle);
system(runChangeMake);

# Remove initial build files
rmExecutable = sprintf("rm ../../omnetpp/omnetpp3/%s/%s",
    simTitle, simTitle);
system(rmExecutable);

# Re-Make simulation--Setting all parameter
reMakeSim = sprintf("cd ../../omnetpp/omnetpp3/%s/ && make"
    , simTitle);
```

```
system(reMakeSim);

# Generating a set of *.ini files to be used with the R simulation runs

for i = 1:R

    ininame = sprintf("../../omnetpp/omnetpp3/%s/%s%i.ini", simTitle, simTitle, i);
    scalarname = sprintf("%s%i", simTitle, i);

    fidout = fopen(ininame, "w", "native");
## Beginning of the INI file
    fprintf(fidout, "[General]\n");
    fprintf(fidout, ";ini-warnings = true\n");
    fprintf(fidout, "preload-ned-files=*.ned\n");
    fprintf(fidout, "network = sim\n");
    fprintf(fidout, "#random-seed = 13\n");
    fprintf(fidout, "sim-time-limit = 3600s\n");
    fprintf(fidout, "snapshot-file = %s.sna\n", scalarname);
    fprintf(fidout, "output-vector-file = %s.vec\n", scalarname);
    fprintf(fidout, "output-scalar-file = %s.sca\n", scalarname);
    fprintf(fidout, "\n");
    fprintf(fidout, "[Tkenv]\n");
    fprintf(fidout, "bitmap-path=\"/home/chr/mobility-fw2.0p3/bitmaps\"\n");
    fprintf(fidout, "default-run=1\n");
    fprintf(fidout, "use-mainwindow = yes\n");
    fprintf(fidout, "print-banners = yes\n");
```

```
fprintf(fidout, "slowexec-delay = 300ms\n");
fprintf(fidout, "update-freq-fast = 10\n");
fprintf(fidout, "update-freq-express = 100\n");
fprintf(fidout, "breakpoints-enabled = yes\n");
fprintf(fidout, "\n");
fprintf(fidout, "[Cmdenv]\n");
fprintf(fidout, "runs-to-execute = 1-%i\n", length(runs)); # Number of runs to be executed, each with different random seed
fprintf(fidout, "event-banners = no\n");
fprintf(fidout, "module-messages = no \n");
fprintf(fidout, "; verbose-snrSimulation = no\n");
fprintf(fidout, "; verbose-snrSimulation = yes\n");
fprintf(fidout, "\n");
fprintf(fidout, "[DisplayStrings]\n");
fprintf(fidout, "\n");
fprintf(fidout, "[Parameters]\n");
fprintf(fidout, "\n");
fprintf(fidout, "######################################################\n");
fprintf(fidout, "#       Parameters for the entire simulation         #\n");
fprintf(fidout, "######################################################\n");
fprintf(fidout, "sim.playgroundSizeX = 50\n");
fprintf(fidout, "sim.playgroundSizeY = 50\n");
fprintf(fidout, "sim.numHosts = %i\n", subscribers);
fprintf(fidout, "sim.host[*].appl.numOfHosts = %i\n", subscribers);
```

```
fprintf(fidout, "; So that I remember to change this
    too!!\n");
fprintf(fidout, "\n");
fprintf(fidout, "# uncomment to enable debug messages
    for all modules\n");
fprintf(fidout, "; **.debug = 0\n");
fprintf(fidout, "**.coreDebug = 0\n");
fprintf(fidout, "\n");
fprintf(fidout, "\n");
fprintf(fidout, "
    #####################################################\
    n");
fprintf(fidout, "#       Parameters for the
    ChannelControl            #\n");
fprintf(fidout, "
    #####################################################\
    n");
fprintf(fidout, "sim.channelcontrol.carrierFrequency =
    2.4e9\n");
fprintf(fidout, "\n");
fprintf(fidout, "# max transmission power [mW]\n");
fprintf(fidout, "sim.channelcontrol.pMax  = 30\n");
fprintf(fidout, "# signal attenuation threshold [dBm]\n
    ");
fprintf(fidout, "sim.channelcontrol.sat   = -120\n");
fprintf(fidout, "# path loss coefficient alpha\n");
fprintf(fidout, "sim.channelcontrol.alpha = 4\n");
fprintf(fidout, "\n");
fprintf(fidout, "sim.channelcontrol.sendDirect = 0\n");
fprintf(fidout, "sim.channelcontrol.useTorus = 0\n");
fprintf(fidout, "\n");
fprintf(fidout, "\n");
```

```
fprintf(fidout, "######################################################\n");
fprintf(fidout, "#       Parameters for the Mobility Module             n");
fprintf(fidout, "######################################################\n");
fprintf(fidout, "# starting position for the hosts \"-1\" means random staring point\n");
fprintf(fidout, "\n");
fprintf(fidout, "sim.host[*].mobility.debug = 0\n");
fprintf(fidout, "\n");
fprintf(fidout, ";sim.host[0].mobility.x = 30\n");
fprintf(fidout, ";sim.host[0].mobility.y = 30\n");
fprintf(fidout, "\n");
fprintf(fidout, ";sim.host[1].mobility.x = 50\n");
fprintf(fidout, ";sim.host[1].mobility.y = 50\n");
fprintf(fidout, "\n");
fprintf(fidout, ";sim.host[2].mobility.x = 420\n");
fprintf(fidout, ";sim.host[2].mobility.y = 120\n");
fprintf(fidout, "\n");
fprintf(fidout, ";sim.host[3].mobility.x = 380\n");
fprintf(fidout, ";sim.host[3].mobility.y = 30\n");
fprintf(fidout, "\n");
fprintf(fidout, ";sim.host[4].mobility.x = 220\n");
fprintf(fidout, ";sim.host[4].mobility.y = 60\n");
fprintf(fidout, "\n");
fprintf(fidout, ";sim.host[5].mobility.x = 450\n");
fprintf(fidout, ";sim.host[5].mobility.y = 320\n");
fprintf(fidout, "\n");
fprintf(fidout, ";sim.host[6].mobility.x = 150\n");
```

```
fprintf(fidout, ";sim.host[6].mobility.y = 155\n");
fprintf(fidout, "\n");
fprintf(fidout, ";sim.host[7].mobility.x = 330\n");
fprintf(fidout, ";sim.host[7].mobility.y = 280\n");
fprintf(fidout, "\n");
fprintf(fidout, ";sim.host[8].mobility.x = 70\n");
fprintf(fidout, ";sim.host[8].mobility.y = 257\n");
fprintf(fidout, "\n");
fprintf(fidout, ";sim.host[9].mobility.x = 150\n");
fprintf(fidout, ";sim.host[9].mobility.y = 300\n");
fprintf(fidout, "\n");
fprintf(fidout, "sim.host[*].mobility.x=-1\n");
fprintf(fidout, "sim.host[*].mobility.y=-1\n");
fprintf(fidout, "\n");
fprintf(fidout, "\n");
fprintf(fidout, "#######################################################\n");
fprintf(fidout, "#        Parameters for the Host                      #\n");
fprintf(fidout, "#######################################################\n");
fprintf(fidout, "\n");
fprintf(fidout, "sim.host[*].applType = \"TestApplLayer\"\n");
fprintf(fidout, "\n");
fprintf(fidout, "#######################################################\n");
fprintf(fidout, "#        Parameters for the Application Layer         #\n");
```

