Networking Protocols for Long Life Wireless Sensor Networks

Eoin O'Connell
BE
104571878

NATIONAL UNIVERSITY OF IRELAND, CORK

School of Engineering

Department of Electrical and Electronic Engineering

Thesis submitted for the degree of Doctor of Philosophy

2nd September 2014

Supervisors: Brendan O'Flynn, Victor Cionca
Head of Department/School: Nabeel Riza

Research supported by EI Electronics, IRCSET & Enterprise Ireland

Contents

List of Figures

I, Eoin O'Connell, certify that this thesis is my own work and I have not obtained a degree in this university or elsewhere on the basis of the work submitted in this thesis.

Eoin O'Connell

List of Tables

Acknowledgements

I would like to thank all of those who contributed to my thesis and helped me during this three year long process. Victor Cionca my co-supervisor, David Boyle of Imperial College London and Brendan O'Flynn supervisor were very supportive in helping to write this thesis.

This work was primarily funded by an industry partnership in conjunction with the Irish Research Council. The industry sponsor was EI Electronics. In the later stages this work was also supported by Enterprise Ireland. For the financial support that I received, I am very grateful.

Acronyms

ACK Acknowledgment

ADC Analog to Digital Converter

AWGN Additive White Gaussian Noise

BOM Bill of Materials

CCA Clear Channel Assessment

CSMA Carrier Sense Multiple Access

DODAG Destination Oriented Directed Acyclic Graph

FEC Forward Error Correction

IC Integrated Circuit

ISM Industrial Scientific and Medical

LED Light Emitting Diode

LO Local Oscillator

LPM Low Power Mode

LQI Link Quality Indicator

MAC Media Access Control

OSI Open Systems Interconnection

PCB Printed Circuit Board

PIR Passive Infrared

PLL Phase Lock Loop

RF Radio Frequency

RFIC Radio Frequency Integrated Circuit

RSSI Received Signal Strength Indicator

RX Receive

SFD Start of Frame Delimiter

SPI Serial Peripheral Interface

TDMA Time Division Multiple Access

TX Transmit

WSN Wireless Sensor Network

Abstract

My original contribution to knowledge is the creation of a WSN system that further improves the functionality of existing technology, whilst achieving improved power consumption and reliability. This thesis concerns the development of industrially applicable wireless sensor networks that are low-power, reliable and latency aware. This work aims to improve upon the state of the art in networking protocols for low-rate multi-hop wireless sensor networks. Presented is an application-driven co-design approach to the development of such a system. Starting with the physical layer, hardware was designed to meet industry specified requirements. The end system required further investigation of communications protocols that could achieve the derived application-level system performance specifications. A CSMA/TDMA hybrid MAC protocol was developed, leveraging numerous techniques from the literature and novel optimisations. It extends the current art with respect to power consumption for radio duty-cycled applications, and reliability, in dense wireless sensor networks, whilst respecting latency bounds. Specifically, it provides 100% packet delivery for 11 concurrent senders transmitting towards a single radio duty cycled sink-node. This is representative of an order of magnitude improvement over the comparable art, considering MAC-only mechanisms. A novel latency-aware routing protocol was developed to exploit the developed hardware and MAC protocol. It is based on a new weighted objective function with multiple fail safe mechanisms to ensure extremely high reliability and robustness. The system was empirically evaluated on two hardware platforms. These are the application-specific custom 868 MHz node and the *de facto* community-standard TelosB. Extensive empirical comparative performance analyses were conducted against the relevant art to demonstrate the advances made. The resultant system is capable of exceeding 10-year battery life, and exhibits reliability performance in excess of 99.9%.

Chapter 1

Introduction

1.1 Problem Statement

Before Wireless Sensor Network technology can be widely accepted as being reliable and dependable, a number of technical problems must still be overcome. The main limitations of existing WSN technology are reliability/ node lifetimes and the trade-off between the two. The industry partner who financed this body of work had similar ideas on the direction this research should take, their wish was to find out if a multi-hop network would be capable of providing highly reliable device status monitoring capabilities (≈10-20 minute basis) on an energy budget that would provide a battery life of 10 years (≈ 2,000 mAh capacity). The primary research challenge is thus, can a fully functional multi-hop network capable of reporting periodic sensor readings be developed with an average current draw budget of 10-15μA?[1] Another system requirement defined by the company is a packet delivery reliability of no less than 99.5% for packets which fall within the pre-defined reporting interval, and 99.99% reliability for packets that arrive within twice the reporting interval. Therefore, the multi-objective design constraint is captured: Can this average current be achieved while guaranteeing a reliability of greater than 99.5%?

[1] 10-15μA is the average drain required to provide the required battery life of 10 years

1.2 Introduction

In the physical world, there are many applications that require distributed monitoring/ sensing of variables that change over time. All industrial processes that require any form of sensed data, require some form of sensor network. Typically, this sensed data is used to create a feedback loop where input variables to the system must be changed dynamically or states must be monitored continuously. Traditionally, wired networks of distributed sensors are used for this purpose. An example of a wired sensor network is the CAN BUS network used in the automotive industry [49].

For applications where data is required to be monitored for relatively short periods of time (i.e. weeks), data loggers are often used at each point of interest for this purpose [87]. A good example of a distributed wired sensor network is a standard home security system. These systems typically sense the state of windows/ doors and report back their state (be it opened or closed), to a central control panel. Newer home security systems may also include a multitude of sensors such as Passive Infrared (PIR) motion detectors, temperature sensors or window vibration sensors. Such systems can be individually wired to a central location, or advanced systems may share a common signal bus, such as an RS485 or Ethernet bus.

Another example of a wired sensor network system is a network of sensors used to monitor industrial processes. Sensors used to monitor these processes are usually of the 4-20 mA current output variety. Each sensor is connected to a central panel via a 2-wire interface, the sensors generate a voltage at their output to drive a current in the wire that is proportional to the sensed variable. Current output type sensors provide excellent noise immunity and robust operation. At the control panel this current is converted into a voltage, before it is converted into the digital domain using an Analog to Digital Converter (ADC).

As with all solutions, there are advantages and disadvantages associated with the wired sensor network approach to distributed sensing. The advantages of which are listed below:

- **Reliability**: With a physical connection to a control panel, wired networks offer high reliability. Reliability in sensor networks, equates to sensor readings being available when demanded by the control panel. Of course catastrophic failures of connectors or structural damage to wiring ducts can occur, but the likelihood of such events is low.

- **Constant Power Source**: Wired networks can easily power the nodes in the network over the connecting wires.
- **Latency**: Independent of the type of wired system that is used (shared bus or individual connections), the latency of data acquisition remains very low. For sensors with individual connections to the control panel, there is zero latency in acquiring sensor readings. For systems that use a shared bus, there may be some minuscule latencies incurred when there is contention on the common shared bus and inherently due to the clock speed of the system. In general these latencies are still in the millisecond range, enabling actuation and control.
- **Longevity**: A properly installed wired sensor network with protective shields on wiring to counteract mechanical wear, provides relatively maintenance free and long-life operation.
- **Low Networking Complexity**: Wired networks for distributed sensing systems do not necessitate complex routing software to deliver sensor readings to the central control panel. This reduction in complexity leads to an increase in reliability as there are fewer subsystems which could potentially malfunction.

Some of the disadvantages associated with wired sensor networks are listed below:

- **Installation Cost**: Installing a durable wired sensor network designed for longevity, involves providing an infrastructure to support the system. In the case of wired sensor networks in buildings, planners must factor in the construction of subsystems to support the required infrastructure for the network. The costs associated with adding a wired network to a new construction or an existing construction are high. Wiring ducts must be placed to protect wires, the expensive wiring itself must be purchased and experienced and qualified personnel must be hired to carry out the installation process. With copper prices ever increasing [88], the cost of running wires around a large construction can become significant.
- **Difficulties in Retrofitting**: Retrofitting a wired sensor network to an existing construction is a time-consuming and costly process. Significant modification of the building must be carried out to support the

installation of the underlying infrastructure for the wired sensor network. The simplest example of this is the installation of a second light switch to actuate a light bulb in a residential premise. For this work to be aesthetically pleasing to the home owner, the installer must often dig out a channel in the concrete wall to conceal the wiring. In essence, a lot of planning and preparation must be performed to install a wired sensor network into an existing system.

- **Difficulty to Maintain**: Wired sensor networks are no doubt reliable in the vast majority of cases, and they will provide almost maintenance free operation. However, in the case of a wiring failure, an effort must be undertaken to pin-point the location of the fault and to replace the faulty wiring. This may involve the laborious task of rerouting hundreds of meters of wiring.

1.3 Wire*less* Sensor Networks, an alternative to Wired Sensor Networks?

WSN are an alternative to wired sensor networks, providing similar functionality while removing the need for physical connections to sensor nodes in the network. A WSN can be described as being a network of two or more distributed devices that monitor some environmental or physical states and communicate with one-another over a wireless channel. Since many Integrated Circuit (IC) manufacturers now mass produce Radio Frequency (RF) transceivers, WSN technology has become an economically viable alternative to wired sensor networks.

Each device in the network is called a node. A node is generally comprised of four key building blocks: 1.RF Communication Block, 2. Microcontroller/ Intelligence Block, 3. Sensors, 4. Power Source. A WSN usually contains a minimum of one central node which acts as the network coordinator. It gathers the sensor readings and interacts with some subsystem that enables visualisation/ usage of the sensed data. In this work, this central node will be referred to as the network sink.

WSNs offer many advantages when compared to wired sensor networks, some of which are listed below:

- **Retrofit-ability:** One of the main advantages of wireless nodes is their retrofit-ability. WSN's can easily be integrated into existing systems with minimal additional infrastructure required to support their installation. Ideally, WSN nodes can be deployed in an ad-hoc fashion, without detailed previous planning. The wireless nodes may only require the installation of a support bracket for mounting. Wireless nodes can also easily be moved to different locations with ease.
- **Low Installation Costs/ Material Needs:** Another advantage is the very low installation costs associated with the installation of a WSN system in new or old buildings. Although the wireless capable hardware needed in a WSN node may be slightly more expensive than a node that uses a physical connection, the cost savings are still justifiable. Recall from page 3, that the bulk of the cost associated with wired sensor networks is labor and wiring costs. In an early publication from 1986, the authors discuss the cost of wired networks for automobile applications [41]. With environmental restrictions and green products becoming ever more important, WSN's outperform wired networks with their minimalistic material requirements. In [10], the authors discuss the advantages of wireless networks over wired networks for industrial applications in terms of cost. The authors show that wired sensor networks only become economically viable when large numbers of sensors are required.
- **Servicing:** Assuming reliable networking protocols, fault finding in a WSN deployment is as easy as replacing the damaged node in question. Fault finding in wired sensor networks is a much more tedious task which can involve re-routing many meters of cabling.

Depending on the application and physical properties of the deployment area, the WSN will need to support various network topologies. The eventual shape of the topology of the network is influenced by the transmission range capabilities of the underlying RF system. Simplistically, network topologies can be categorised into 2 different subgroups.

If all nodes in the network have sufficient transmission range and they can communicate directly with the network sink, this hierarchy is referred to as single hop or 'Star Network'. As soon as one of the distributed wireless sensor nodes can no longer directly communicate with the network sink, it must use neighbouring nodes to relay its sensed data to the network sink. This type of network is referred to as a 'Multi-Hop Network'. The difference between a star and multi-hop network is depicted in Figure 1.1. From Figure 1.1, the Network sink in Red, WSN nodes in Blue. The Star network on the left has a maximum of one hop, the Multi-Hop network on the right requires 2 hops to keep the network connected.

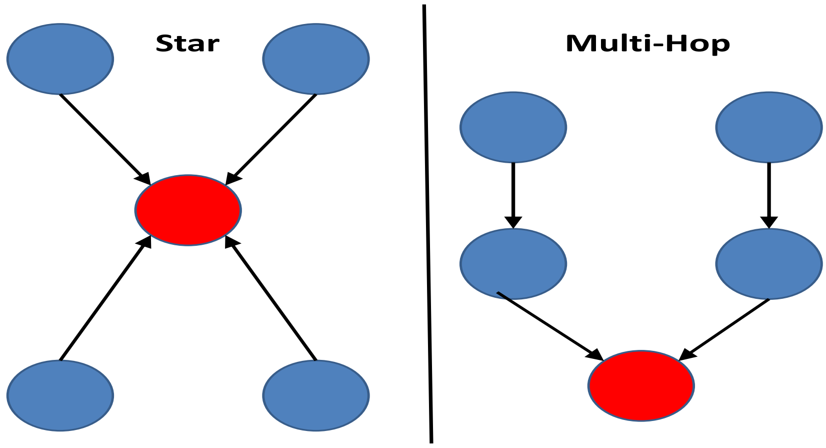


Figure 1.1: **Star and Multi-Hop topologies**

1.3.1 Difficulties and Challenges in low-power WSN

Networking protocols used in WSNs that support only star topology operation tend to be straightforward as the destination of all packets is the network sink. As soon as multi-hop networking capabilities are necessitated/ required, the complexity of the embedded software increases. The difficulties and challenges associated with WSNs are listed below, in general the points listed below apply to both star and multi-hop deployments:

- **Power Consumption**: For WSN technology to be truly *wireless*, all nodes except for the network sink should not require an AC mains power supply. Removing the necessity of an AC mains power supply adds to the idea of maintenance free WSN technology. Without an AC power supply, WSN nodes can be either powered from batteries, renewable Energy Harvesting sources [77], or hybrid sources [7]. In the case of operation from batteries, the average power consumption of the node must be low enough to achieve the desired lifespan. The node's lifetime will depend on the capacity of the battery in use, the leakage (self discharge) of the battery and the average power consumption of the node. When Energy Harvesting sources are used, the power consumption of the node must be carefully managed, on average it must remain equal to or below the level of power that is being harvested from the environment. Recently this area has become an active research area [42]. Recent work in [23, 30], addresses the area of energy aware routing protocols for multi-hop WSNs.

These limitations place strict requirements on the power consumption of the nodes in the WSN. The duty-cycle (portion of time spent in active-mode) of the radio, microcontroller and sensors must be very carefully managed to reduce the power consumption of the nodes to a level that enables sufficient lifetime from batteries or sustainable operation from Energy Harvesting sources. This has an impact on the functionality and complexity of the communication protocols. This work targets an improvement in energy efficiency by adopting a cross layer approach, the power saving techniques used in this work are presented in Section 3.3.