```
fprintf(fidout, "######################################################\n");
fprintf(fidout, "\n");
fprintf(fidout, "# debug switch\n");
fprintf(fidout, "sim.host[*].appl.debug = 1\n");
fprintf(fidout, "sim.host[*].appl.headerLength=12\n");
fprintf(fidout, "sim.host[*].appl.burstSize=3\n");
fprintf(fidout, "\n");
fprintf(fidout, "######################################################\n");
fprintf(fidout, "#       Parameters for the Network Layer              #\n");
fprintf(fidout, "######################################################\n");
fprintf(fidout, "sim.host[*].net.headerLength=24\n");
fprintf(fidout, "sim.host[*].net.debug = 0\n");
fprintf(fidout, "sim.host[*].net.coreDebug = 0\n");
fprintf(fidout, "\n");
fprintf(fidout, "######################################################\n");
fprintf(fidout, "#       Parameters for ARP                            #\n");
fprintf(fidout, "######################################################\n");
fprintf(fidout, "sim.host[*].arp.debug = 0\n");
fprintf(fidout, "\n");
fprintf(fidout, "\n");
```

```
fprintf(fidout, "#####################################################\n");
fprintf(fidout, "#       Parameters for the MAC Layer                #\n");
fprintf(fidout, "#####################################################\n");
fprintf(fidout, "\n");
fprintf(fidout, "sim.host[*].nic.mac.debug = 1\n");
fprintf(fidout, "sim.host[*].nic.mac.queueLength=5\n");
fprintf(fidout, "sim.host[*].nic.mac.headerLength=24\n");
fprintf(fidout, "sim.host[*].nic.mac.busyRSSI=-97\n");
fprintf(fidout, "sim.host[*].nic.mac.slotDuration=0.01\n");
fprintf(fidout, "sim.host[*].nic.mac.difs=0.006\n");
fprintf(fidout, "sim.host[*].nic.mac.maxTxAttempts=14\n");
fprintf(fidout, "sim.host[*].nic.mac.defaultChannel = 0\n");
fprintf(fidout, "sim.host[*].nic.mac.bitrate = 15360\n");
fprintf(fidout, "sim.host[*].nic.mac.contentionWindow = 31\n");
fprintf(fidout, "sim.host[*].nic.mac.rtsCtsThreshold = 3\n");
fprintf(fidout, "sim.host[*].nic.mac.autoBitrate = 0\n");
fprintf(fidout, ";SNR values for the different supported Bitrates\n");
fprintf(fidout, "sim.host[*].nic.mac.snr2Mbit = 10\n");
```

```
fprintf(fidout, "sim.host[*].nic.mac.snr5Mbit = 10\n");
fprintf(fidout, "sim.host[*].nic.mac.snr11Mbit = 10\n")
    ;
fprintf(fidout, "\n");
fprintf(fidout, "sim.host[*].nic.mac.
    neighborhoodCacheSize = 10\n");
fprintf(fidout, "sim.host[*].nic.mac.
    neighborhoodCacheMaxAge = 5\n");
fprintf(fidout, "; Minimum required reliability\n");
fprintf(fidout, "sim.host[*].nic.mac.Pcfmin = .9\n");
fprintf(fidout, "\n");
fprintf(fidout, "
    ######################################################\
    n");
fprintf(fidout, "#       Parameters for the decider
                            #\n");
fprintf(fidout, "
    ######################################################\
    n");
fprintf(fidout, "sim.host[*].nic.decider.snirThreshold
    = -9\n");
fprintf(fidout, "\n");
fprintf(fidout, "
    ######################################################\
    n");
fprintf(fidout, "#       Parameters for the radio
                              #\n");
fprintf(fidout, "
    ######################################################\
    n");
fprintf(fidout, "\n");
fprintf(fidout, "sim.host[*].nic.radio.swSleep = 0\n");
```

```
fprintf(fidout, "sim.host[*].nic.radio.swSend = 192e-6\n");
fprintf(fidout, "sim.host[*].nic.radio.swRecv = 192e-6\n");
fprintf(fidout, "sim.host[*].nic.radio.debug = 0\n");
fprintf(fidout, "\n");
fprintf(fidout, "\n");
fprintf(fidout, "######################################################\n");
fprintf(fidout, "#       Parameters for the Physical Layer            #\n");
fprintf(fidout, "######################################################\n");
fprintf(fidout, "\n");
fprintf(fidout, "# debug switch\n");
fprintf(fidout, "sim.host[*].nic.snrEval.debug = 1\n");
fprintf(fidout, "sim.host[*].nic.snrEval.publishRSSIAlways = 0\n");
fprintf(fidout, "sim.host[*].nic.snrEval.headerLength=16\n");
fprintf(fidout, "# transmission power [mW]\n");
fprintf(fidout, "sim.host[*].nic.snrEval.transmitterPower=1\n");
fprintf(fidout, "\n");
fprintf(fidout, "sim.host[*].nic.snrEval.carrierFrequency=2.4e9\n");
fprintf(fidout, "sim.host[*].nic.snrEval.thermalNoise=-110\n");
fprintf(fidout, "sim.host[*].nic.snrEval.sensitivity=-96\n");
```

```
    fprintf(fidout, "sim.host[*].nic.snrEval.pathLossAlpha=4\n");
    fprintf(fidout, "\n");
    fprintf(fidout, "sim.host[*].nic.decider.debug = 1\n");
    fprintf(fidout, "sim.host[*].nic.decider.snrThresholdLevel=10;[dB]\n");
    fprintf(fidout, "\n");
# End General set up

# Run specific parameters
    for j = 1:length(runs)
        fprintf(fidout, "[Run %d]\n", j);        # Indicator
        fprintf(fidout, "description = \"No movement\"\n");
        fprintf(fidout, "seed-0-lcg32 = %i\n", intervalSeeds(i));    # Random seed
        fprintf(fidout, "sim.host[*].mobility.updateInterval = .1\n");
        fprintf(fidout, "sim.host[*].appl.msgFreq = uniform(8, 12)\n");
        fprintf(fidout, "; Frequency with which the messages gets send, and this really means frequency in Hz\n");
        fprintf(fidout, "sim.host[*].appl.streamFreq = %f\n", runs(j)/10);
        fprintf(fidout, "\n");
    endfor
    fclose(fidout);

    command = sprintf("rm -f %s.sca", scalarname);   # Removing all old scalar files

    system(command);
```

```
    sim = sprintf("../../omnetpp/omnetpp3/%s/%s", simTitle,
         model);
    ini = sprintf("../../omnetpp/omnetpp3/%s/%s%i.ini",
        simTitle, simTitle, i);