- **Reliability**: Due to the unpredictable, lossy and sporadic nature of the wireless channel [89], transmission failures will occur. This poses a challenge for low-power WSNs which must guarantee reliable data delivery at a low energy cost. Where end-to-end packet delivery mechanisms and resends come at a high energy cost, WSNs must provide multiple failsafe mechanisms to improve reliability. The overall reliability of a path over multiple hops decreases rapidly, it is equal to the product of the individual link reliabilities along a path. A path that offers 99% reliability for each of its 5 links, would equate to an overall reliability of 95.09% (99^5).

  The unpredictable and lossy nature of links in WSN's necessitates routing protocols which are capable of quickly reacting to dynamic changes in the network topology. Neighbours that appeared to offer stable links initially, may not offer the same stability at a later stage. The routing protocol must adapt quickly and find more stable links. The main difficulty and hence research challenge surrounding reliability in low-power WSNs, is the inherent trade-off between reliable and low-power operation. This work investigated this topic and offers improved performance on the balance between low-power/ reliable operation currently offered in the literature.

- **Latency**: In a wireless node, the radio transceiver is typically the most power hungry component. The key to achieving low-power operation is radio duty-cycling. This is primarily achieved by nodes being configured to wake periodically to check briefly for incoming data packets. Between periodic receive checks the radio transceiver is in a sleep (low-power) state. The main drawback associated with using a duty-cycled approach, is the latency incurred during each hop traversed in the network. Overall end-end latencies become large over multiple hops.

Due to the fact that nodes now spend the majority of their time in a non-listening state, in the worst case scenario, the latency incurred over each hop will be equal to the time interval between periodic receive checks. With each receive check contributing to the node's overall power budget, a balance must be struck between low-latency/ low-power operation. Recent work in [24, 25] uses a time synchronous approach to lowering latency in WSNs while achieving low-power operation. Time synchronous approaches require additional synchronisation overhead transmissions. Achieving low-latency operation while retaining low-power consumption still remains to be a research challenge. This work addresses the area of achieving low latency transmission over multiple hops while using a semi-synchronous approach.

- **Legal Restrictions**: A number of bands in the RF spectrum can be used license free for WSN applications. Depending on the frequency band of operation, regulatory bodies place restrictions on maximum allowed transmit power levels and transmit duty-cycles. The latter restriction is implemented to allow multiple devices to share a single band fairly. Transmit power level restrictions are in place to minimize carrier leakage into nearby license compulsory bands. Both of the aforementioned restrictions limit the functionality of the WSN, transmit power level restrictions limit the transmission range and necessitate multi-hop networking, transmit duty-cycle restrictions limit the functionality of the WSN in terms of the number of sensor updates which can be sent per unit time. Operation in the European 868 MHz ISM band limits transmit duty-cycle to 0.1% (3.6 s/hr) and maximum transmit power levels are 10 dBm.

  These restrictions pose many research challenges, transmit power level restrictions require efficient and intelligent multi-hop networking protocols to reliably deliver data over multiple hops. Transmit duty-cycle restrictions necessitate extremely careful management of the radio duty-cycle to remain within legal requirements. This work targets both of these areas; it aimed to improve the functionality of the network while remaining within strict legal requirements. This will be discussed in more detail in Sections 5.3.7 and 3.3.3.

1.3.2 Early WSN Deployments and Evolution

This section documents the progression of low-power WSN technology, specifically the evolution of real WSN deployments. WSN is a highly practical, goal oriented domain, with most of the research motivated by real world problems. Some of the earliest examples of research carried out on the practicality and feasibility of WSN technology are presented in [3, 83]. In [54], the authors study the applicability of WSN technology to industrial applications, in [43] the authors present a summary of the experience gained from a WSN deployment at a semiconductor plant. In [64] from 2005, the authors present results from a WSN used for structural health monitoring. The results presented in [43] are a star network, in [64] the topology is a multi-hop one, but no low-power radio techniques are employed.

In 2002, the authors present results and analysis of a large testbed style deployment of 150 nodes [26]. The authors study the performance of a flooding algorithm at different transmission power levels. One of the main findings of their research was that the range which a single node can cover, tends not to be circular, but rather it appears to be strongly directional. The authors also comment on the apparent asymmetry of links in the deployment. Detailed results on average flooding times are also presented.

In [82], the authors present a study into the challenges associated with self-organising multi-hop networking protocols. They address the main challenges, develop some fundamental ideas for routing protocols, and present simulated and empirical test data. The authors focus on achieving reliable communication over multiple hops. The authors conclude by stating that the reliability of the networks is closely coupled to the nodes ability to estimate link quality to neighbouring nodes, and their ability to keep an updated view of the nodes in their neighbourhood.

In 2006, researchers from Delft University of Technology in Holland presented results from an outdoor deployment of 100 nodes [45]. The deployment used the TinyOS operating system and Mica2 (868 MHz) hardware running the T-MAC protocol, its purpose was agricultural monitoring. The authors describe how the system collapsed and failed to provide reliable sensor readings (2%) over the course of the deployment time. They conclude by listing a number of issues that caused the deployment to fail in a hope to help others with future deployments. They say the main downfall of their work was the lack of pre-deployment testing and the lack of debug tools to enable real-time monitoring of the deployment.

In [85] 2002 and [67] in 2003, the first MAC protocols are presented that actively use radio duty-cycling to reduce power consumption to a level which enables battery powered operation. The MAC protocol presented in [67] was the default MAC protocol in the TinyOS operating system. The earliest routing protocol used in the TinyOS operating system was Multihop LQI [79]. In 2007, the authors in [9] present a synchronous approach to multi-hop networking for low power WSNs, the authors achieve low radio duty-cycle in their work, but the protocol does not support event detection capabilities and is designed only for extremely regular data reporting intervals. CTP now the standard routing protocol in TinyOS [28] was introduced in 2009, it is fully asynchronous. It improves upon Multihop LQI by using a new metric in selecting routing parents and is more dynamic through the use of periodic beacon transmissions. Based on results presented in [28], CTP achieves approximately 95% when used in conjunction with radio duty-cycled MAC protocols, it does not offer low latency operation.

Recent developments use either synchronous flooding approaches [25, 24] or new opportunistic routing techniques [44, 18, 4]. The synchronous flooding approaches of [25, 24] achieve high reliability and low latency at the cost of increased power consumption caused by synchronisation overhead. The MAC protocol used in synchronous flooding approaches stems from the work presented in [86]. [44, 18, 4] are asynchronous opportunistic cross layer solutions, these techniques are highly dynamic and can potentially offer high reliability and low latency in certain scenarios. Using opportunistic routing, paths are not fixed, parents are selected dynamically and opportunistically. The drawbacks associated with opportunistic approaches are potential duplicate packets, difficulties in detecting routing loops and shallow neighbourhood knowledge.

1.3.3 Evolution of IEEE Standards

In the year 2000 a task-force was established to create an IEEE standard for low-power, low-rate, short communication range devices. Special emphasis was placed on achieving low-power, highly functional operation, while using hardware with low manufacturing costs. This standard is the IEEE 802.15.4 standard.

The initial standards envisaged a physical layer that operates in one of three possible license free bands. One version in the 2.4 GHz band with a data rate of 250 kbps, another in the 902-928 MHz band (20-40 kbps), and a final frequency in the 868 MHz band (20-40 kbps). These were revised in 2006 to support more channels within the mentioned bands and higher data rates in the lower frequency bands.

The MAC layer was modified from the original standard, as the 802.15.4 standard uses a shorter maximum payload length compared to Ethernet frames. It accesses the medium using an asynchronous CSMA/CA protocol as opposed to time synchronised approaches of Time Division Multiple Access (TDMA) protocols. Recent work in 2012 resulted in the creation of a new lower-power amended 802.15.4e standard [11]. This variation provides lower-power MAC operation by using transmission scheduling in conjunction with a receiver duty-cycled approach. It also modifies the original Acknowledgment (ACK) frame, adding extra bytes allowing nodes to exchange data.

6LoWPAN [35](IPv6 over Low power Wireless Personal Area Networks), is based around the 802.15.4 standard, it uses header compression techniques allowing for IPv6 packets to be transmitted using 802.15.4 compliant hardware. The aim of 6LoWPAN is to extend the Internet Protocol to even the smallest of devices, providing low-power resource constrained devices with Internet connectivity. This is an important component in the emerging concept of Internet of Things [81].

RPL was developed in 2010 as a networking protocol that provides a mechanism for data transmission within a multi-hop network of nodes running 6LoWPAN [36]. It provides a routing mechanism for both upstream and downstream traffic.

1.3.4 Industrial WSN Standards

A number of industrial WSN systems exist today. Many of these have developed as standards and chipsets can be purchased to enable products which can participate in these networks. From the literature, it is very difficult to attain exact figures on typical power consumption, reliability, latency etc for industrial WSN protocols such as the ones listed below. Therefore a like for like comparison cannot be easily made between them.

1.3.4.1 Z-Wave[2]

Z-Wave is a low-rate mesh wireless networking standard, it was designed especially for home automation/ monitoring applications. It operates in the ISM bands of 868 MHz in Europe or 908 MHz in the USA, it uses a bit-rate of 40 kbps and is tailored for low-data rate applications. A typical Z-Wave system consists of a single gateway and multiple sensor/ actuation nodes in a home. The gateway has internet connectivity and can also interface with the WSN, acting as a bridge between the two. The distributed nodes in the home can serve multiple purposes, a variety of nodes can be purchased. Some of the functionality supported is sensing, actuation and monitoring.

1.3.4.2 Zigbee [3]

Zigbee is an industrial alliance and is based around an IEEE 802.15 standard. The first Zigbee specification was drafted in 2004. It was designed for mesh networking applications that require low data-rate, long battery-life and secure data transfer. Typical application spaces are; home automation, smart lighting, industrial control, medical data collection. It can support large network sizes due to its 16-bit addressing scheme. Most Zigbee compliant hardware operates at 2.4 GHz at a data-rate of 250 kbps, but sub-GHz hardware operating in the 868/900 MHz ISM bands also exists. Zigbee supports star and mesh topologies and uses a MAC layer defined in the IEEE 802.15.4 standard.

For low-power applications, Zigbee requires the use of periodic beacon transmissions to maintain synchronisation. Zigbee networks can contain three classes of devices: Network Coordinators, Routers and End devices. Coordinators act as network sinks and coordinate the wireless network, Routers are nodes that are capable of running application software and also receive and forward network packets, End-Devices are nodes which have only one task, they either sense or actuate and communicate with their parent, they are not capable of forwarding packets for neighbouring nodes. Zigbee uses AODV (Ad hoc On-Demand Distance Vector) as its routing layer protocol [65].

[2] http://www.z-wave.com/
[3] https://www.zigbee.org/

1.3.4.3 Wireless-HART

Wireless HART (Highway Addressable Remote Transducer) uses a highly time-synchronised, self organising mesh networking system and is an industry standard since 2007. It is based around a TDMA MAC protocol. WirelessHart is a wireless extension to the earlier HART wired protocol for 4-20 mA current output type industrial sensors. It is used primarily in industrial process control. It includes a proprietary frequency hopping technique and all devices are time synchronised and communication takes place in pre-scheduled fixed length time-slots. In [47] the authors state that industry demanded secure and reliable communication, but static and multi-path fading sometimes blocked ZigBee due to its use of one static channel. Therefore Wireless-HART is preferred over Zigbee due to its frequency hopping agility and secure communication.

1.4 Research Challenge

In Section 1.3.1 the challenges associated with achieving reliable operation in low-power WSNs were presented. In Section 1.3.2 the evolution of WSN technology was presented, giving an overview of today's current state of the art and the techniques used to achieve low-power, low-latency, reliable operation. The most recent work uses either fully synchronous or asynchronous protocols to achieve their desired performance characteristics. The challenge is to achieve an optimal balance of low-power, low-latency and ultra-reliable operation. To achieve this goal, the designed solution uses a novel semi-synchronous cross-layer protocol, built upon ad hoc hardware. The power consumption overhead associated with using a fully synchronous approach was considered to be excessive when a target battery lifetime of 10 years is expected.

The primary research challenge and goal of this work is to create a multi-hop networking protocol stack, capable of providing sensor readings on a 5-minute basis, while achieving a power consumption level that would ensure 10-year operation from a battery of 2,000 mAh capacity (see Section 1.5). If this level of power consumption could be achieved, WSN technology could become a very attractive technology of the future due to low maintenance costs. In terms of reliability, this work aims to provide a quality of service which guarantees reliability of no less than 99.5%. In the context of this work, reliability is measured as the timely arrival of sensor readings from deployed nodes at the network sink. Sensor readings should arrive within 20% of the pre-defined reporting interval. A further challenge of this work is to guarantee 99.9% + reliability on sensor readings expected to fall within twice the pre-defined reporting interval.

For many applications such as fire-detection and security, close to real-time sensor readings are of great importance to the system's end users. To date, reducing inherent latency in low-power duty-cycled networks was achieved using either fully synchronous or opportunistic approaches. This work investigated if a latency aware routing protocol could be created, where parents are selected based on knowledge of the end-end latency they would provide.

1.5 Industry Performance Requirements

This PhD research was co-funded by EI Electronics [4]. EI Electronics, is an acknowledged market leader in Residential Fire Safety Products, one of Ireland's largest indigenous electronic companies. EI employs over 500 people at their 18,000 m^2 campus in Shannon, Co.Clare. In recent years EI Electronics incorporated wireless connectivity into their range of fire safety products, this technology is called RadioLINK. This technology allows up to 32 devices to be learned together to create a home network of fire safety products. Any alarm on the premise which detects above threshold levels of smoke/ heat sounds its horn, it also triggers other alarms on the premise to sound their horns via transmission of a radio message. The radio system sends status update messages every 20 minutes. This enables basic star topology alarm monitoring, but the system lacks networking capabilities where multi-hop transmissions are required/ necessitated.

To guarantee the 10-year battery life for their products, the average current consumption must be $\approx 10\mu$A. To achieve this, their current system uses a receive check interval of 2.8 s and simple status transmissions every 20 minutes. Recent changes in regulations that govern wireless fire safety products require for more stringent monitoring of residential fire safety products. These changes include the installation of a user control panel that is capable of alerting home users if any device on the premise has malfunctioned. For this reason, the company were interested in investigating the feasibility of implementing a

[4] www.eielectronics.ie

network stack that is capable of providing ultra reliable multi-hop communication on a current consumption budget which guarantees a 10 year battery life. The system should provide similar functionality in terms of responding to event detection but with the added functionality of device status monitoring on a 10-20 minute basis.

Fire detection is one example application for the work developed as part of this thesis. There are a number of other applications that suit the performance characteristics designed into the networking protocols designed in this work. Examples of applications that require periodic monitoring and event detection capabilities are: Home Security, Environmental Monitoring, Structural Health Monitoring. These applications require periodic sensor readings/ updates from each node in the network while supporting event detection capabilities.