    runAloha(sim, ini, length(runs))
endfor

## Notify the user about the finished simulation
system('../sendMail.pl')
getResults(intervalSeeds, simTitle)
```

Bibliography

[1] *Scalable Infrastructure for Distributed Sensor Networks.* k. Chakrabarty and S. S. Iyengar, 2005.

[2] *Sensor Network—Where Theory Meets Practice.* Dr. Gianluigi Ferrari, 2009.

[3] S. Ganesh and R. Amutha, "Energy Efficient Transport Protocol for Wireless Sensor Networks," in *Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on*, August 2009, pp. 464–468.

[4] J. Gomez and A. T. Campbell, "Variable-Range Transmission Power Control in Wireless Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 1, pp. 87–99, 2007.

[5] K. H. Kwong, K. Sasloglou, H. Goh, T. T. Wu, B. Stephen, M. Gilroy, C. Tachtatzis, I. Glover, C. Michie, and I. Andonovic, "Adaptation of wireless sensor network for farming industries," in *Networked Sensing Systems (INSS), 2009 Sixth International Conference on*, 2009, pp. 1–4.

[6] K. H. kwong, H. G. Goh, T. T. Wu, B. Stephen, C. Michie, I. Andonovic, D. Ross, J. Hyslop, and C. X. Liu, "Wireless telemetry for livestock condition monitoring," in *7th World Congress on Computers in Agriculture Conference Proceedings, 22-24 June 2009, Reno, Nevada*, June 2009.

[7] ISO/IEC, *Information technology — Open Systems Interconnection — Basic Reference Model: The Basic Model*. International Organization for Standardization, 1994.

[8] H. Zimmermann, "OSI Reference Model—The OSI Model of Architecture for Open System Interconnection," *IEEE TRANSACTION ON COMMUNICATIONS*, vol. 28, no. 4, pp. 425–432, 1980.

[9] N. Abramson, "The ALOHA System – Another Alternative for Computer Communication," ser. Fall Joint Computer Conference, April 1970, pp. 281–285.

[10] A. S. Tannebaum, *Computer Networks*. Upper Saddle River, New Jersey 07458: Prentice-Hall Inc., 1996.

[11] M. Ilyas and I. Mahgoub, *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*. 2000 N.W. Corporate Blvd., Boca Raton, Florida 33431: CRC Press LLC, 2005.

[12] I. I. O. for Standardization), "iso.org," 2014, accessed: Aug 04http://www.iso.org/iso/home.htm.

[13] B. Hartpence, *Packet Guide to Core Network Protocols*. O'Reilly Media, Inc., 2011.

[14] G. Hura and M. Singhal, *Data and Computer Communications: Networking and Internetworking*. Taylor & Francis, 2001.

[15] J. Com and J. Dong, *Network Dictionary*, ser. ITPro collection. Javvin Technologies, Incorporated, 2007.

[16] R. Jurdak, *Wireless Ad Hoc and Sensor Networks: A Cross-Layer Design Perspective*, ser. Signals and Communication Technology. Springer Science + Business Media, 2007.

[17] N. Abramson, "The Throughput of Packet Broadcasting Channels," *Communications, IEEE Transactions on*, vol. 25, no. 1, pp. 117–128, 1977.

[18] S. Nikoletseas and J. D. Rolim, *Wireless Sensor Networks.* Springer, 2006.

[19] A.-B. García-Hernando, J.-F. Martínez-Ortega, J.-M. López-Navarro, A. Prayati, and L. Redondo-López, *Problem Solving for Wireless Sensor Networks.* Springer, 2008.

[20] S. Lin, J. Zhang, G. Zhou, L. Gu, J. A. Stankovic, and T. He, "Atpc: Adaptive transmission power control for wireless sensor networks," in *Proceedings of the 4^{th} International Conference on Embedded Networked Sensor Systems*, ser. SenSys '06. New York, NY, USA: ACM, 2006, pp. 223–236.

[21] A. Salem and A. Kumar, "Power optimization in mobile networks using genetic algorithm to guarantee qos," in *Intelligent Information Technology*, ser. Lecture Notes in Computer Science, G. Das and V. Gulati, Eds. Springer Berlin / Heidelberg, 2005, vol. 3356, pp. 147–159, 10.1007/978-3-540-30561-3_16.

[22] T. S. Rappaport *et al.*, *Wireless communications: principles and practice.* Prentice Hall PTR New Jersey, 1996, vol. 2.

[23] I. Demirkol, C. Ersoy, and F. Alagöz, "Mac protocols for wireless sensor networks: A survey," *IEEE Communications Magazine*, pp. 115–121, April 2006.

[24] J. Dunlop and S. D. G., *Telecommunications Engineering*, 3rd ed. Cheltenham, United Kingdom: Stanley Thornes Ltd, 1998.

[25] T. Singal, *Wireless Communications.* McGraw-Hill Education (India) Pvt Limited, 2010.

[26] L. Kleinrock and F. Tobagi, "Packet switching in radio channels: Part i–carrier sense multiple-access modes and their throughput-delay characteristics," *Communications, IEEE Transactions on*, vol. 23, no. 12, pp. 1400–1416, Dec 1975.

[27] C. Kozierok, *The TCP/IP Guide: A Comprehensive, Illustrated Internet Protocols Reference*, ser. No Starch Press Series. No Starch Press, 2005.

[28] C. Raghavendra, K. Sivalingam, and T. Znati, *Wireless Sensor Networks*, ser. Ercoftac Series. Springer, 2004.

[29] TI, *2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver*, Texas Instruments Incorporated, 2014.

[30] H. J. Ahn, A. Aktasa, B. Drannon, B. Hodges, T. Hornak, M. Ismail, M. LeFerve, E. McCune, J. Y. McNair, M. R. Moore, J. T. Moring, A. Nasipuri, G. Niu, P. Okrah, L. Pelletier, R. Poor, D. Runton, H. Sadjadpour, and M. C. Valenti, *Handbook of RF and Wireless Technologies*, F. Dowla, Ed. Elsevier, 2004.

[31] TI, *A True System-on-Chip solution for 2.4 GHz IEEE 802.15.4 / ZigBee®*, Texas Instruments Incorporated, 2013.

[32] OKI, *OKI Semiconductor, ML7050LA*, Oki Semiconductor, 2001.

[33] J. Zhang, J. Chen, J. Fan, W. Xu, and Y. Sun, "Omnet++ based simulation for topology control in wireless sensor network: A case study," in *Wireless Communications and Mobile Computing Conference, 2008. IWCMC '08. International*, 2008, pp. 1130–1134.

[34] Z. Zhao, B. Mosler, and T. Braun, "Performance evaluation of opportunistic routing protocols: A framework-based approach using omnet++," in *Proceedings of the 7^{th} Latin American Networking Conference*, ser. LANC '12. New York, NY, USA: ACM, 2012, pp. 28–35.

[35] "Ieee standard for information technology- telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements-part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications," *IEEE Std 802.11-1997*, pp. i–445, 1997.

[36] G. Bianchi, "Performance analysis of the ieee 802.11 distributed coordination function," *Selected Areas in Communications, IEEE Journal on*, vol. 18, no. 3, pp. 535–547, March 2000.

[37] ——, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE Journal on Selected Areas in Communication*, 2000.

[38] B. Fette, *RF & Wireless Technologies*, ser. Newnes know it all series. Newnes/Elsevier, 2008.

[39] A. Cerpa, J. Wong, L. Kuang, M. Potkonjak, and D. Estrin, "Statistical model of lossy links in wireless sensor networks," in *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, 2005, pp. 81–88.

[40] J. Al-Karaki and A. Kamal, "Routing techniques in wireless sensor networks: a survey," *Wireless Communications, IEEE*, vol. 11, no. 6, pp. 6–28, 2004.

[41] S. Adibi and S. Erfani, "A Multipath Routing Survey for Mobile Ad-Hoc Networks," *IEEE CCNC proceedings*, 2006.

[42] Rajashree.V.Biradar, V. .Patil, D. S. R. Sawant, and D. R. R. Mudholkar, "Classification and comparison of routing protocols in wireless sensor networks," *Special Issue on Ubiquitous Computing Security Systems*, vol. 4, pp. 704–711, 2009.

[43] K. Iwanicki and M. van Steen, "On hierarchical routing in wireless sensor networks," in *Proceedings of the 2009 International Conference on Information Processing in Sensor Networks*, ser. IPSN '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 133–144.

[44] L. Shu, Y. Zhang, L. Yang, Y. Wang, and M. Hauswirth, "Geographic routing in wireless multimedia sensor networks," in *Future Generation Communication and Networking, 2008. FGCN '08. Second International Conference on*, vol. 1, 2008, pp. 68–73.

[45] O. Chipara, Z. He, G. Xing, Q. Chen, X. Wang, C. Lu, J. Stankovic, and T. Abdelzaher, "Real-time power-aware routing in sensor networks," in *Quality of Service, 2006. IWQoS 2006.* 14^{th} *IEEE International Workshop on*, 2006, pp. 83–92.

[46] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," *Ad Hoc Networks*, vol. 1, no. 2–3, pp. 293–315, 2003, sensor Network Protocols and Applications.

[47] T. Zhu, Z. Zhong, T. He, and Z.-L. Zhang, "Exploring link correlation for efficient flooding in wireless sensor networks." in *NSDI*, 2010, pp. 49–64.

[48] S. Wu and K. Candan, "GMP: Distributed geographic multicast routing in wireless sensor networks," in *Distributed Computing Systems, 2006. ICDCS 2006.* 26^{th} *IEEE International Conference on*, 2006, pp. 49–57.

[49] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient mac protocol for wireless sensor networks," in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, 2002, pp. 1567 – 1576 vol.3.

[50] C. Intanagonwiwat, R. Govindan, and D. Estrin, Eds., *Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor*

Networks. In Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MobiCOM '00), August 2000, Boston, Massachussetts, March 2000.

[51] P. Levis, N. Patel, D. Culler, and S. Shenker, "Trickle: A Self-Regulating Algorithm for Code Propagation and Maintenance in Wireless Sensor Networks," in *NSDI'04: Proceedings of the 1st conference on Symposium on Networked Systems Design and Implementation*. Berkeley, CA, USA: USENIX Association, 2004, pp. 2–2.

[52] "Optimized Link State Routing Protocol (OLSR)," United States, 2003.

[53] J. N. Al-karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: A survey," *IEEE Wireless Communications*, vol. 11, pp. 6–28, 2004.

[54] E. Felemban, C.-G. Lee, and E. Ekici, "MMSPEED: multipath multi-speed protocol for qos guarantee of reliability and. timeliness in wireless sensor networks," *Mobile Computing, IEEE Transactions on*, vol. 5, no. 6, pp. 738–754, 2006.

[55] M. Pearlman and Z. Haas, "Improving the performance of query-based routing protocols through ldquo;diversity injection rdquo;," in *Wireless Communications and Networking Conference, 1999. WCNC. 1999 IEEE*, 1999, pp. 1548–1552 vol.3.

[56] J. Kulik, W. Heinzelmann, and H. Balakrishnan, "Negotiation-based protocols for disseminating information in wireless sensor networks," *Wireless Networks*, pp. 169–185, March 2002.

[57] J. Ben-Othman and B. Yahya, "Energy efficient and qos based routing protocol for wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 70, no. 8, pp. 849–857, 2010.

[58] G. Halkes and K. Langendoen, "Crankshaft: An energy-efficient mac-protocol for dense wireless sensor networks," in *Wireless Sensor Networks*, ser. Lecture Notes in Computer Science, K. Langendoen and T. Voigt, Eds. Springer Berlin Heidelberg, 2007, vol. 4373, pp. 228–244.

[59] B. Krishnamachari, D. Estrin, and S. Wicker, "The impact of data aggregation in wireless sensor networks," in *Distributed Computing Systems Workshops, 2002. Proceedings. 22nd International Conference on*, 2002, pp. 575–578.

[60] J. Deng, R. Han, and S. Mishra, "Security support for in-network processing in wireless sensor networks," in *Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks*, ser. SASN '03. New York, NY, USA: ACM, 2003, pp. 83–93.

[61] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communications*, vol. 30, no. 14–15, pp. 2826–2841, 2007, network Coverage and Routing Schemes for Wireless Sensor Networks.

[62] E. F. Nakamura, H. S. Ramos, L. A. Villas, H. A. de Oliveira, A. L. de Aquino, and A. A. Loureiro, "A reactive role assignment for data routing in event-based wireless sensor networks," *Computer Networks*, vol. 53, no. 12, pp. 1980–1996, 2009.

[63] S. C. Bhupesh Kumar Gupta, "A survey on node placement for wireless sensor network using pso algorithm," India, pp. 28–33.

[64] N. Vlajic and D. Xia, "Wireless sensor networks: To cluster or not to cluster?" in *Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks*, ser. WOWMOM '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 258–268.

[65] M. Meghji and D. Habibi, “Transmission power control in single-hop and multi-hop wireless sensor networks,” in *Multiple Access Communications*, ser. Lecture Notes in Computer Science, C. Sacchi, B. Bellalta, A. Vinel, C. Schlegel, F. Granelli, and Y. Zhang, Eds. Springer Berlin Heidelberg, 2011, vol. 6886, pp. 130–143.

[66] M. Hanson, H. Powell, A. Barth, K. Ringgenberg, B. Calhoun, J. Aylor, and J. Lach, “Body area sensor networks: Challenges and opportunities,” *Computer*, vol. 42, no. 1, pp. 58–65, 2009.

[67] R. Rajagopalan and P. Varshney, “Data-aggregation techniques in sensor networks: A survey,” *Communications Surveys Tutorials, IEEE*, vol. 8, no. 4, pp. 48–63, 2006.

[68] Stanford. (2013) CC2420 layer descriptions.

[69] P. Jiang, “A new method for node fault detection in wireless sensor networks,” *Sensors*, vol. 9, no. 2, pp. 1282–1294, 2009.

[70] H. Alwan and A. Agarwal, “A survey on fault tolerant routing techniques in wireless sensor networks,” in *Sensor Technologies and Applications, 2009. SENSORCOMM’09. Third International Conference on.* IEEE, 2009, pp. 366–371.

[71] H. Dai and R. Han, “A node-centric load balancing algorithm for wireless sensor networks,” in *Global Telecommunications Conference, 2003. GLOBECOM ’03. IEEE*, vol. 1, December 2003, pp. 548–552 Vol.1.

[72] G. Cugola and M. Migliavacca, “A context and content-based routing protocol for mobile sensor networks,” in *Wireless Sensor Networks*, ser. Lecture Notes in Computer Science, U. Roedig and C. Sreenan, Eds. Springer Berlin Heidelberg, 2009, vol. 5432, pp. 69–85.

[73] R. Rajagopalan and P. Varshney, "Connectivity analysis of wireless sensor networks with regular topologies in the presence of channel fading," *Wireless Communications, IEEE Transactions on*, vol. 8, no. 7, pp. 3475–3483, July 2009.

[74] G. Anastasi, M. Conti, M. D. Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Ad Hoc Networks*, vol. 7, no. 3, pp. 537–568, 2009.

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

[76] J. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin, and D. Ganesan, "Building efficient wireless sensor networks with low-level naming," *SIGOPS Oper. Syst. Rev.*, vol. 35, no. 5, pp. 146–159, Oct 2001.

[77] A. Mahapatra, K. Anand, and D. P. Agrawal, "Qos and energy aware routing for real-time traffic in wireless sensor networks," *Computer Communications*, vol. 29, no. 4, pp. 437–445, 2006, current areas of interest in wireless sensor networks designs.

[78] K. Akkaya and M. Younis, "An energy-aware qos routing protocol for wireless sensor networks," in *Distributed Computing Systems Workshops, 2003. Proceedings. 23rd International Conference on*, May 2003, pp. 710–715.

[79] F. Ishmanov, A. S. Malik, and S. W. Kim, "Energy consumption balancing (ecb) issues and mechanisms in wireless sensor networks (wsns): a comprehensive overview," *European Transactions on Telecommunications*, vol. 22, no. 4, pp. 151–167, 2011.