The system currently used by EI Electronics uses a flooding mechanism to alert all nodes in the network in the event of fire detection. This works by nodes broadcasting when a fire event is detected and neighbouring nodes simply repeat this message. This mechanism is implemented directly in this work as it is a proven technique, achieving timely network wide flooding of alarm events. As alarm events occur extremely infrequently, these do not impact on the average power consumption of the network. The main goal of this work is to create an energy efficient and reliable way of delivering a different class of messages, periodic monitoring messages. As each node in the network must send periodic updates to the network sink, these transmissions must be carried out in an energy efficient manner to meet the system battery lifetime requirements.

1.6 Thesis Content

This thesis explores the development of a WSN system from the ground up, with low power, low latency and high reliability as the main design considerations throughout the entire process. Chapter 2 describes the development of a custom WSN platform operating at 868 MHz. It was designed to help meet the system requirements described in Section 1.5. Included in Chapter 2, is a brief overview on commercially available WSN platforms and an overview of RF transceivers suited for low power WSN applications.

Chapter 3 describes the design, modeling and evaluation of a novel semi-synchronous hybrid MAC protocol. It is called *IX-MAC* as it is an improved version of the XMAC protocol [8]. *IX-MAC* optimises some previously known techniques, integrating these and new novel techniques into a single MAC protocol for receiver duty-cycled low-power networks. It was designed for applications where low power consumption, high reliability and low latency are desired. Section 3.2 of Chapter 3 discusses the current state of the art in MAC protocols for low-power WSNs. Chapter 4 presents the results of models that were developed around the design of the *IX-MAC* protocol. Section 4.6 empirically compares the operation of this work against a number of other protocols.

IX-MAC targets monitoring applications where long battery life (10 year)/ operation from energy harvesting sources are required. It achieves ultra low-power operation through the use of a semi-synchronous hybrid approach. It achieves synchronous operation without the need for additional packet transmissions or data exchange as in previous work. Reliability is also improved with respect to the current art, this is achieved through multiple failsafe mechanisms, precision timing and a unique CSMA/ CA implementation.

Chapter 5 documents the design and implementation of a routing protocol for multi-hop networking applications. Section 5.2 gives an up to date overview of the current art in routing protocols for low-power WSN applications. Section 5.3 describes and explains the various building blocks which the routing protocol is comprised of, and Section 5.4 presents results of a 51 node deployment spanning three stories. Chapter 6 presents a side-by-side comparison against the state of the art using the TelosB hardware platform.

The routing protocol supports a converging tree topology where multiple sensor nodes report back to central sink node/ nodes. Coupled with loose time synchronisation at the MAC layer, it offers novel latency aware parent selection in attempt to lower the inherent (often large) end-end latencies associated with duty-cycled, multi-hop, WSNs. Multiple other parameters are considered during parent selection. Data required for parent selection is intelligently exchanged, avoiding the need for nodes to overhear beacon transmissions before joining the network. Previous techniques necessitated periodic beacon transmissions to update the tree topology.

The thesis is concluded in Chapter 7 where the outcomes and main contributions of this work are highlighted. Some suggestions on how this work could be continued are given in Section 7.2.

1.7 Publications

During the time frame of this thesis a number of different articles were published, these also capture the major contributions of this work.

- In [62], this work involved an investigation into clock drift, latency and energy efficiency in duty cycled, multi-hop WSN's. The title of the paper was 'Clocks, latency and energy efficiency in duty cycled, multi-hop Wireless Sensor Networks'. This work formed the basis of the lightweight and efficient neighbour schedule learning scheme used at the MAC layer which is one of the main contributions of this work.
- In [60], a paper was presented on the neighbour schedule learning system implemented in Chapter 3, specifically it looked at the reduction in the transmit duty cycle for operation in ISM bands. This paper was awarded a 'Best Paper Award' at the SENSORCOMM 2013 conference. The title of the paper was 'Techniques for Increasing Network Functionality while Remaining within Legal Maximum TX Duty Cycle Requirements'. This work described one of the main contributions of this work, a MAC protocol that increases the functionality of the network while adhering to legal TX duty cycle restrictions.
- In [61], a demo was presented at the REALWSN conference. This demonstration showcased the multiple features described in Chapter 5 and the low-power operation of the *IX-MAC* protocol. The title of the demo was 'Cross Layer Design for Low Power, Low Delay, High Reliability Radio Duty-Cycled Multi-hop WSNs'. This publication detailed another one of the main contributions of this work, the lightweight operation of the latency aware, lightweight routing protocol.
- In [58], a paper was presented at the 2014 SenSys Doctoral Colloquium in Rome. The title of the paper was 'Clean Slate System for Minimum-Power Maximum-Reliability Low-Rate Multi-Hop Wireless Sensor Networks'. This publication captured another one of the main contributions of this work, the encapsulation and integration of a number of power saving and reliability enhancing features into a single cross layer protocol for low-rate multi-hop WSN's.
- In [63], a paper was presented at The International Conference on Design of Reliable Communication Networks. This publication described in detail

the operation of the *IX-MAC* protocol and presented the parameter optimisation and reliability enhancing features of the protocol. The title of the paper was 'Energy & Reliability Optimal MAC for WSNs'.

- In [59], a journal paper was published that described the operation of the transmit duty cycle reduction techniques, it included a comparative analysis of the technique and results from the 52 node deployment. The title of the journal paper was 'Software Techniques for Maximizing Network Functionality in Duty Cycle Restricted ISM Bands'.

Two further journal publications are under review. One in the Microelectronics Journal (ISSN: 0026-2692) and one in Ad Hoc & Sensor Wireless Networks (ISSN: 1551-9899).

1.8 Contribution to Knowledge

- This work contributes to the existing knowledge in WSN technology by describing the operation of a lightweight semi-synchronous system.
- This work showed that semi-synchronous operation can be achieved without nodes sharing a common notion of time or without the need for nodes to exchange scheduling information.
- The extensive experimental evaluation proves it outperforms the state of the art in terms of power consumption and reliability.
- Using the neighbour schedule learning technique described, this work successfully decouples the energy required to transmit data from the receive check interval.
- It showed that layer 2 receive check mechanisms can perform well in terms of power consumption when a careful design is implemented.
- It successfully combines a number of power saving and reliability enhancing features into one communication protocol.
- This work achieves latency aware route selection and describes precisely how it can be achieved.
- The routing protocol describes how exchange of routing information can be achieved without the transmission of beacons.

- This thesis proves that an ultra reliable multi-hop WSN can provide a battery life which exceeds 10 years.

Chapter 2

Design of Fire Mote

This chapter deals primarily with a description of the WSN hardware platform that was developed in the context of this work. An integrated micro-system capable of meeting the fundamental technical requirements specified in Section 1.5, was designed. This chapter will begin with a brief overview on state of the art RF transceivers which are suitable for use in low-power WSNs and then justify the selection of the low-power hardware components. The various design steps associated with the manufacturing process will also be described. A brief overview of existing commercial WSN hardware is also be given.

2.1 SoA Transceivers for low-power WSNs

Software considerations aside, the choice of the wireless transceiver will have an impact on the functionality and power consumption of the WSN node. Choosing an RF transceiver is highly application specific, and in the application space of low-power duty-cycled WSNs, some desirable performance characteristics are:

- Fast/ Power Efficient Startup: Because the RF transceiver will be required to switch frequently from sleep to active modes, the manner in which this operation is carried out becomes critical. The startup phase typically consists of 3 phases, 1: Powerdown → Oscillator Active, this phase involves taking the RF transceiver from an inactive state to one where its crystal oscillator is active and stable, 2: Oscillator Active → Phase Lock Loop (PLL) Locked, this phase involves enabling the frequency synthesizer and Local Oscillator (LO) blocks, 3: PLL Locked → Transmit (TX)/ Receive (RX) Ready.

- Low Sleep Current: Considering that in a low-power WSN the RF transceiver will spend more than 99% of its lifetime in this powerdown state, the current drain in sleep mode is significant.
- Register Data Retention: RF transceivers must be configured by the host microcontroller with the application specific settings (frequency, bit-rate, RX bandwidth etc). This involves writing multiple register settings to configure the transceiver. Some RF transceivers will lose these configuration settings in powerdown mode and will require reconfiguration on startup. This process can become energy inefficient if the host microcontroller must reconfigure the transceiver after every startup.
- Receiver Sensitivity: With the aim of reducing the number of hops in multi-hop networks, the sensitivity of the RF transceiver will impact this performance characteristic. A transceiver with a sensitivity of -90 dBm may require 5 hops to cover a specific area, while a transceiver with -100 dBm sensitivity may achieve the same coverage in fewer hops, thus increasing reliability.
- Receive Mode Current: Each node in a multi-hop WSN must wake periodically to check if neighbours are trying to send data, typically a node will wake every 1-2 seconds and remain in RX mode for a few milliseconds. The magnitude of the current in RX mode will have a profound effect on the node's battery life. Neglecting startup times and startup currents, Node A which drains 20 mA for 1 ms consumes 4 times more energy than Node B which drains 5 mA for 1ms. This energy per receive check operation is one of the main contributors to a node's overall power budget.

Other characteristics such as low TX mode current drain, high blocking immunity and high spectral purity are also desirable. TX mode current drain is not considered to be as high priority as RX current drain as the time spent in TX mode is less than that spend in RX mode, due to periodic receive checks.

Listed in Table 2.1, are a number of transceivers that are commonly used in WSN and transceivers which have been identified as suitable for use in low-power WSN applications.

The commercially available transceivers listed in Table 2.1, achieve varying current consumption figures for receive and transmit mode. In terms of receive current consumption performance, the best performer is the SX1211 transceiver

Table 2.1: **WSN Transceivers**

Transceiver	**Frequency**	**RX(mA)**	**TX(mA)@0 dBm**	**Sleep(μA)**
CC2520	2.4 GHz	18.5	25.8	<1
CC2420	2.4 GHz	18.8	17.4	<1
CC430	300-930 MHz	18	16	<1
SX1211	860-960 MHz	3	15	0.1
SX1272	860-1000 MHz	10	15	0.1
CC1000	868 MHz	9.6	16.5	<1
Si106x	142-1050 MHz	10	12	0.6
SPIRIT1	150-956 MHz	9	12	0.6
[39]	868 MHz	1.5	6	NA
[76]	868/915 MHz	1.6	1.8 (-6dBm)	NA

from Semtech (3 mA)[1], the best performers in terms of TX current drain are the Si106x SOC transceiver from Silicon Labs and the SPIRIT1 transceiver from ST Microelectronics ($\approx$ 12 mA). If propagation and transmission range were to dominate the selection of the RF transceiver's frequency, transceivers that operate at 433 MHz and below would be an attractive option. According to the Friis equation, for identical transmit power levels, receiver sensitivities and antenna gains, operating at a lower frequency provides increased transmission range. The Friis equation is given below 2.1:

$$\frac{P_{\mathrm{r}}}{P_{\mathrm{t}}} = G_{\mathrm{t}} G_{\mathrm{r}} (\frac{\lambda^2}{4\pi R}) \tag{2.1}$$

In Equation 2.1, G_t is the transmitter's antenna gain and G_r is the receiver's antenna gain. As λ, the wavelength of propagating signal is increased, so too does the ratio of the received power (P_r) to the transmitted power (P_t). Simply put, low frequency signals propagate further for the same transmitted power/ receiver sensitivity and antenna gains. Unfortunately, there are some drawbacks associated with using RF transceivers that operate at low frequencies i.e. 433 MHz and 169 MHz Industrial Scientific and Medical (ISM) bands. The issues surrounding the use of low frequency RF transceivers in a hardware solution that is both energy efficient and physically compact are; bit rate/ bandwidth limitations and difficulty in efficient miniaturised antenna design. The task of creating a compact but efficient antenna becomes more challenging as wavelengths decrease [2]. A 1/4 wavelength antenna at 433 MHz is 17.32 cm and 44.38 cm at 169 MHz. These dimensions are quite large when compared to the

[1] http://www.semtech.com/images/datasheet/sx1211.pdf
[2] http://www.ti.com/lit/an/swra161b/swra161b.pdf

Table 2.2: **ISM Frequency Band Comparison**

Area	2.4 Ghz	868 MHz	433 MHz
Range	Poor	Good	Excellent
Bit-Rate/ Bandwidth	kbps-Mbps	bps-0.5 Mbps	bps-kbps
Antenna Design	Easy	Feasible	Challenging

required form factor for some applications. At these wavelengths, antennas which can be integrated into a Printed Circuit Board (PCB) are also not an option, considering the required size they would need to be to create an efficient antenna. There are also cost implications with larger PCBs.

Bit rate and bandwidth limitations also become an issue at lower frequencies. Higher carrier frequencies have greater capacity when it comes to throughput and bandwidth. Commercial RF transceivers that operate at and below 433 MHz cannot compete with the bit rates which RF transceivers operating at 868 MHz and 2.4 GHz can provide. Transmitting at higher bit rates has the effect of reducing transceiver active time per message and is pivotal to low-power operation. A data packet (of equal length) being transmitted at 250 kbps takes 5 times less time than a packet being transmitted at 50 kbps, therefore at identical TX current consumptions energy per transmitted packet is reduced for higher bit rates. This effect also increases the energy per received packet due to longer packet lengths. The disadvantage of using higher bit rates is reduced receiver sensitivity. A comparison of the various frequency bands is given in Table 2.2.

The receive sensitivity of any electromagnetic transceiver is dictated by the Signal to Noise Ratio present at the receiver's input. Increasing the allowed bandwidth of the incoming signal also increases the thermal noise present in the system [84, 56]. Therefore, a compromise must be struck between bit-rate of operation and desired receiver sensitivity.

Considering the topics discussed above, a decision was made to choose an 868 MHz radio transceiver. 868 MHz provides a good trade-off between bit-rate, transmission range and antenna size for the required application.

2.2 Commercial WSN Platforms, an Overview

A number of WSN platforms exist for both commercial and academic use. These are 'ready to go' type solutions and typically contain multiple on-board sensors for basic environmental sensing such as light intensity, temperature and humidity. Over the past decade numerous WSN platforms were created to suit a large range of different sensing applications. Some of the developed hardware platforms were developed with integrated on-board sensors and others were designed to be modular. The modular designs usually contain a main layer with processing and communication capabilities, various sensor layers can then be purchased additionally and plugged into the main layer. One such example of a modular type design is the Mica2/ MicaZ from CrossBow Technology.

This section serves as a brief overview of the most popular commercially available WSN hardware platforms. It is not an exhaustive state of the art review of all commercially available hardware platforms, as these already exist in the literature [40, 33].