[80] S. K. Singh, M. P. Singh, and D. Singh, “Routing protocols in wireless sensor networks–a survey,” *International Journal of Computer Science & Engineering Survey (IJCSES)*, vol. 1, no. 2, November 2010.

[81] M. Kochhal, L. Schwiebert, and S. Gupta, “Role-based hierarchical self organization for wireless ad hoc sensor networks,” in *Proceedings of the 2Nd ACM International Conference on Wireless Sensor Networks and Applications*, ser. WSNA ’03. New York, NY, USA: ACM, 2003, pp. 98–107.

[82] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, “Highly-resilient, energy-efficient multipath routing in wireless sensor networks,” *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 5, no. 4, pp. 11–25, October 2001.

[83] A. Manjeshwar, Q.-A. Zeng, and D. Agrawal, “An analytical model for information retrieval in wireless sensor networks using enhanced apteen protocol,” *Parallel and Distributed Systems, IEEE Transactions on*, vol. 13, no. 12, pp. 1290–1302, Dec 2002.

[84] I. F. Akyildiz, D. Pompili, and T. Melodia, “Challenges for efficient communication in underwater acoustic sensor networks,” *SIGBED Rev.*, vol. 1, no. 2, pp. 3–8, July 2004.

[85] D. Macedo, L. Correia, A. Dos Santos, A. A. F. Loureiro, and J. M. S. Nogueira, “A pro-active routing protocol for continuous data dissemination in wireless sensor networks,” in *Computers and Communications, 2005. ISCC 2005. Proceedings. 10th IEEE Symposium on*, June 2005, pp. 361–366.

[86] Y.-H. Wang, C.-P. Hsu, Y.-C. Lin, C.-S. Kuo, and H.-Y. Ho, “A routing method by reactive energy decision in wireless sensor networks,” in *Advanced Information Networking and Applications Workshops, 2007,*

AINAW '07. 21st International Conference on, vol. 2, May 2007, pp. 701–706.

[87] A. Manjeshwar and D. P. Agrawal, "Apteen: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks." in *ipdps*, vol. 2, 2002, p. 48.

[88] L. Zhuang, J.-B. Zhang, Y. Zhao, M. Luo, D. Zhang, and Z. Yang, "Power-aware service-oriented architecture for wireless sensor networks," in *Industrial Electronics Society, 2005. IECON 2005. 31st Annual Conference of IEEE*, Nov 2005, pp. 6 pp.–.

[89] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, 2005.

[90] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," in *Proceedings of the 5^{th} annual ACM/IEEE international conference on Mobile computing and networking*, ser. MobiCom '99. New York, NY, USA: ACM, 1999, pp. 174–185.

[91] J. Kulik, W. Heinzelman, and H. Balakrishnan, "Negotiation-based protocols for disseminating information in wireless sensor networks," *Wirel. Netw.*, vol. 8, no. 2/3, pp. 169–185, March 2002.

[92] B. Krishnamachari, D. Estrin, and S. Wicker, "Modelling data-centric routing in wireless sensor networks," in *IEEE infocom*, vol. 2, 2002, pp. 39–44.

[93] D. Braginsky and D. Estrin, "Rumor routing algorithm for sensor networks," 2002, pp. 22–31.

[94] N. F. Ye and F. Y. A. Chen, "A scalable solution to minimum cost forwarding in large sensor," 2001, pp. 304–309.

[95] C. Schurgers and M. Srivastava, "Energy efficient routing in wireless sensor networks," in *Military Communications Conference, 2001. MILCOM 2001. Communications for Network-Centric Operations: Creating the Information Force. IEEE*, vol. 1, 2001, pp. 357–361 vol.1.

[96] M. Chu, H. Haussecker, F. Zhao, M. Chu, H. Haussecker, and F. Zhao, "Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks," *International Journal of High Performance Computing Applications*, vol. 16, 2002.

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

[98] N. Sadagopan, B. Krishnamachari, and A. Helmy, "The acquire mechanism for efficient querying in sensor networks," in *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on*, May 2003, pp. 149–155.

[99] D. P. T. Vanathi and R. Vidhyapriya, "Energy aware routing for wireless sensor networks," 2007.

[100] S. D. Servetto and G. Barrenechea, "Constrained random walks on random graphs: routing algorithms for large scale wireless sensor networks," in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, ser. WSNA '02. New York, NY, USA: ACM, 2002, pp. 12–21.

[101] L. R. Ford and D. R. Fulkerson, *Flows in Networks.* Princeton University Press, 1962.

[102] A. Hassanien, J. Abawajy, A. Abraham, and H. Hagras, *Pervasive Computing: Innovations in Intelligent Multimedia and Applications*, ser. Computer Communications and Networks. Springer London, 2009.

[103] J. Ibriq and I. Mahgoub, "Cluster-based routing in wireless sensor networks: issues and challenges," in *SPECTS*, vol. 4, 2004, pp. 759–766.

[104] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *Wireless Communications, IEEE Transactions on*, vol. 1, no. 4, pp. 660–670, Oct 2002.

[105] H. Junping, J. Yuhui, and D. Liang, "A time-based cluster-head selection algorithm for leach," in *Computers and Communications, 2008. ISCC 2008. IEEE Symposium on*, July 2008, pp. 1172–1176.

[106] L. D. Mendes and J. J. Rodrigues, "A survey on cross-layer solutions for wireless sensor networks," *Journal of Network and Computer Applications*, vol. 34, no. 2, pp. 523–534, 2011, efficient and Robust Security and Services of Wireless Mesh Networks.

[107] I. Gupta, D. Riordan, and S. Sampalli, "Cluster-head election using fuzzy logic for wireless sensor networks," in *Communication Networks and Services Research Conference, 2005. Proceedings of the 3rd Annual*, May 2005, pp. 255–260.

[108] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *System Sciences, 2000. Proceedings of the* 33^{rd} *Annual Hawaii International Conference on*, vol. 2, January 2000, p. 10 pp.

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

[110] ——, "Pegasis: Power-efficient gathering in sensor information systems," in *Aerospace Conference Proceedings, 2002. IEEE*, vol. 3, 2002, pp. 3–1125–3–1130 vol.3.

[111] A. Manjeshwar and D. P. Agrawal, "Teen: A routing protocol for enhanced efficiency in wireless sensor networks."

[112] ——, "Apteen: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks." in *ipdps*, vol. 2, 2002, p. 48.

[113] V. Rodoplu and T. Meng, "Minimum energy mobile wireless networks," *Selected Areas in Communications, IEEE Journal on*, vol. 17, no. 8, pp. 1333–1344, August 1999.

[114] L. Li and J. Halpern, "Minimum-energy mobile wireless networks revisited," in *Communications, 2001. ICC 2001. IEEE International Conference on*, vol. 1, June 2001, pp. 278–283.

[115] H. Picciotto, *Algebra: Themes, Tools, Concepts*. Creative Publications, 1994.

[116] L. Subramanian and R. Katz, "An architecture for building self-configurable systems," in *Mobile and Ad Hoc Networking and Computing, 2000. MobiHOC. 2000 First Annual Workshop on*, 2000, pp. 63–73.

[117] Q. Fang, F. Zhao, and L. Guibas, "Lightweight sensing and communication protocols for target enumeration and aggregation," in *Proceedings of the 4^{th} ACM international symposium on Mobile ad hoc networking & computing*, ser. MobiHoc '03. New York, NY, USA: ACM, 2003, pp. 165–176.

[118] ——, "Lightweight sensing and communication protocols for target enumeration and aggregation," in *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking &Amp; Computing*, ser. MobiHoc '03. New York, NY, USA: ACM, 2003, pp. 165–176.