- The TelosB/ TMote Sky (Figure 2.1) was developed by the University of California, Berkeley [66]. It was designed to be low-power and easy to program. It is also open source and IEEE 802.15.4 compliant. It contains a CC2420 2.4$\tilde{G}$Hz transceiver and an MSP430F1611 microcontroller with 48 kB of ROM and 10 kB of RAM. Amongst an array of sensors such as temperature/ humidity and light levels, it also supports USB programming and features additional external flash memory. Most WSN testbeds use the TelosB platform, Kansei [22], WISEBED [14], Indriya [16] and Twist [32] to name just a few. It also features an AA battery holder. The TelosB can still be purchased for a price of €77 [3]. It achieves a sleep current of approximately 8-10 μA.
- The MicaZ mote (Figure 2.2), at its core contains an Atmega 128L with 128 kB of ROM and 4 kB of RAM and a CC2420 2.4 GHz transceiver. The MicaZ platform is modular in that it does not contain any on-board sensors, rather it supports external sensor expansion boards via a 51-pin connector. Its modular design provides an easily customisable system whereby a wide variety of sensor platforms can be plugged into its main layer. The Atmega 128L microcontroller is not as low-power as the MSP430 used in the TelosB platform. The MicaZ is used in the Motelab [80] testbed. It achieves a sleep current of approximately 15 μA.
- The Sun SPOT WSN hardware platform features a 180 MHz 32-bit ARM

[3] https://telosbsensors.wordpress.com/tag/telosb-mote-price/
[4] http://inst.eecs.berkeley.edu/ cs194-5/sp08/lab1/
[5] http://www.eecs.harvard.edu/ konrad/projects/motetrack/

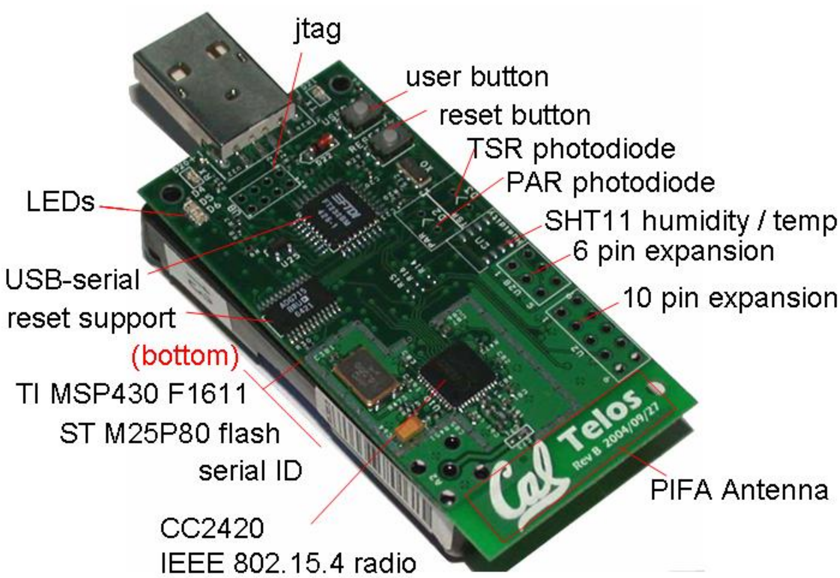


Figure 2.1: **TelosB platform.**[4]

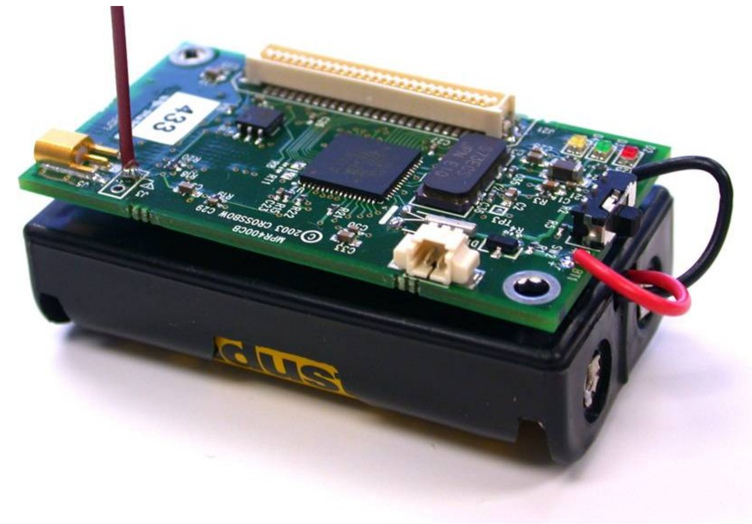


Figure 2.2: **Commercially available MicaZ platform.**[5]

processor with 512 kB of RAM and 4 Mb of flash. The main processor board contains the 32-bit ARM processor paired with a CC2420 transceiver, it uses a modular design like the MicaZ. It ships with a generic sensor platform that plugs into the main processor layer. The Sun Spot WSN motes are programmed in Java and pre-written software libraries are provided. The system is aimed at users with little or no experience in interfacing with low level hardware components through protocols such as UART, SPI and I2C. It runs the Squawk virtual machine.

The Sun SPOT platform is not as low power as the MicaZ or the TelosB due to its 32-bit ARM processor, it also achieves a high sleep mode current of 36 μA. These traits make this platform unsuitable for long battery life applications, but more suitable for processor intensive applications.

One of main negative points associated with the aforementioned hardware platforms is the relatively high sleep-mode currents. For a hardware platform to be able to guarantee a 10-year battery life from a 2,000 mAh battery cell, a sleep mode current of 10-15 μA already consumes a large portion of the current budget. This was one of the factors that motivated the development of new WSN hardware platform.

The majority of the commercially available WSN platforms use 2.4 GHz radio transceivers. This frequency band is highly congested as it is shared with some 802.11 WLAN bands and the useful communication range is also reduced when compared to lower frequencies. This was another motivating factor to design a new WSN hardware platform with an RF transceiver that operates in lower frequency bands.

2.3 Hardware Selection

TinyOS and Contiki are community developed open-source real-time operating systems for WSN development. It was decided not to use these operating systems as they would limit the hardware selection to components which are supported by the TinyOS or Contiki operating systems. The debug environments of these operating systems are also less powerful than the C based integrated development environments provided by the popular microcontroller manufacturers.

A number of low-power 868 MHz transceivers were reviewed and evaluated, finally the SX1211 transceiver from Semtech was chosen. The Semtech transceiver operates from 863-960 MHz. It consumes 3 mA in RX mode and 25 mA in TX mode at 10 dBm output power. It operates over a wide voltage supply range of 2.1-3.6 V and provides a standard Serial Peripheral Interface (SPI) interface. There are few configuration registers ($\approx$ 20) and the transceiver supports address based packet filtering and automatic CRC checking. Its oscillator startup time is reasonable at 1-1.5 ms and consumes just 1.1 mA in PLL lock state. It does however have some drawbacks, it requires 600 μs to go from an oscillator stable state to TX or RX modes and its receive sensitivity is slightly lower than other competitor transceivers. It also has a relatively low-cost Bill of Materials (BOM), requiring few external components.

In terms of microcontroller selection, a decision was made to use a PIC24F from the XLP (eXtreme Low Power) family. The reasoning behind this decision was based on three key points; 1) Ability to Debug, Microchip's microcontrollers have excellent development tools, both hardware and software. Microchip also provide good sample code and documentation is excellent. 2) Flexible Peripherals, Microchip's peripherals offer great flexibility, for example the MSP430 is a 16-bit processor but its SPI peripheral only supports 8-bit wide unbuffered transfers. The PIC24F provides eight-word (128 bit) deep SPI buffers allowing for bulk transfers with minimal software overhead. 3) Current consumption in low-power modes, the PIC24F XLP family achieves 0.6 μA with the real time clock enabled and sourced from an external 32.768 kHz crystal resonator. The MSP430F1611 typically consumes 2 μA in the equivalent low-power mode.

2.4 First Stage Prototype

As a proof of concept and to evaluate the performance of the combination of the SX1211 transceiver interfaced to the PIC24F microcontroller, a prototype was built. It consisted of 2 separate PCBs connected together via wires. The first PCB was an SX1211 evaluation kit, the second was a PIC24F PCB from a previous project. The prototype is shown below in Figure 2.3, where the SX1211 evaluation kit is on the left, underneath a PCB containing a PIC24F.

The prototype proved the concept and gave expected current consumption results. RX current with the microcontroller in sleep mode was 3.1 mA and TX current was 25.8 mA at 10 dBm output power (measured with 3 V supply).

2.4.1 Sleep Current Minimisation

The expected sleep current for the pair of ICs was 0.6 μA for the PIC24F and approximately 0.2 μA for the SX1211 transceiver. To achieve close to the expected sleep current of 0.8 μA for the pair a number of pullup/ pulldown resistors were required. SX1211 pins DATA and CLKOUT required 47k pullup resistors. MISO required a 47k pulldown resistor. After these modifications, a sleep mode current of 1.1 μA was measured. A screen-shot of the measured

Figure 2.3: **Prototype of SX1211 and PIC24F**

sleep current is given below in Figure 2.4. It was measured using a calibrated N6705B DC Power Analyser.

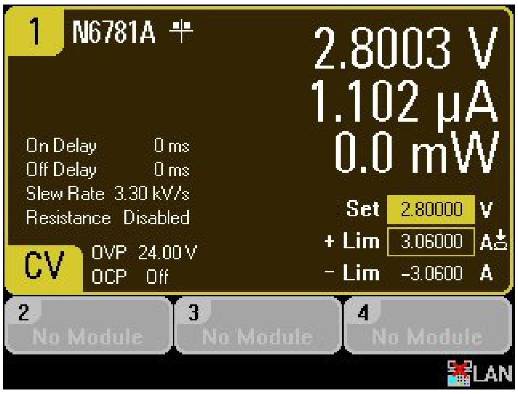


Figure 2.4: **Sleep Current of FireMote Hardware**

The expected current of 0.8 μA was reached, but at a supply voltage which approached the minimum operating voltage of the system. A difference of 0.3 μA was acceptable considering that electrolytic capacitor and battery leakage currents are typically much higher.

2.5 Wireless Reprogramming Support

To provide an easy mechanism to support over the air reprogramming, it was decided to include external RAM on the PCB. Over the air reprogramming involves sending a new program memory image wirelessly and for the node to overwrite its own internal memory with this new image. Over the air reprogramming techniques already exist in WSN [31, 78], these techniques use external flash ROM to buffer received radio packets. Because the new program memory image is often larger in size than the microcontrollers internal RAM, this necessitates external memory to buffer the new incoming program memory image. In this work, fast wireless reprogramming is achieved by using external RAM connected to the host microcontroller via a high speed SPI bus. This technique offers high speed buffering of incoming chunks of the new program memory image. External ROM would also provide similar functionality but at a reduced speed due to the time taken to write/erase each page of flash ROM.

The RAM that was chosen was the 23LC512 from Microchip, it provides 512 Kbit of memory with write speeds of up to 20 MHz. In standby mode it drains 4 μA and in active mode 3 mA. To remove the 4 μA of current that the RAM would contribute to the overall quiescent current of the hardware, a GPIO on the microcontroller is used to power the supply pin of the IC.

2.6 On-board Sensors

An analog output temperature sensor IC was included in the design. It is manufactured by Microchip and its part number is MCP9700A [6]. It consumes 12μA of current and its accuracy is $\pm 1\,^{\circ}$C. To reduce this current drain from the overall budget, the power to the IC is only applied during measurement phases.

[6] http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en027103

2.7 Level Shifters and Interfacing to Smoke Alarm Housing

To allow the PCB to be able to interface with the electronics in the smoke alarm housing, a standard pitch 4x2 header was used. The smoke alarm module provides 9V logic level signals when it detects smoke levels above a threshold, or if the button on the housing is pressed for a button-test. To interface these 9V signals to the 3V powered microcontroller, simple level shifters using single N-Channel MOSFETs are implemented. The gate signals are attached to the 9V control signals from the smoke alarm housing through 10 kΩ pull-down resistors, the source is connected to ground and the drain is connected to the microcontroller's input, internal software enabled pull-up resistors are used on the microcontroller's input pins to allow the voltage on the drain to alternate between 0 and 3V as the gate voltage is toggled.

The microcontroller must also provide a 9V logic signal to the smoke alarm if an alarm radio message is received. This triggers the piezoelectric horn in the smoke alarm's housing. A 9V supply line is available on the 4x2 header and this is used in combination with an N-Channel P-Channel level shifter pair. The level shifter circuit is shown in Figure 2.5.

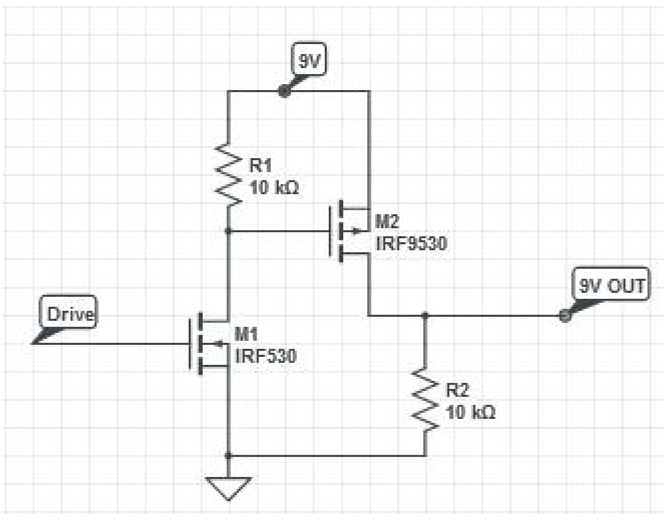


Figure 2.5: **3-9V Level Shifter**

2.8 Battery

A non-rechargeable lithium based battery was selected, it has a nominal voltage output of 3V and is manufactured by FUJI. The part number is CR$\frac{2}{3}$8L and its capacity is 2,000 mAh to 2V @ 23 °C [7].

2.9 Design and Manufacturing

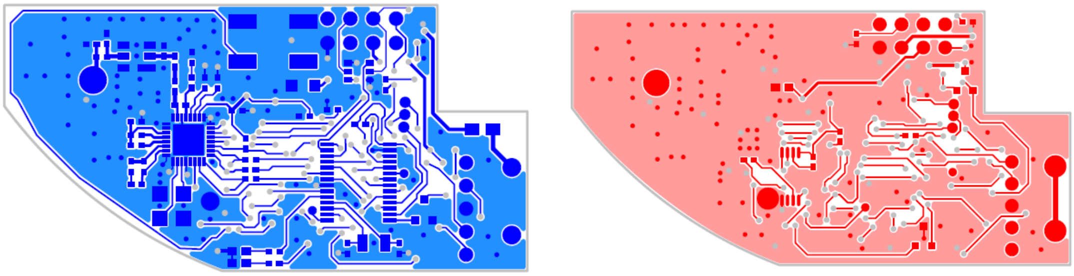
With the system prototype providing expected performance, the next step was to create a custom PCB to enable further development on a mechanically stable and robust platform. The PCB material used was 2-layer FR4 with 1.6 mm thickness. A 4-layer PCB was not necessary as the complexity of the routing was low and the Radio Frequency Integrated Circuit (RFIC)'s manufacturer did not specify the use of a 4-layer PCB. The relatively low total IC count of 10 also meant that an adequate ground plane could be created by using a copper flood connected to ground on the top and bottom of the 2-layer PCB. In terms of cost, 2-layer PCBs are also more economical than 4-layer. The schematics are given in Appendix A.

Mechanical Design Criteria: One of the main mechanical design criteria for the hardware was that it should mate with the smoke/ CO detector housing. The smoke/ CO detector housing provides an 8-pin connector, through these pins the smoke/ CO detector communicates the status of the instantaneous smoke/ CO levels along with alarm signals and optional 9 V power. The physical shape of the cutout in the detector's housing must also be closely adhered to when designing a PCB that is designed to be slotted into the housing. The PCB should provide drill holes where a lithium 3 V battery can easily be soldered and desoldered.

PCB Screenshots: To create a solid ground plane on the top and bottom of the PCB, copper is flooded onto the normally open spaces on the top and the bottom of the PCB. This is shown in Figures 2.6a and 2.6b.

Design for Production Scale Assembly: To facilitate the use of a 'pick and place' assembly machine, the PCBs layout was panelised before being sent for manufacturing. Panelising involves placing multiple copies of a design onto a large panel. Individual PCBs on the panel are connected to other PCBs and the

[7] http://www.fdk.co.jp/battery/lithium/battery_cr238l.html



(a) **Snapshot of the PCB's top layer after copper flooding**

(b) **Snapshot of the PCB's bottom layer after copper flooding**

Figure 2.6: **PCB after Copper Flooding**

outline of the panel via fiducials. These are thin pieces of PCB material which allow for the individual PCBs to be broken away from the larger panel. An image of the final panelised PCB is shown below in Figure 2.7.

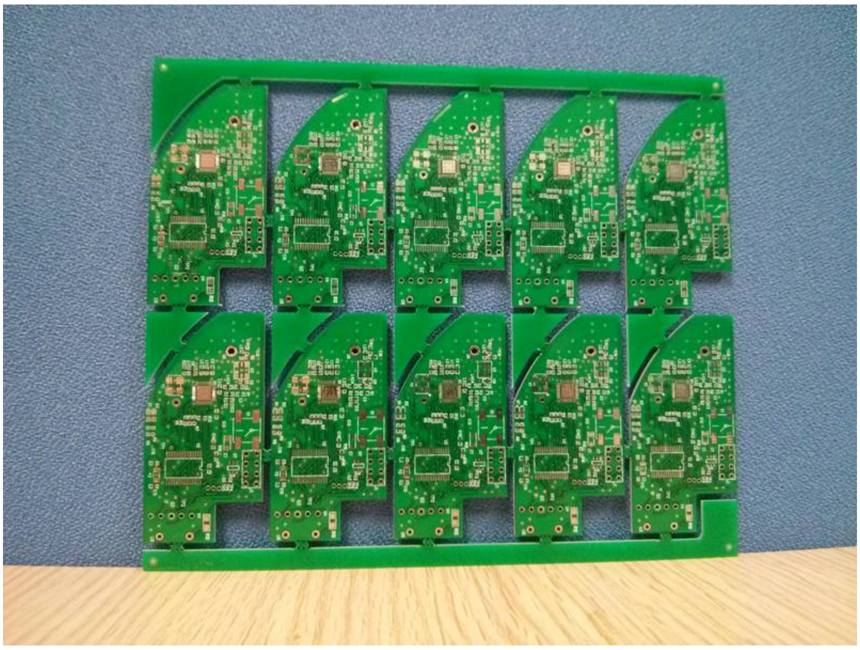


Figure 2.7: **Unpopulated panelised final PCB, 5 x 2 array.**

The final assembled PCB is shown in Figure 2.8.

Figure 2.9 shows the final hardware platform seated in the smoke alarm housing.

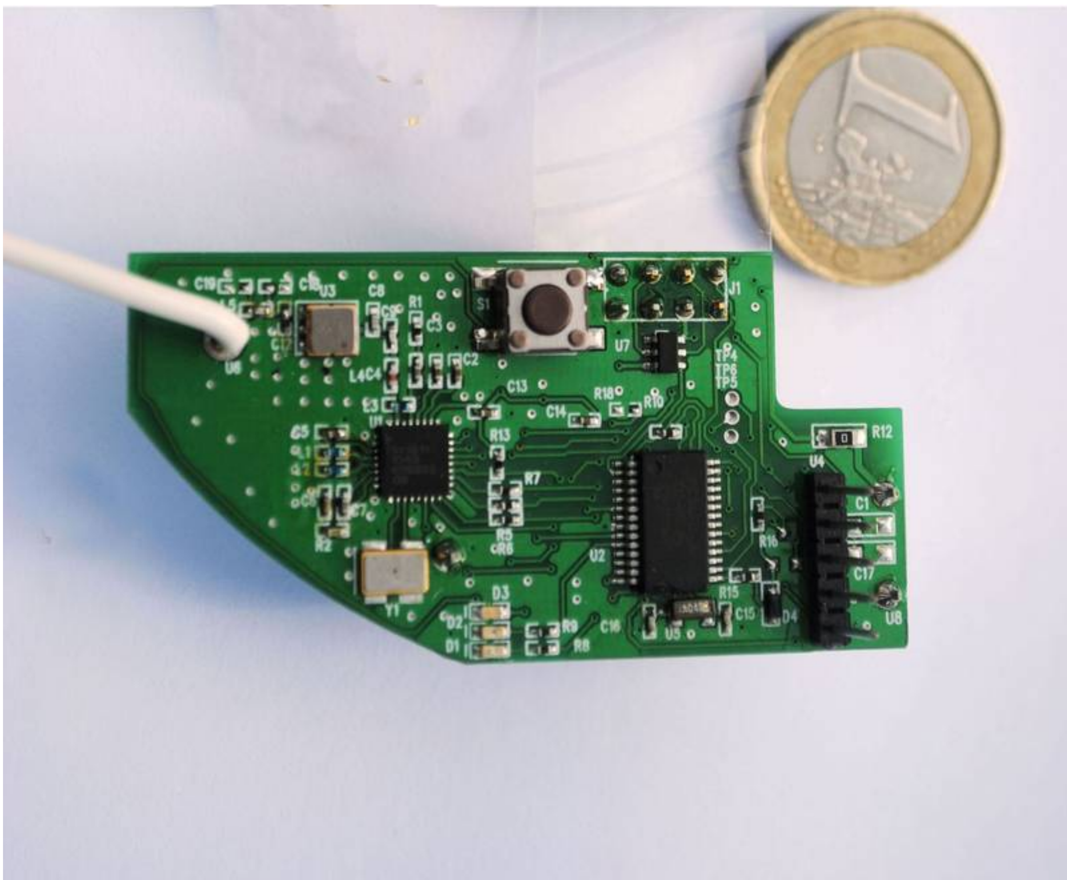


Figure 2.8: **Final Machine Assembled PCB**

Figure 2.9: **FireMote in Alarm Housing**

2.10 Conclusions

This chapter described the design and manufacturing process of an ultra low-power consumption WSN mote. It was designed to help meet the performance requirements described in Section 1.5. With low-power and reliable operation in mind, the hardware components were specifically selected to help achieve the desired levels of performance. A radio transceiver operating in the 868 MHz ISM band was chosen to minimise interference sources, such as in band interference in the 2.4 GHz band. The 868 MHz band was also chosen to maximise communication range. However, operation in the 868 MHz license free band brings other complications such as transmit duty-cycle restrictions. The solutions to overcoming these complications are explained in further details in Chapter 3. The platform achieves a very low sleep current drain of 1.1 μA and uses the lowest current consumption transceiver on the market with a receive mode current drain of just 3 mA.

Chapter 3

IX-MAC, An ultra low-power semi synchronous MAC protocol for WSN applications

This chapter presents a hybrid semi-synchronous MAC protocol that achieves low-power, low-latency and high reliability at the same time, improving on the state of the art. In Section 3.1 an introduction is given into how MAC protocols for duty-cycled operation function and the challenges associated with receiver duty-cycled approaches. Section 3.2 introduces the reader to the current state of the art in low-power MAC protocols and highlights the aspects which this work improved upon. Section 3.3 presents the techniques used in this work and explains their operation. To complete this chapter, the contribution of this body of work is presented in Section 3.6.

3.1 MAC Protocols for duty-cycled operation

The MAC protocol layer is part of the Open Systems Interconnection (OSI) model of computer networking. It acts as an intermediary layer between the networking layer and the physical layer. It applies to both wired and wireless networks. The main goal of the MAC layer is to mediate access of senders to a shared communication medium. Other responsibilities of the MAC layer are:

- Sending data frames: The MAC layer is responsible for sending data packets. Depending on the complexity and required level of quality of service, the send process can involve multiple different steps. Transmissions may be acknowledged or non acknowledged, synchronous or asynchronous etc. Transmissions can also be unicast, broadcast or multicast type transmissions.

- Receiving Data Frames: The MAC layer is responsible for receiving data frames. When acknowledge based protocols are used, the MAC layer must send ACK frames. The MAC layer must be able to filter incoming packets based on their destination addresses and reject packets that are not destined for it. It is also responsible for stripping data bytes from incoming packets which upper layers do not require, these can be preamble bytes or error checking bytes added to frames.

- Handle Collision Detection/ Avoidance: Another fundamental function of the MAC protocol is to handle collision avoidance or detection. When 2 or more networked nodes using a shared medium attempt to gain access to the medium at the same instant, contention will occur. The function of the collision detection/ avoidance scheme is to minimise the probability that contention will occur and detect when it does happen.

- Discard Incoming Frames with Errors: Depending on FEC (Forward error correction schemes) being used or not, the MAC protocol will need to either reject incoming data frames which have errors or attempt to recover the original data packet using an appropriate FEC scheme.

Star wireless networks are networks where all nodes must communicate directly with the network sink. In the context of WSN, these networks are thus single hop networks. Single-hop/ star wireless networks suffer from limited range. The maximum transmission range of a star network is determined by the link budget that the radio transceiver offers.

An example of an RF transceiver suitable for WSNs is the CC2520 2.4 GHz transceiver. It has a receive sensitivity of -98 dBm and a maximum transmit power of 5 dBm. This equates to a link budget of 103 dB and a typical outdoor line of sight range of 400m [1]. In an indoor environment the range of such a transceiver would be greatly reduced to perhaps 10's of meters. To create a network of CC2520 transceivers which should provide coverage of a large building, multiple nodes would be used to create a mesh network. Two nodes separated by distances which result in received signal strength levels outside the

[1] www.ti.com/lit/ds/symlink/cc2520.pdf

link budget, must use intermediary nodes to communicate with one another. The maximum transmission range between any two nodes may be only 10's of meters, but the mesh network increases the useful coverage of the deployed nodes. The maximum coverage range that the mesh network can provide is no longer limited by the link budget of the radio transceivers in use, but by the density and maximum number of hops allowed in the network.

In such a mesh wireless network, all nodes must be capable of receiving and forwarding data packets from neighbours. The simplest and perhaps least energy efficient approach is for all nodes in the mesh network to remain in a mode where they can always receive data packets. When networked nodes remain in an always listening state, the following traits can be attributed to communication between nodes:

- Low Packet Latency: Because nodes are in an always listening state, the latency of transmitted packets over single hops is determined primarily by the bit-rate of the physical layer
- Low Energy per Transmitted Packet: Acknowledge based MAC protocols expect to receive an ACK packet from the destination after transmitting a packet. Because the packet recipient is always listening, the sender generally receives an immediate acknowledgment after its first transmission attempt. The end result is low energy per transmitted packet and low TX duty-cycle due to the short time spent in TX mode per packet.
- Increased Reliability: Again, due to the fact that all nodes in the network are always listening, sending nodes can easily re-send packets in the case that an acknowledge packet is not received. Having the ability to easily re-send unacknowledged packets at a low energy cost per transmission, increases reliability dramatically. Senders can afford to try multiple times without consuming excessive energy per transmission attempt.

The baseline minimum quiescent current consumption of an always listening type WSN node is approximately equal to the current drain of the RF transceiver in receive mode. If the network demands the lowest implementable latency possible and power consumption is not an issue, the network designer should choose an always listening style approach. When specific WSN deployments require nodes to operate from energy constrained power sources and latency requirements are not as stringent, a duty-cycled approach should be employed.

Duty-cycling refers to moving from an always listening approach to a system that switches between a low-power non-listening state and a listening state. This process of receiver duty cycling is shown in Figure 3.1, where the duty-cycled approach listens for T_{RC} ms every T_W seconds. From Figure 3.1,

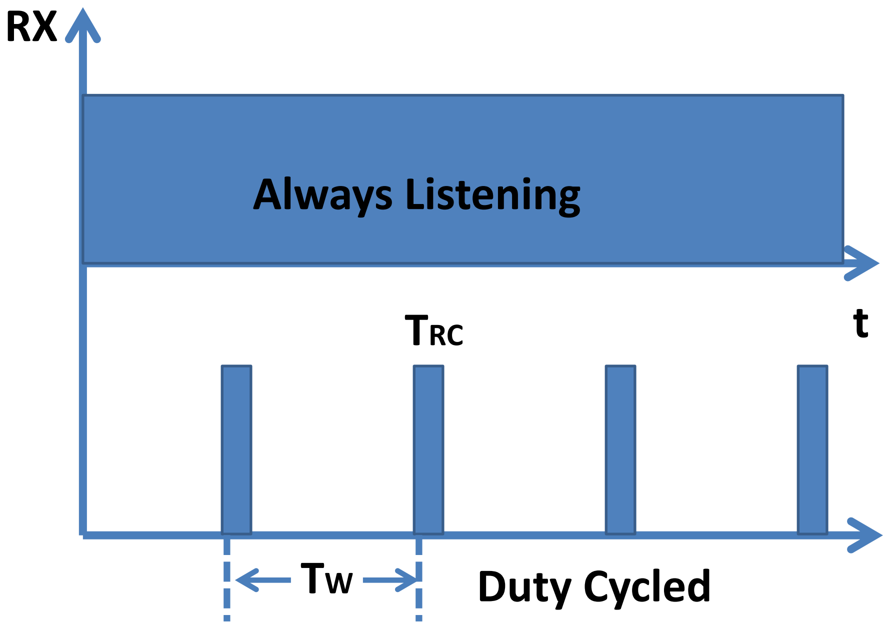


Figure 3.1: **Always Listening approach vs. Duty Cycled approach**

the interval between periodic receive events is defined as T_W. Each event where the receiver is enabled is called a receive check. The length of time the receiver remains active during each receive check is T_{RC}. In terms of the contribution which the time spent in RX mode has on the overall power consumption of a wireless node, the duty-cycled approach offers large savings. The savings can be approximated by the following formula:

$$Saving \approx \frac{T_{\mathrm{W}}}{T_{\mathrm{RC}}}$$

If T_{RC} is 10 ms in length and T_W is 1s, this would equate to a reduction of time spent in RX mode by a factor of 100. Converting to a reduction in power consumption, the power consumption of the node in power-down and receive mode must be factored in. The reduction in power consumption can be estimated by:

$$Reduction \approx \frac{P_{RX} \times (T_W)}{(P_{RX} \times T_{RC}) + (P_{Sleep} \times (T_W - T_{RC}))}$$

If power consumption in RX (P_{RX}) is 60 mW, P_{Sleep}=10 μW, T_{RC} = 10 ms and T_W=1 s. The reduction in power consumption would be a factor of 99.344, or a power consumption of 604 μW compared to 60 mW.