[119] J. N. Al-Karaki, R. Ul-Mustafa, and A. E. Kamal, "Data aggregation and routing in wireless sensor networks: Optimal and heuristic algorithms," *Computer Networks*, vol. 53, no. 7, pp. 945–960, 2009.

[120] Q. Li, J. Aslam, and D. Rus, "Hierarchical power-aware routing in sensor networks," in *In Proceedings of the DIMACS Workshop on Pervasive Networking*, 2001.

[121] Q. Dong, S. Banerjee, M. Adler, and A. Misra, "Minimum energy reliable paths using unreliable wireless links," in *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '05. New York, NY, USA: ACM, 2005, pp. 449–459.

[122] H. Luo, F. Ye, J. Cheng, S. Lu, and L. Zhang, "Ttdd: two-tier data dissemination in large-scale wireless sensor networks," *Wireless Networks*, vol. 11, no. 1-2, pp. 161–175, 2005.

[123] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang, "A two-tier data dissemination model for large-scale wireless sensor networks," in *Proceedings of the 8th annual international conference on Mobile computing and networking*. ACM, 2002, pp. 148–159.

[124] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *Proceedings of the 7^{th} annual international conference on Mobile computing and networking*, ser. MobiCom '01. New York, NY, USA: ACM, 2001, pp. 70–84.

[125] ——, "Geography-informed energy conservation for ad hoc routing," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '01. New York, NY, USA: ACM, 2001, pp. 70–84.

[126] Y. Yu, R. Govindan, and D. Estrin, "Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks," Technical report ucla/csd-tr-01-0023, UCLA Computer Science Department, Tech. Rep., 2001.

[127] H. Takagi and L. Kleinrock, "Optimal transmission ranges for randomly distributed packet radio terminals," *Communications, IEEE Transactions on*, vol. 32, no. 3, pp. 246–257, 1984.

[128] E. Kranakis, H. Singh, and J. Urrutia, "Compass routing on geometric networks," in *in Proc. 11 th Canadian Conference on Computational Geometry*, 1999.

[129] I. Stojmenovic and X. Lin, "Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 12, no. 10, pp. 1023–1032, Oct 2001.

[130] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Worst-case optimal and average-case efficient geometric ad-hoc routing," in *Proceedings of the 4^{th} ACM international symposium on Mobile ad hoc networking & computing*, ser. MobiHoc '03. New York, NY, USA: ACM, 2003, pp. 267–278.

[131] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," *Wireless Networks*, vol. 8, no. 5, pp. 481–494, 2002.

[132] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*. The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England: John Wiley & Sons, Ltd, 2006.

[133] D. S. Feit, *TCP/IP—Architecture, Protocols, and Implementation with IPv6 and IP Security*, S. Elliot, Ed. 11 West 19^{th} Street, New York, NY10011: McGraw Hill, 1996.

[134] S. Misra, S. C. Misra, and I. Woungang, *Guide to wireless sensor networks*. Springer, 2009, vol. 7.

[135] K. R. Fall and W. R. Stevens, *Tcp/ip illustrated*. Addison-Wesley Professional, 2011, vol. 1.

[136] S. Hashmi, H. T. Mouftah, and N. D. Georganas, "A new Transport Layer Sensor Network Protocol," *IEEE CCECE/CCGEI*, pp. 1116–1119, 2006.

[137] F. Bonomi and K. Fendick, "The rate-based flow control framework for the available bit rate atm service," *Network, IEEE*, vol. 9, no. 2, pp. 25–39, Mar 1995.

[138] P. Nar and E. Cayirci, "Pcsmac: a power controlled sensor-mac protocol for wireless sensor networks," in *Wireless Sensor Networks, 2005. Proceeedings of the Second European Workshop on*, Jan 2005, pp. 81–92.

[139] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *Communications Magazine, IEEE*, vol. 40, no. 8, pp. 102–114, August 2002.

[140] I. Akyildiz, M. C. Vuran, and O. Akan, "A cross-layer protocol for wireless sensor networks," in *Information Sciences and Systems, 2006 40th Annual Conference on*, March 2006, pp. 1102–1107.

[141] L. Van Hoesel, T. Nieberg, J. Wu, and P. J. M. Havinga, "Prolonging the lifetime of wireless sensor networks by cross-layer interaction," *Wireless Communications, IEEE*, vol. 11, no. 6, pp. 78–86, Dec 2004.

[142] T. Melodia, M. Vuran, and D. Pompili, "The state of the art in cross-layer design for wireless sensor networks," in *Wireless Systems and Network Architectures in Next Generation Internet*, ser. Lecture Notes in Computer Science, M. Cesana and L. Fratta, Eds. Springer Berlin Heidelberg, 2006, vol. 3883, pp. 78–92.

[143] W. Su and T. Lim, "Cross-layer design and optimization forwireless sensor networks," in *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2006. SNPD 2006. Seventh ACIS International Conference on*, June 2006, pp. 278–284.

[144] J. Vacca, *Computer and Information Security Handbook.* Elsevier Science, 2012.

[145] M. Kuorilehto, J. Suhonen, M. Kohvakka, M. Hännikäinen, and T. D. Hämäläinen, "Experimenting TCP/IP for low-power Wireless Sensor Networks," *17th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'06)*, 2006.

[146] Y. Tian, K. Xu, and N. Ansari, "TCP in wireless environments: Problems and Solutions," *IEEE Communications Magazine*, vol. 43, no. 3, pp. 27–32, 2005.

[147] K. Fall and S. Floyd, "Simulation-based comparisons of tahoe, reno and sack tcp," *Computer Communication Review*, vol. 26, no. 3, pp. 5–21, 1996.

[148] CrossBow, "MicaZ–Wireless Measurement System," Crossbow Technology, Inc., Datasheet. [Online]. Available: www.openautomation.net/uploadsproductos/micaz_datasheet.pdf

[149] S. Kulkarni, A. Iyer, and C. Rosenberg, "An address-light, integrated mac and routing protocol for wireless sensor networks," *IEEE/ACM*

Trans. Netw., vol. 14, pp. 793–806, August 2006. [Online]. Available: http://dx.doi.org/10.1109/TNET.2006.880163

[150] S. Bhowmick, J. Küng, and R. Wagner, *Database and Expert Systems Applications: 20th International Conference, DEXA 2009, Linz, Austria, August 31 - September 4, 2009, Proceedings*, ser. LNCS sublibrary: Information systems and applications, incl. Internet/Web, and HCI. Springer, 2009.

[151] K. Nguyen, T. Nguyen, C. K. Chaing, and M. Motani, "A prioritized mac protocol for multihop, event-driven wireless sensor networks," in *Communications and Electronics, 2006. ICCE '06. First International Conference on*, 2006, pp. 47–52.

[152] Y. Liu, H. Huang, and K. Xu, "Multhi-path-based Distributed TCP Caching for Wireless Sensor Networks," 2007.

[153] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, 2005.

[154]

[155] D. Bear, *Principles of Telecommunication-Traffic Engineering*, P. J. Flood, Ed. The Institut of Electrical Engineers, London and New York, 1976.

[156] *Configuring Cisco Voice Over IP 2E.* Elsevier Science, 2002.

[157] R. Nelson, *Probability, Stochastic Processes, and Queueing Theory: The Mathematics of Computer Performance Modeling.* Springer, 1995.

[158] A. O. Allen, *Probability, Statistics, and Queueing Theory with Computer Science Applications*, D. S. Werner Rheinboldt, Ed. Academic Press Limited, 1990.

[159] W. Leo, *Techniques for Nuclear and Particle Physics Experiments: A How-To Approach*. U.S. Government Printing Office, 1994.

[160] C. Reviews, *e-Study Guide for: Probability and Statistics for Engineers and Scientists by Anthony J. Hayter, ISBN 9780495107576*. Cram101, 2012.

[161] M. Gupta and R. Khanna, *Quantitative Techniques for Decision Making*. PHI Learning Pvt. Ltd.

[162] S. Dyer, *Wiley Survey of Instrumentation and Measurement*. Wiley, 2004.