The above example emphasizes the reduction in power consumption that can be achieved by using a receiver duty-cycled approach. The previous example only emphasises the impact that periodically listening to the medium for incoming packets has on the overall power consumption budget. This periodic listening is referred to as receive checks. In reality, there are many factors which contribute to the average power consumption of a wireless node in a multi-hop network. However, when it comes to the disadvantages and challenges associated with using duty-cycled approaches, there are unfortunately many. Some of these are listed and summarised below:

- High Energy Per Transmitted Packet: Now that nodes are no longer always listening, senders must continue attempting to contact them until the packet recipient's receive check event occurs
- High Energy Incurred by re-transmission attempts: If a node fails to acknowledge the first transmission attempt the sender must continue attempting to deliver the packet until the recipient's next scheduled receive check, each transmission attempt incurs a full T_W seconds worth of transceiver active energy, shown in Figure 3.2
- High Latency: Packet latency is increased with receiver duty-cycling approaches. Worst case scenario latencies can be T_W seconds per-hop

The energy per transmitted packet is significantly higher when duty-cycled approaches are used. The average amount of time for which the RF transceiver is active during transmissions is equal to:

$$Average\, Time\, per\, Message \approx \frac{T_{\mathrm{W}}}{2}$$

There are two ways that a sender can contact a duty-cycled receiver. When a non-synchronised ACK based MAC protocol is used, the energy per transmitted message is directly proportional to T_W the receive check interval. As T_W the time between receive checks increases, so too does the average energy per

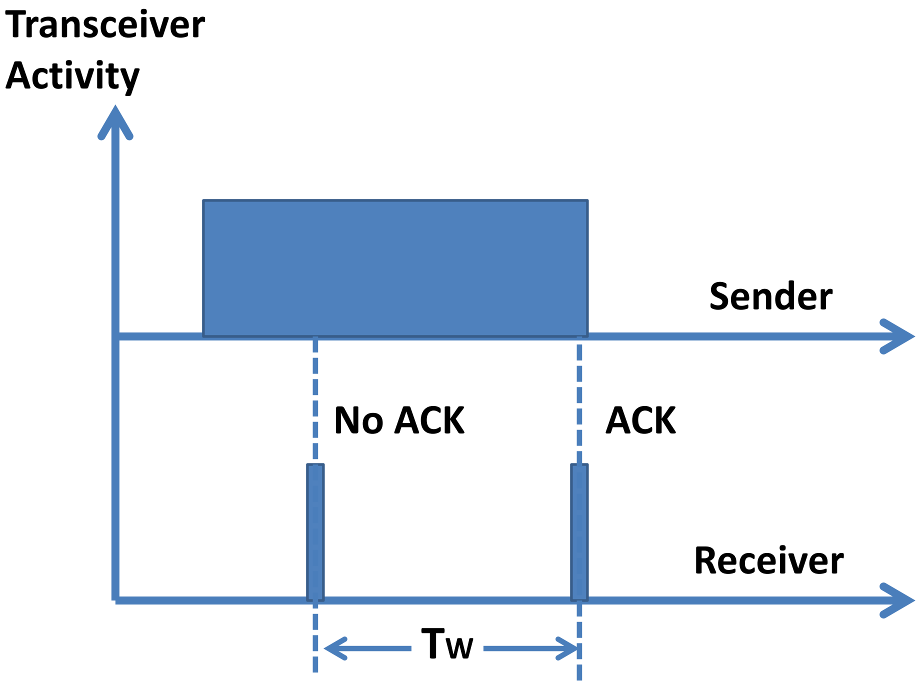


Figure 3.2: **Failure to receive an ACK Energy**

transmitted packet. In the context of this work where reliable communication is a priority, only acknowledge based MAC protocols are considered.

When a synchronous approach is used, the energy per transmitted message can be decoupled from T_W. Synchronous approaches usually have scheduled active phases and will be discussed further in Section 3.2.

In acknowledge based MAC protocols for duty-cycled WSNs, senders interrogate recipients by performing some variation of send-listen for ACK. Senders are aware that packet recipients are duty-cycled and know to perform multiple send-listen for ACK cycles until the recipient responds with an ACK. If the recipient is to be given one chance to decode the message, the sender performs send-listen cycles for one full T_W interval. This is depicted in Figure 3.3.

3.2 MAC Protocols Literature Review

A significant number of MAC protocols suitable for use in low duty-cycle wireless sensor networks have been reported in the literature. This section will highlight the strengths and drawbacks associated with the current art in MAC protocols for low-rate duty-cycled protocols. MAC protocols that are primarily

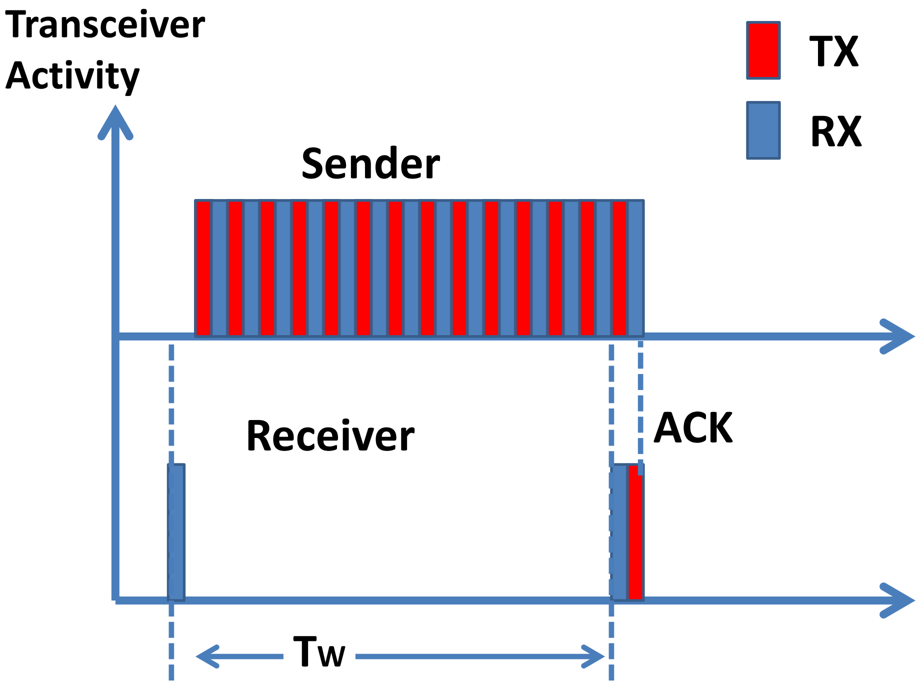


Figure 3.3: **Send Listen stream of duty-cycled MAC**

designed for use in receiver duty-cycled WSNs can be broadly categorised into two types: synchronous and asynchronous.

3.2.1 Asynchronous MAC protocols

The first category of protocols to be examined is asynchronous protocols. No additional time synchronisation is built into these protocols and node's wakeup schedules are asynchronous relative to one another. B-MAC [67] is one of the first low-power MAC protocols designed specifically for use in duty-cycled WSNs. B-MAC works by first transmitting a long, uninterrupted preamble sequence of duration T_W. After having transmitted this long preamble sequence, it assumes that the destination node is awake; at which point the data payload is transmitted. The recipient sends an optional ACK frame. B-MAC must use a layer 1 receive check as the preamble stream does not contain address information. This means any node that is within a one-hop radius of the transmitting node will detect this preamble sequence, and remain awake in receive mode expecting to receive a data payload. If no payload is received, the node will timeout, and return to a low-power mode.

BoX-MAC1 [57] takes this approach one step further, and packetises the preamble sequence by sending multiple copies of the payload which contains all MAC headers with address information. This approach solves some of the over-listening problems associated with B-MAC. Nodes which may overhear these packets by detecting the presence of a carrier will quickly return to a low-power state after determining they were not the intended destination of the packet. A significant shortcoming of the packetising approach of Box-MAC1 is the lack of handshaking. The transmitting node can never know if the data was received correctly at the destination without an ACK.

Both B-MAC and BoX-MAC1 achieve ultra short receive check lengths due to the fact that the sender transmits a constant uninterrupted wakeup signal for the duration of one full T_W interval. Nodes must only sample the medium for a very short period to be able to determine if a transmission is taking place. For physical layers that use FSK modulation techniques, the period for which B-MAC must sample the medium to decide if a carrier is present or not can, in theory, be infinitesimally short. This is because the carrier signal will always be present if a sender is beginning a transmission. In reality, this period will be limited by the specifications of the physical layer itself. Typical RF transceivers require an RX settling/ warm-up time of a few hundreds of μs.

Table 2 in [67] allows us to calculate the energy needed per listen operation. The figure is 23.85μJ per listen operation. Box-MAC1 on the TelosB [66] quotes a figure of 32.26μJ per listen operation with $\mathrm{T_{RC}}$ = 780 μs. Including oscillator startup times of almost 1 ms for the transceiver ([2]), the overall energy figure per receive check is more likely to be $\approx$ 50 μJ. While B-MAC and BoX-MAC1 perform well in terms of energy per receive check, they fall down in terms of the energy needed to transmit to duty-cycled neighbors as the transceiver must remain active for the entire duration of the T_W period. This period may be seconds in networks where a very low receiver duty cycle is being used.

BoX-MAC2 [57] improves further on B-MAC and BoX-MAC1 by adding an interruptible send-listen stream. It sends the full payload and waits for an ACK packet to be received. This cycle is carried out until the destination responds. As soon as an ACK is received by the sender, it ceases its send-listen stream and returns to a low-power mode, knowing the payload was received error free. Because the wakeup stream is now an interruptible acknowledge based type, the length of $\mathrm{T_{RC}}$ increases to 5.61 ms. The wake-up stream no longer consists of an always present carrier signal, but one which switches rapidly from TX-RX

[2]http://www.ti.com/lit/ds/symlink/cc2420.pdf

mode. Using the TelosB hardware, this results in high energy per receive check of a minimum of 280 μJ. BoX-MAC2 uses a layer 1 energy based receive check and hence suffers from over-listening and reduced receiver sensitivity.
BoX-MAC2 is also the standard MAC protocol in TinyOS.

X-MAC [8] and C-MAC [52] are examples of acknowledge based protocols which replace the traditional method of sending the full payload/ wait for ACK stream. They transmit multiple RTS (Request to Send) packets with each individual RTS packet followed by a brief period where the node listens for an ACK/CTS (Clear to Send) message from its sleeping neighbour. As soon as an ACK/CTS is received from the destination, the payload is transmitted and optionally acknowledged by the recipient.

C-MAC's RTS packets do not contain any target address information, and therefore rely only on carrier/preamble sense techniques to detect the RTS packet. It is based on anycast, whereby the first neighbouring node to respond with a CTS packet will receive the data payload. Based on the C-MAC paper, it is unclear how unicast transmissions are handled. This uncertainty arises because the first neighbouring node to wake up during the alternate preamble transmission and listen for CTS sequence will receive the data payload. Furthermore, C-MAC provides very low idle listening power consumption because it relies on preamble based sampling techniques. Due to the fact that the preamble sequence is no longer uninterrupted, C-MAC adopts a double preamble sampling technique to address the gaps in preamble sequence, during which the sender listens for an ACK/CTS message.

X-MAC also uses a RTS/CTS stream, but its RTS packets contain source and destination address information. This overcomes the issues associated with C-MAC where no destination address information is included in the RTS packet. This enables X-MAC to support unicast. X-MAC uses a layer 2 receive check mechanism to detect incoming packets during receive checks. X-MAC tries to decode valid packets from the transceiver during each receive check. Upon reception of a correctly addressed RTS packet at the receiver, the receiver responds with a CTS packet. If the sender in turn receives this CTS acknowledge packet, it proceeds by sending the payload data and includes an optional payload ACK. The operation of X-MAC is shown graphically in Figure 3.4

The authors of X-MAC report a 20 ms period for T_{RC}. X-MAC uses a layer 2 based receive check, and hence does not suffer from over-listening or reduced

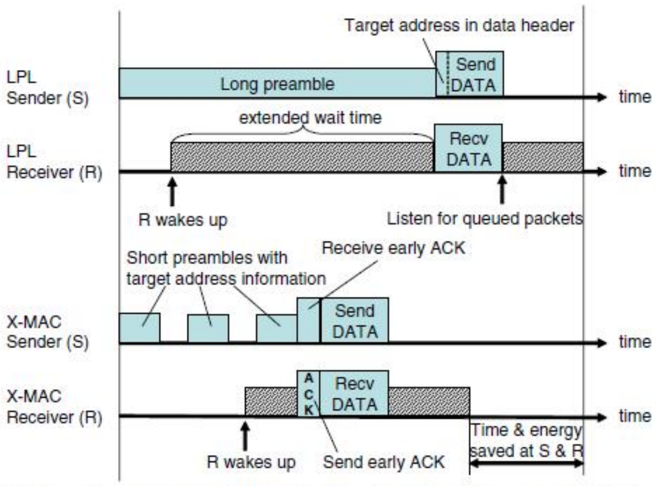


Figure 1. Comparison of the timelines between LPL's extended preamble and X-MAC's short preamble approach.

Figure 3.4: **Taken from [8], shows operation of X-MAC compared to LPL protocol**

receiver sensitivity.