[163] S. Efromovich, *Nonparametric Curve Estimation: Methods, Theory and Applications*, ser. Springer Series in Statistics. Wiley, 1999.

[164] W. Nelson, *Recurrent Events Data Analysis for Product Repairs, Disease Recurrences, and Other Applications*, ser. ASA-SIAM Series on Statistics and Applied Probability. Society for Industrial and Applied Mathematics, 2003.

[165] P. Levis, "Tinyos programming," http://www.tinyos.net/tinyos-2.x/doc/pdf/tinyos-programming.pdf, Delta-T Devices, October 2006.

[166] N. Dopico, C. Gil-Soriano, I. Arrazola, and S. Zazo, "Analysis of ieee 802.15.4 throughput in beaconless mode on micaz under tinyos 2," in *Vehicular Technology Conference Fall (VTC 2010-Fall), 2010 IEEE 72nd*, September 2010, pp. 1–5.

[167] A. Sinha and A. Chandrakasan, "Dynamic power management in wireless sensor networks," *Design Test of Computers, IEEE*, vol. 18, no. 2, pp. 62–74, Mar 2001.

[168] S. Tennina, M. Di Renzo, F. Graziosi, and F. Santucci, "Locating zigbee® nodes using the ti®s cc2431 location engine: A

testbed platform and new solutions for positioning estimation of wsns in dynamic indoor environments," in *Proceedings of the First ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments*, ser. MELT '08. New York, NY, USA: ACM, 2008, pp. 37–42.

[169] H. Xia, "A simplified analytical model for predicting path loss in urban and suburban environments," *Vehicular Technology, IEEE Transactions on*, vol. 46, no. 4, pp. 1040–1046, Nov 1997.

[170] S. Misra, I. Woungang, and S. Misra, *Guide to Wireless Sensor Networks*, ser. Computer Communications and Networks. Springer, 2009.

[171] P. Santi, *Topology Control in Wireless Ad Hoc and Sensor Networks*. John Wiley & Sons, Ltd, 2006.

[172] J. Hansen and I. of Electrical Engineers, *Spherical Near-field Antenna Measurements*, ser. IEE electromagnetic waves series. P. Peregrinus, 1988.

[173] C. T. Inc., "Micaz wireless measurement system," accessed: Aug 07 www.openautomation.net/uploadsproductos/micaz_datasheet.pdf.

[174] M. Ilyas and I. Mahgoub, *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*. Taylor & Francis, 2004.

[175] C. Tarhini and T. Chahed, "Modeling of streaming and elastic flow integration in OFDMA-based IEEE802.16 WiMAX," *ScienceDirect*, 2007.

[176] J. Pearce, "Eb/N0 Explained," 2009, accessed: Mar 2 http://www.sss-mag.com/ebn0.html.

[177] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless communications and mobile computing*, vol. 2, no. 5, pp. 483–502, 2002.

[178] J. Ariyakhajorn, P. Wannawilai, and C. Sathitwiriyawong, "A comparative study of random waypoint and gauss-markov mobility models in the performance evaluation of manet," in *Communications and Information Technologies, 2006. ISCIT '06. International Symposium on*, Oct 2006, pp. 894–899.

[179] U. of Southern California, "ns2," Website, http://nsnam.isi.edu/nsnam/index.php/Main_Page.

[180] L. Bajaj, M. Takai, R. Ahuja, K. Tang, R. Bagrodia, and M. Gerla, "Glomosim: A scalable network simulation environment," Tech. Rep., 1999.

[181] Y. J., L. M., and N. B., "Random waypoint considered harmful," in *IEEE Infocom, San Francisco, CA*, 2003, pp. 1312–1321.

[182] J. Zheng, Y. Zhang, L. Wang, and J. Chen, "Adaptive location update area design for pcs networks under 2d markov walk model," in *Local Computer Networks, Proceedings 2006 31st IEEE Conference on*, November 2006, pp. 247–253.

[183] T. A. Patterson, L. Thomas, C. Wilcox, O. Ovaskainen, and J. Matthiopoulos, "State–space models of individual animal movement," *Trends in Ecology & Evolution*, vol. 23, no. 2, pp. 87–94, 2008.

[184] M. Bhardwaj and A. Chandrakasan, "Bounding the lifetime of sensor networks via optimal role assignments," in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, 2002, pp. 1587–1596 vol.3.

[185] M. Bhardwaj and A. P. Ch, "Chandrakasan , upper bounds on the lifetime of wireless sensor networks," in *In Proc. of IEEE International Conference on Communications (ICC)*, 2001.

[186] R. Technology, "Riverbed modeler," 2014, accessed: 5 Aug http://www.riverbed.com/products/performance-management-control/network-performance-management/network-simulation.html#Modeler_University_Program.

[187] K. B. Kredo and P. Mohapatra, "Medium access control in wireless sensor networks," *Computer Networks*, vol. 51, pp. 961–994.

[188] OMNeT++ Community, "What is omnet++?" Website, http://www.omnetpp.org/home/what-is-omnet.

[189] O. Technologies Inc., "Network Planning Operations," Website, http://www.opnet.com/solutions/network_planning_operations/index.html.

[190] Nsnam, "User information," Website, http://nsnam.isi.edu/nsnam/index.php/User_Information#The_Network_Simulator_-_ns-2.

[191] P. Levis and N. Lee, "Tossim: A simulator for tinyos networks," Collage of Engineering, Electrical Engineering and Computer Science, Tech. Rep., 2003.

[192] S. University, "Tossim," 2013, accessed: Oct 19th http://tinyos.stanford.edu/tinyos-wiki/index.php/TOSSIM.

[193] R. Technology, "Steelcentral netplanner," 2014, accessed: Oct 14th http://www.riverbed.com/products/performance-management-control/network-performance-management/network-planning.html.

[194] O. Technologies Inc., "Network Planning Operations," Website, http://www.opnet.com/university_program/research_with_opnet/research_requirements.html.

[195] S. K. C. University, "Real 5.0 overview," Aug 1997, accessed: Apr 29 http://www.cs.cornell.edu/skeshav/real/overview.html.

[196] D. A. R. P. A. (DARPA), "Darpa–creating & preventing strategic suprise," 2014, accessed: Apr 29 http://www.darpa.mil/default.aspx.

[197] U. U. of Southern California, "Collaborative simulation for education and research conser," Mar 2002, accessed: Apr 29 http://www.isi.edu/conser/index.html.

[198] J. Ousterhout and K. Jones, *Tcl and the Tk Toolkit.* Pearson Education, 2009.

[199] www.tcl.tk, "Tcl developer xchange," Feb 2014, accessed: Apr 29 http://www.tcl.tk/about/.

[200] T. F. Project, "Freebsd, the power to serve," Website, http://www.freebsd.org/.

[201] E. Siever, A. Weber, S. Figgins, R. Love, and A. Robbins, *Linux in a Nutshell.* " O'Reilly Media, Inc.", 2005.

[202] Oracle, "Solaris resource kit," Website, http://www.oracle.com/webapps/dialogue/ns/dlgwelcome.jsp?p_ext=Y&p_dlg_id=9272351&src=7058224&Act=8&sckw=WWSN10055327MPP002.GCM.9326.

[203] E. Bott and C. Siechert, *Microsoft Windows XP Inside Out.* Redmond, WA, USA: Microsoft Press, 2001.

[204] U. U. of Southern California, "Nam: Network animator," Jul 2002, accessed: Apr 29: .

[205] ——, "Using ns and nam in education," Oct 2001, accessed: 29 Apr http://www.isi.edu/nsnam/ns/edu/index.html.

[206] O. Community, "User manual omnet++ version 4.4.1," 2013, accessed: Apr 29 http://www.omnetpp.org/doc/omnetpp/manual/usman.html#sec116.

[207] S. Robinson, R. Brooks, K. Kotiadis, and D. Van Der Zee, *Conceptual Modeling for Discrete-Event Simulation*. Taylor & Francis, 2010.

[208] ORACLE, "Java," 2014, accessed: Apr 29 http://www.java.com/en/.

[209] T. E. Foundation, "eclipse – home / ide," 2014, accessed: Oct 19th .

[210] R. Blum and C. Bresnahan, *Linux Command Line and Shell Scripting Bible*, ser. Bible. Wiley, 2011.

[211] M. Löbbers and D. Willkomm, "A mobility framework for omnet++ user manual," Jan 2007, accessed: Apr 29 http://mobility-fw.sourceforge.net/manual/index.html.

[212] M. M. Qabajeh, A.-H. A. Hashim, O. O. Khalifa, L. K. Qabajeh, and J. I. Daoud, "Performance evaluation in manets environment," in *Australian Journal of Basic and Applied Sciences*, 2012, pp. 143–148.

[213] O. Community, "Omnet++," 2013, accessed: Apr 29 http://www.omnetpp.org/.

[214] G. Tabi, "Get into gned," 2008, accessed: Apr 29 http://www2.ensc.sfu.ca/~ljilja/ENSC835/Spring08/News/Presentations/OMNeT++/GetIntoGNED.pdf.

[215] A. Varga and O. Ltd., "Omnet++ user manual," 2014, accessed: Apr 29 http://www.omnetpp.org/doc/omnetpp/Manual.pdf.

[216] U. of Wisconsin, "octave," Website, http://www.gnu.org/software/octave/.

[217] *A Mobility Framework for OMNeT++—User Manual*. digital, 2007.

[218] D. Plummer, "Ethernet address resolution protocol: Or converting network protocol addresses to 48. bit ethernet address for transmission on ethernet hardware," 1982.

[219] A. Varga, *OMNeT++ Discrete Event Simulation System—User Manual*, Mar 2005.

[220] N. Mahalik, *Sensor Networks and Configuration: Fundamentals, Standards, Platforms, and Applications*. Springer, 2007.

[221] C. I. Inc., "Dell latitude d630 specifications," 2014, accessed: Jun 7 http://www.cnet.com/products/dell-latitude-d630/specs/.

[222] E. Raggi, K. Thomas, A. Channelle, T. Parsons, and S. van Vugt, *Beginning Ubuntu Linux – The complete introduction to Ubuntu*. Apress, 2010.

[223] C. L. Ubuntu and C. are registered trademarks of Canonical Ltd., "ubuntu," 2014, accessed: Aug 07 http://www.ubuntu.com/.

[224] B. O'Sullivan, *Mercurial: The Definitive Guide: The Definitive Guide*, ser. Animal Guide. O'Reilly Media, 2009.

[225] N. Patwardhan, E. Siever, and S. Spainhour, *Perl in a Nutshell: A Desktop quick reference*. " O'Reilly Media, Inc.", 2002.

[226] R. Blum, *Linux Command Line and Shell Scripting Bible*, ser. Bible. Wiley, 2008.

[227] R. Mecklenburg, *Managing Projects with GNU Make*, ser. Nutshell Handbooks. O'Reilly Media, 2004.

[228] J. McGregor and D. Sykes, *Object-Oriented Software Development: Engineering Software for Reuse*, ser. VNR Computer Library. An Nostrand Reinhold, 1992.

[229] J. Levine, *Linkers and Loaders*, ser. Operating Systems Series. Morgan Kaufmann, 2000.

[230] P. Misra, N. Ahmed, D. Ostry, and S. Jha, "Characterization of asymmetry in low-power wireless links: an empirical study," in *Distributed Computing and Networking*. Springer, 2011, pp. 340–351.

[231] I. A. Glover and P. M. Grant, *Digital Communications*. Pearson Education Ltd., 2004.

[232] S. Y. Shin, S. Choi, H. S. Park, and W. H. Kwon, "Packet Error Rate Analysis of IEEE 802.15.4 under IEEE 802.11b Interference," in *Conference on Wired/Wireless Internet Communications (WWIC 2005)*, vol. 3510, May 2005.

[233] K. Srinivasan and P. Levis, "RSSI is Under Appreciated," in *Proceedings of the Third Workshop on Embedded Networked Sensors (EmNets 2006)*, 2006.

[234] T. C. R. Network, "Class template std::map," 2013, accessed: Mar 6 http://www.cplusplus.com/reference/map/map/.

[235] K. H. Kwong, K. Sasloglou, H. Goh, T. T. Wu, B. Stephen, M. Gilroy, C. Tachtatzis, I. Glover, C. Michie, and I. Andonovic, "Adaptation of wireless sensor network for farming industries," in *Networked Sensing Systems (INSS), 2009 Sixth International Conference on*, June 2009, pp. 1–4.

[236] I. Andonovic, C. Michie, M. Gilroy, H. G. Goh, K. H. Kwong, K. Sasloglou, and T. Wu, "Wireless sensor networks for cattle health monitoring," in *ICT Innovations 2009*, D. Davcev and J. M. Gómez, Eds. Springer Berlin Heidelberg, 2010, pp. 21–31, 10.1007/978-3-642-10781-8_3.

[237] M. Caccamo, L. Zhang, L. Sha, and G. Buttazzo, "An implicit prioritized access protocol for wireless sensor networks," in *Real-Time Systems Symposium, 2002. RTSS 2002. 23rd IEEE*, 2002, pp. 39–48.

[238] M. Chatterjee, S. K. Das, and D. Turgut, "Wca: A weighted clustering algorithm for mobile ad hoc networks," *Cluster Computing*, vol. 5, no. 2, pp. 193–204, 2002.

[239] E. Battery Manufacturing Inc., "Alkaline manganese dioxide – handbook and application manual," 2008–2012, http://data.energizer.com/PDFs/alkaline_appman.pdf.

[240] K. Piotrowski, P. Langendörfer, and S. Peter, "How Public Key Cyptography Influences Wireless Sensor Node Lifetime," Innovations for high performance microelectronics, Presentation, 2006. [Online]. Available: http://www.ics.uci.edu/~steffenp/files/SASN_piotrowski.ppt

[241] K. Piotrowski, P. Langendoerfer, and S. Peter, "How public key cryptography influences wireless sensor node lifetime," in *Proceedings of the Fourth ACM Workshop on Security of Ad Hoc and Sensor Networks*, ser. SASN '06. New York, NY, USA: ACM, 2006, pp. 169–176.

[242] A. Hac, *Mobile Telecommunications Protocols for Data Networks.* Wiley, 2003.

[243] F. Bouabdallah, N. Bouabdallah, and R. Boutaba, "Load-balanced routing scheme for energy-efficient wireless sensor networks," in *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, Nov 2008, pp. 1–6.

[244] P. Li, Q. Shen, Y. Fang, and H. Zhang, "Power Controlled Network Protocols for Multi-Rate Ad Hoc Networks," *Wireless Communications, IEEE Transactions on*, vol. 8, no. 4, pp. 2142–2149, April 2009.

[245] S. Rahamatkar, A. Agarwal, and N. Kumar, "Analysis and comparative study of clock synchronization schemes in wireless sensor networks," *Analysys*, vol. 2, no. 3, pp. 536–541, 2010.

[246] A. Camilli, C. E. Cugnasca, A. M. Saraiva, A. R. Hirakawa, and P. L. CorrÃªa, "From wireless sensors to field mapping: Anatomy of an application for precision agriculture," *Computers and Electronics in Agriculture*, vol. 58, no. 1, pp. 25–36, 2007, precision Agriculture in Latin America.

[247] A. Barberis, L. Barboni, and M. Valle, "Evaluating energy consumption in wireless sensor networks applications," in *Digital System Design Architectures, Methods and Tools, 2007. DSD 2007. 10th Euromicro Conference on*, Aug. 2007, pp. 455–462.

[248] L. Jiang, J.-H. Huang, A. Kamthe, T. Liu, I. Freeman, J. Ledbetter, S. Mishra, R. Han, and A. Cerpa, "Sensearch: Gps and witness assisted tracking for delay tolerant sensor networks," in *Proceedings of the 8th International Conference on Ad-Hoc, Mobile and Wireless Networks*, ser. ADHOC-NOW '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 255–269.

[249] M. Stemm and R. H. Katz, "Measuring and reducing enegry consumption of network interface in hand-held devices," in *IEICE Transactions on Communications, E80-B(8)*, August 1997, pp. 1125–1131.

[250] G. de Meulenaer, F. Gosset, F.-X. Standaert, and O. Pereira, "On the energy cost of communication and cryptography in wireless sensor networks," in *Proceedings of the 2008 IEEE International Conference on Wireless & Mobile Computing, Networking & Communication.* Washington, DC, USA: IEEE Computer Society, 2008, pp. 580–585.

[251] M. Braasch and A. van Dierendonck, "Gps receiver architectures and measurements," *Proceedings of the IEEE*, vol. 87, no. 1, pp. 48–64, January 1999.

[252] Q. Xiao-gang and Q. Chen-xi, "An improvement of gaf for lifetime elongation in wireless sensor networks," in *Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference on*, Sept. 2009, pp. 1–4.

[253] A. Varga, *OMNeT++ Discrete Event Simulation System*, 2013 (accessed April 24, 2014). [Online]. Available: http://www.omnetpp.org/doc/omnetpp/manual/usman.html