A recent, novel approach is presented by Dutta et al. (2010) in [19]. They proposed A-MAC; a new receiver initiated transmission protocol. Periodic receive checks are replaced by periodic transmissions. Using a standard sender initiated approach, a sender would normally transmit and listen for an ACK continuously, the receiver initiated approach transmits a short packet during each wakeup interval to inform its neighbours it is awake. If a node wishes to transmit data to a neighbour, it simply switches to receive mode, and waits for the destination to transmit its 'I'm awake' packet. As soon as this packet is received, the waiting sender acknowledges this wakeup packet with an ACK. At this point having received an ACK packet, the receiver knows a neighbouring node is waiting to send it a data payload. It remains in receive mode and waits to receive a data payload, the data payload transmission is also acknowledged by the receiver.

During each periodic wakeup, nodes must perform a minimum of 2 or perhaps 3 operations. The 2 operations which must be carried out are to send a short 'I'm Awake' packet and to wait for a potential sender to acknowledge this incoming packet.

A-MAC has the following strengths: It reduces false positives to a minimum. False positives typically happen when a node overhears a nearby transmission taking place and remains in receive mode hoping to receive a packet. A-MAC reduces this by minimizing the time which nodes spend in TX mode and by nodes not performing an energy based listen for ACK after sending its 'I'm Awake' packet during each wakeup interval. It also uses a novel receiver initiated approach.

A-MAC has the following drawbacks: Each wakeup event consumes excessive energy due to the send-listen operation which must be carried out during each wakeup. The authors quote a figure of 263μJ per receive check operation. Due to the fact that each node must transmit a short packet during each wakeup interval, dense networks which wakeup very often could in theory approach 100% channel occupancy just because of these periodic wakeup packets.
A-MAC does not allow an always listening receiver as it would need to constantly transmit probe frames.

ContikiMAC [17] is another example of a low-power asynchronous MAC protocol designed for WSN's for the Contiki operating system. The main goal of the ContikiMAC protocol was to achieve ultra low radio duty-cycle.
ContikiMAC uses a similar approach to BoX-MAC2 to transmit data between receiver duty-cycled nodes. It sends multiple copies of the payload with a period of listening between each payload transmission. The main difference between the two approaches is that ContikiMAC uses a modified receive check that lowers the energy per listen event. Instead of the receiver remaining active for the sender's full quiet period between payload transmissions, it uses a more efficient dual stage receive probe where the transceiver can be put into a lower power mode during the probes. If the sender's listen for acknowledge period between payload transmissions is 2 milliseconds, ContikiMAC performs a single channel probe at at t=0 and t=2 ms. This guarantees detection of the sender's transmission with a low power period between probes.

Depending on the radio transceiver in use, there may only be fractional savings to be made by switching the transceiver to a low power state between probes. This is due to the time taken to transition to lower power modes and the current consumption in these modes. The authors quote a figure of 12 μJ per receive check operation. ContikiMAC also describes a phase-lock optimisation whereby sender's can learn where receivers are expected to wakeup, this reduces the energy required to send packets to receiver duty-cycled nodes.

3.2.2 Synchronous MAC Protocols

This group of low-power MAC protocols use elements of time synchronisation. Some are TDMA based protocols and others are a mix of CSMA and TDMA techniques.

μ-MAC is an example of a TDMA based MAC protocol [5], it was designed for long-lifetime WSN's with the aim of minimising radio duty-cycle. The authors propose a synchronous MAC protocol where all nodes share a common notion of time. The protocol operation alternates between contention and contention free periods. The contention period is used to build network topology and the contention free period is used exchange data between nodes. The contention free period is divided up into time slots to avoid collisions. μ-MAC is capable of dynamically adding additional slots when it detects changes in traffic patterns but the requirement of global time synchronisation adds to the operational overhead of the protocol.

S-MAC [85] is another synchronous example, it requires knowledge of the sleep/wake schedules of neighbouring nodes. S-MAC sends periodic broadcast traffic to inform neighbours of its sleep/wake schedule. Neighbouring nodes that hear this broadcast message are required to adjust their schedules to ensure they too wake up at the same instant as the node that transmitted the synchronisation message. Nodes form virtual clusters where they learn each other's sleep/wake schedule. This scheme clearly provides a very large saving in terms of transmit power (as each node in the virtual cluster knows it's neighbours will be awake listening when it wishes to transmit), as it removes the need to send a long interrupted preamble sequence such as in B-MAC. The drawbacks of such a scheme include: requirements to send frequent SYNC messages that use periodic unnecessary transmissions, and increases in idle listening consumption (as nodes need to wake frequently to keep their sleep/wake schedules synchronised with multiple neighbouring nodes). Scalability is another concern, as it is not stated with how many nodes a node can attempt to synchronise.

SCP-MAC [86] is another example of a MAC protocol designed for duty cycled WSNs. It attempts to synchronise all nodes to a network-wide receive check schedule. The intention is a system where all nodes wake at the same instant. If neighbouring nodes are guaranteed to be awake during the same interval, this results in low energy per transmitted packet because packets can be acknowledged after the first attempt. SCP-MAC requires periodic transmissions to maintain synchronisation between nodes. The authors encounter issues with synchronising all nodes to one global receive check schedule, this is also identified in [50]. The protocol also suffers from high latency per hop, nodes wishing to forward data packets must wait one full receive check interval before they can progress data packets towards the network sink.

Langendoen and Meier evaluated a number of MAC protocols for low-rate low-power WSNs [46]. They identified WiseMAC [20] as the most energy efficient MAC protocol. WiseMAC uses a neighbour schedule learning system to reduce packet send energy to a minimum.

El-Hoiydi and Decotignie proposed WiseMAC, it is a low-power MAC protocol where nodes learn the individual sleep/wake schedules of neighbouring nodes. When two nodes communicate for the first time, they exchange scheduling information in ACK packets. During the next transmission the sender adds a preamble only wakeup tone before the recipient is expected to wake and perform its receive check. The recipient hears this wakeup tone by detecting the presence of a carrier, it then remains in receive mode waiting for a payload to be received. A payload ACK is sent after the payload is received. The operation of WiseMAC is shown in Figure 3.5, where the access point learns when to begin transmitting preamble before the destination is expected to wake.

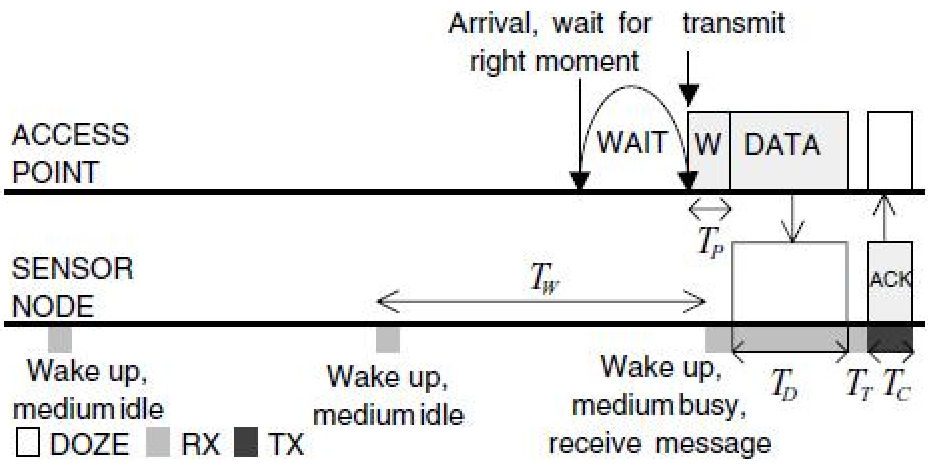


Figure 3.5: **Taken from [20], shows the operation of WiseMAC**

WiseMAC's preamble wakeup tone is smart in that packet recipients can be guaranteed an uninterrupted carrier will be present if a packet is about to be received. This allows for ultra short layer 1 receive check lengths such as those described in B-MAC and BoX-MAC1. The downside again is over-listening and reduced receiver sensitivity. The probability of over-listening is reduced thanks to WiseMAC's short preamble wakeup tones. Depending on the elapsed time between transmissions WiseMAC also adjusts the length of the wakeup preamble tone to overcome oscillator drift.

In WiseMAC, it is stated that the system was only designed for down-link traffic (network sink to sensor nodes). WiseMAC also requires nodes to exchange periodic scheduling information to maintain synchronisation. WiseMAC also cannot perform acknowledged broadcasts. Because nodes perform very short energy based receive checks, a constant carrier must be present for it to detect an incoming transmission. Senders cannot perform a send-listen type stream during broadcasts because the gaps between transmitted packets where the transceiver switches to RX mode could coincide with potential receiver's energy based receive checks. The authors do not present detailed results on real WSN hardware.

LWB (low-power Wireless Bus) [25], developed by Ferrari *et al.* in 2012, is another MAC protocol designed for duty-cycled low-power resource constrained wireless networks. The authors leverage off the ideas of SCP-MAC, all nodes in the network try to perform their periodic receive checks at the same instant. When a single node wishes to send a packet, it waits until it knows all nodes are awake and listening, it then broadcasts the packet. Any node within a one-hop radius of the sender receives this packet and they all broadcast it at the same instant without any CCA check. The aim is for the packet transmissions of all neighbours to occur at the same instant in time, these potentially multiple transmissions will not interfere with one another in a destructive fashion because all packets are identical. Neighbouring nodes will hear and receive these transmissions in phase and in-turn re-broadcast the same packet. This approach can achieve network wide low latency coverage in milliseconds. It does not require any overlying layer 3 networking/ routing protocols.

Each node in the network will broadcast any packet that is generated anywhere else in the network because of the flooding effect. LWB achieves very low latency to the sink in multi-hop networks and the node which generated the packet consumes little energy in transmitting its data.

The drawbacks associated with this approach are the following: If a packet is to be forwarded to the sink over multiple hops within a single receive check event, receive checks must long enough to support this. If a node 10 hops removed

Table 3.1: **Receive Check Energy Comparison**

Protocol	**Type**	**Energy**
B-MAC [67]	L1	17.3μJ
BoX-MAC1	L1	Same as B-MAC
SCP-MAC [86]	L1	150μJ
BoX-MAC2 [57]	L1	348μJ
X-MAC	L2	High
A-MAC [19]	L2	263μJ
WiseMAC	L1	Low
ContikiMAC [17]	L1	12

from the network sink generates a packet of 128 bytes in length (MAX packet length in 802.15.4 and 4.096 ms long), it will incur 4.096 ms of latency over each hop. Nodes within a one-hop radius will receive the packet immediately with only speed of light propagation delays, but nodes further up the tree will need to listen for integers multiples of the max packet length to guarantee reception of the packet. Nodes 2 hops removed need to listen for a minimum of 4.096 ms, nodes 3 hops removed 8.192 ms and so on. Nodes at level 0 (within range of the network sink) will need to listen for a minimum of 40.96 ms to guarantee reception. Listening for this length of time during each receive check results in excessive energy overhead.

Another issue that can occur in LWB is non-constructive interference due to propagation delays. Nodes only a few meters removed from the node which generated the packet will receive the payload with minimal delay, 3.33 nanoseconds per meter. A node 50 meters removed receives the packet 167 nanoseconds later. The underlying bit-rate of 802.15.4 compliant 2.4 GHz transceivers is in fact 8 times the effective throughput of 250 kbps (2 Mbps). This equates to 500 nanoseconds per transmitted bit. When staggered delays occur which are comparable to this underlying bit rate of 2 Mbps, corruption can occur.

3.2.3 Summary

To summarise the shortcomings of the above mentioned protocols are the following:

- Excessive receive check energy: The receive check energy of some of the aforementioned protocols are summarised below in Table 3.1.

- Except for WiseMAC, SCP-MAC and LWB, other protocols have high energy requirements for communicating with duty-cycled neighbours, due to excessive radio on-time. WiseMAC achieves this through neighbour schedule learning with exchange of scheduling information. SCP-MAC and LWB achieve this using network wide receive check scheduling. Without a TDMA or time synchronised approach, the energy required to transmit data packets scales poorly with increasing T_W. Protocols without time synchronisation, typically induce approximately $T_W/2$ seconds of transceiver time per transmission.

- Elements of time synchronisation when present, induced large system over heads or high latency. WiseMAC requires nodes to exchange scheduling information periodically. SCP-MAC introduces approximately T_W seconds for every hop and requires periodic message broadcasts to maintain synchronisation. LWB requires periodic transmissions to achieve good flooding performance.

- Reliability: Except for BoX-MAC, the aforementioned papers fail to give an in-depth analysis of the reliability achieved at the MAC layer. They also fail to give an in-depth analysis of the reliability of the protocol under contention.

Figure 3.6, illustrates the current art in terms of which protocol characteristics they provide and the characteristics which this work aims to provide. Currently there is no protocol which provides ultra low-power operation coupled with low-latency and high-reliability. Figure 3.6 shows approximately into which category each of the studied MAC protocols fall. 1:WiseMAC, 2:LWB, 3:A-MAC, 4:WirelessHART, * = Goal of this work, low-latency, low-power and high reliability.

The MAC protocols reviewed all have their strong points, some excel at providing low latency data and some in low-power operation, but none can provide both. The cumulative shortcomings motivated the design of an improved MAC that leverages techniques from the SoA and combines new optimisations/ modifications to meet the intersecting requirements for energy efficiency, low latency and reliability. Figure 3.6 depicts this concept. None of the current art offers the combined desired performance in terms of power consumption, reliability and low-latency. Utilising a blend of techniques used in previous work and a number of new modifications, the aim was to create a MAC protocol designed specifically for use in duty-cycled WSN deployments. It

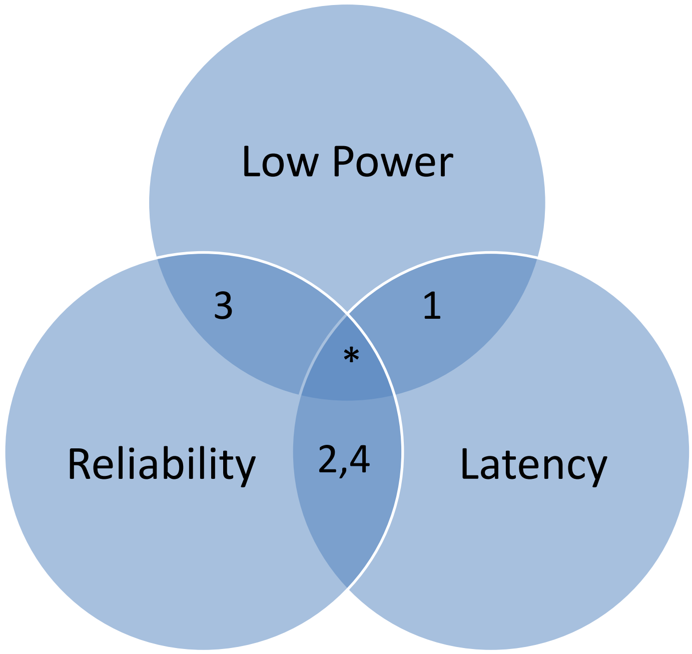


Figure 3.6: **MAC Protocols and their Performance Characteristics**

should meet the requirements specified in Section 1.5 and improve on the performance of the existing art.

3.3 Design of the IX-MAC Protocol

In this section a detailed description will be given of the individual components that make up the MAC protocol which was designed in the context of this work. It is called *IX-MAC* (Improved X-MAC), as it is shares some low-level functionality with the X-MAC protocol. The basic functionality such as unicast and broadcast will be explained along with the multiple power saving and reliability enhancing features. It was designed to meet the initial technical requirements specified in Section 1.5 and was implemented on an 868 MHz and 2.4 GHz transceiver. It aims to provide ultra low-power, low latency and high reliability operation, thus improving upon the previous art in MAC protocols for duty-cycled operation.

3.3.1 Objectives

The objective was to design a protocol to overcome the limitations of existing MAC protocols while optimizing for *energy efficiency* and *reliability*. The primary design goals were as follows:

- Minimise Idle-Listening Power: Idle-listening, be it from periodic medium samples or periodic send-listen operation in the case of A-MAC [19], contributes significantly to a nodes energy budget. From basic calculations which will be shown later in this chapter (Section 3.3), it is shown that improvements could be made by performing these checks more efficiently.
- Develop a simple schedule learning technique to decrease the energy needed to transmit data packets to duty cycled neighbours, hence decoupling transmit energy from the T_W interval. Leveraging some of the ideas presented in WiseMAC, a simpler more robust and more efficient neighbour learning schedule is built which supports multi-hop networks and down-link traffic and up-link traffic. As opposed to WiseMAC which only supports down-link synchronisation. This technique also reduces the TX duty-cycle which helps with regulatory requirements.
- Provide extremely high reliability as the target application demands: The target application demands extremely high reliability as the safety of buildings occupants rely on the protocols reliability.
- Enable 10-year battery life with 2,000 mAh capacity and reasonable sensor reporting intervals (i.e. minutes): The previous art when required to provide frequent low latency sensor readings/ event detection is not able to provide the battery life required by this end application. Taking the example of A-MAC [19], it consumes 253 μJ of energy per receive check, operating on a 1-second receive check interval for low-latency this would give an average current drain of approximately 85 μA or 140 weeks on 2,000 mAh battery. This falls well below the 10-year expectation.
- Support unicast, broadcast and multicast transmission
- Improve packet delivery reliability during contention periods: When multiple senders attempt to access the wireless channel at or close to the same instant, a reliable and efficient CSMA-CD-CA protocol is required. This *IX-MAC* protocol aims to provide 100% packet delivery where channel contention is below 100%.
- Enable acknowledged broadcast transmissions: Acknowledged broadcast transmissions are required when ultra reliable event detection capabilities

must be provided.

The main techniques which were used to meet the design objectives are listed below:

- An energy efficient RTS/ CTS system
- Short efficient layer 2 receive checks
- Neighbour Schedule Learning to reduce energy per transmitted message, without data transfer
- Dynamic TX Power Levels

The main reliability enhancing features are summarised below:

- 2 Stage ACK system with CTS and payload ACK packet
- Efficient CSMA implementation
- Fully Acknowledged Broadcast Transmissions

3.3.2 Receive Check Mechanisms

In a multi-hop WSN, nodes must be capable of acting as routers. For this reason, it is necessary for each routing node to access the wireless medium periodically to check for RF packets for which it may have some responsibility. These receive checks can be performed using 2 possible mechanisms or a combination of both.

Layer 1 receive checks sample only the channel energy during each receive check event. Layer 2 receive checks rely on valid data being decoded by the RF transceiver. Hybrid approaches use a combination of layer 1 and layer 2 techniques, the simplest hybrid approach is to wait for the RF transceiver to receive any data modulated at the correct data rate. An example of this would be RF transceivers that support manchester encoding, if the transceiver detects any valid manchester encoded data at the correct bit-rate, it would notify the microcontroller.

Both of the aforementioned layer 1 and layer 2 techniques have advantages and disadvantages. The advantages of layer 1 based receive checks are the following:

- Short in length.
- More energy efficient.

Table 3.2: **Layer 1 and 2 Receive Check Comparison**

Area	**L1**	**L2**	**Comment**
Length	1	0	L1 Checks are shorter in length
Overhearing	0	1	Zero overhearing with L2
Sensitivity	0	1	L1 places threshold on RSSI level
Selectivity	0	1	L1 wakes for all energy detected

The disadvantages of layer 1 based receive checks are the following:

- Lack of selectivity, nodes will wake on any RF activity.
- Unnecessary overhearing as a result of lack of selectivity.
- Prone to increased power consumption when interference is present.
- Decreased effective receiver sensitivity due to threshold placed on channel energy levels. If the threshold is too low, fluctuations in background noise floor levels will continuously trigger false wakeups.

The advantages of layer 2 based receive checks are the following:

- They utilise the full sensitivity of radio transceiver.
- Zero overhearing, other transmissions will not trigger lengthened wakeups.

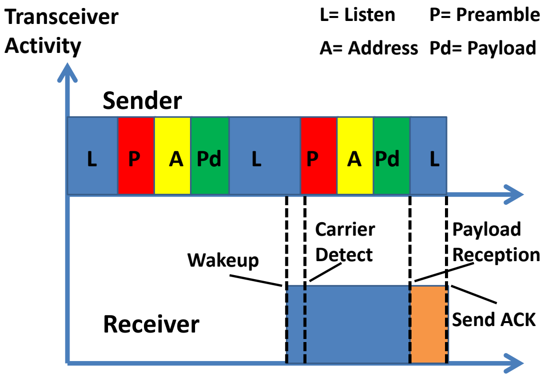
The disadvantages of layer 2 based receive checks are the following:

- Longer in length than layer 1.
- Increased power consumption due to increased length of receive checks.

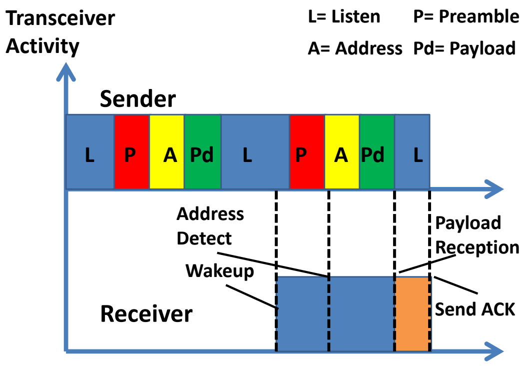
Table 3.2 summarises and compares both techniques. WiseMAC and BoX-MACs [57] are popular protocols that use Layer 1 receive checks. X-MAC [8] uses a Layer 2, packet-based receive check.

Layer 1 receive checks sample only the RSSI during each receive check event. A transmission taking place nearby, which may not be destined for it, will register as being a potentially incoming transmission and cause the node to remain active for longer than its usual receive check length (T_{RC}). High levels of background noise will either cause multiple false wakeups or reduced sensitivity.

Layer 2 receive checks are packet based, where the host microcontroller waits for valid data to arrive at the transceiver. A Layer 2 receive check mechanism exploits the full receive-sensitivity of the physical layer, and outperforms Layer 1 mechanisms in noisy environments [70]. *IX-MAC* thus implements a Layer 2 receive check to eliminate overhearing and maximise link budget.



(a) Layer 1 Receive Check Operation



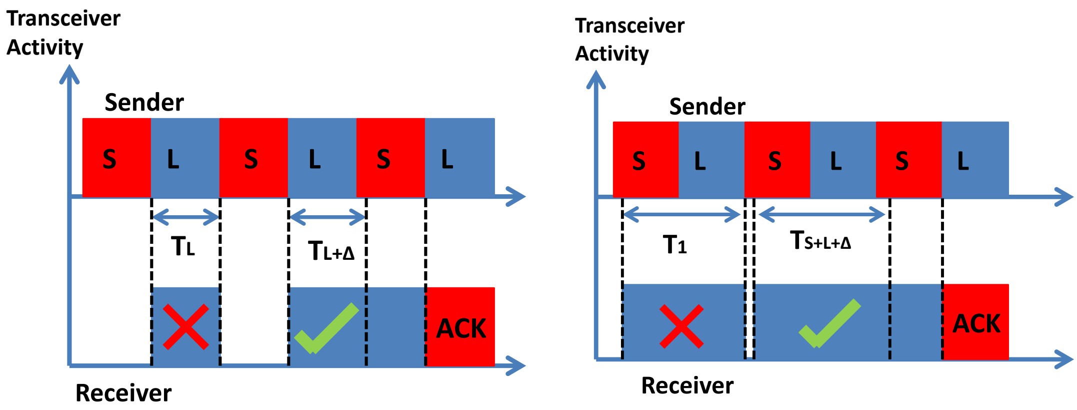
(b) Layer 2 Receive Check Operation

Figure 3.7: **Receive Check Comparison**

Figure 3.7 illustrates how layer 1 and layer 2 receive checks function respectively. It also shows the subtle differences between an energy based (Layer 1) receive check and a packet based one (Layer 2). Figure 3.7a shows an example of a L1 check, the receiver wakes up during sender's listen phase, remains in receive mode and detects the preamble phase of the next send-listen operation of the sender. Figure 3.7b shows the workings of a L2 check, the receiver wakes up during the sender's listen phase, it remains in receive mode and detects the address data of the next send-listen operation of the sender. The sender can use an identical send-listen for ACK stream for both L1 and L2 receive check mechanisms. Immediately after having transmitted the packet, the sender switches to receive mode and listens for an ACK packet, if an ACK is not received within a pre-defined time limit, the packet is re-transmitted.

Figure 3.8 shows the minimum length of both receive check mechanisms when used in a duty-cycled MAC protocol. In the case of layer 1 energy based checks, it depicts the worst case scenario whereby the receiver wakes just after the sender has switched from TX-RX mode. For layer 2 packet based checks, it depicts the worst case scenario whereby the receiver wakes just after the sender has sent the first data bit of the packet.

From Figure 3.8, T_L is the time for which the sender listens for an ACK between transmission attempts. T_S is the length of time spent transmitting the packet. The layer 1 based check must first detect channel energy that is above a threshold and then wait for valid data to be received. The initial receive energy check must be longer than the listen period (T_L) of the sender to guarantee the receiver will detect some carrier signal. The layer 2 based check must be longer



(a) L1 Check Minimum Length

(b) L2 Check Minimum Length

Figure 3.8: **Receive Check Length Comparison**

if it is to detect valid data. If it listens for less than T_{S+L} (the length of one send-listen cycle), it cannot ensure reliable reception of the beginning of the sender's packet.

3.3.2.1 Towards Energy Efficient Layer 2 Receive Checks

When using an ACK-based MAC protocol, radio duty cycled nodes must listen for a minimum length of time to guarantee detection of an incoming packet, this is shown in Figure 3.9. Receive check lengths must reflect the gaps between packets where the sender listens for an ACK packet. A MAC protocol is fundamentally limited by TX/RX turnaround times at the physical layer (i.e. a feature of the RFIC), and the time needed for the destination to send an ACK packet. These restrictions place a theoretical bound on the time which the sender needs to listen for an ACK frame between attempts to send data packets to neighbours.

In systems like BoX-MAC2, the entire payload is sent multiple times with gaps between each transmission where the sender listens for an ACK. If a system like BoX-MAC2 used a Layer 2 receive check, the receiver would need to listen for a minimum time to guarantee reception of a packet. This time is equal to the minimum time required to receive an ACK frame, plus the maximum length of one frame. For the CC2520 2.4 GHz transceiver, this would equate to 4.096 ms for the maximum payload length of 128 bytes, and approximately 600-1000 μs for the listen for ACK time. This gives a required minimum listen-time of approximately 5ms. A minimum listen time of 5 ms would result in a total of

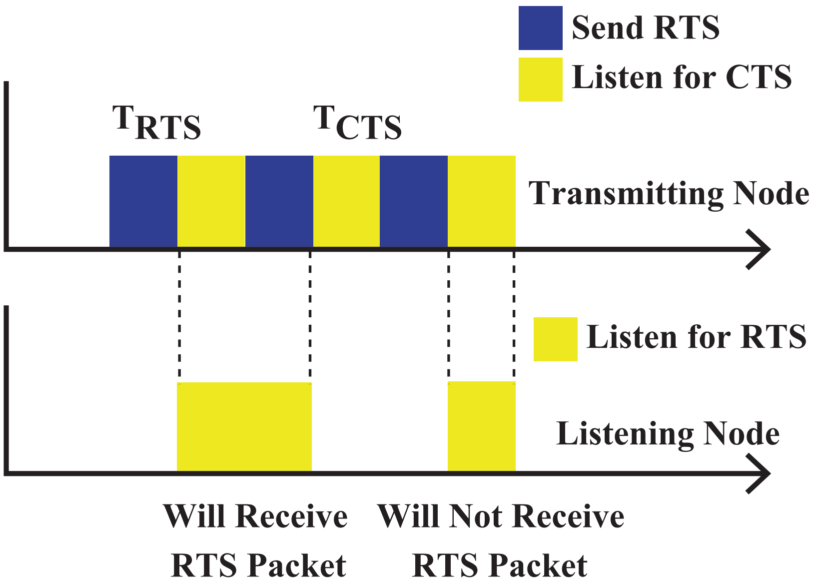


Figure 3.9: **The minimum T_{RC} to guarantee reception of a full RTS packet**

approximately 300 μJ per receive check. This type of system is inefficient because the T_{RC} must be designed to cater for the largest packets. If packets are shorter, the receiver does not need to listen for as much time to guarantee reception of a packet. Thus, the overall energy required is reduced.

In this *IX-MAC* protocol fixed-length wake-up packets are implemented to reduce the energy per receive check. RTS/ CTS which comes from old RS-232 systems was first used in low-power MAC protocols in S-MAC [85]. In *IX-MAC's* RTS/CTS implementation, senders initiate communication with receivers by sending short packets (RTS), followed by waiting for a response (CTS). A node that receives a correctly addressed RTS packet responds with a CTS (containing source and destination address) packet. Upon reception of the CTS, the transmitter sends the payload data and then waits for a payload ACK message. With a fixed length wake-up packet, instead of a variable-length payload, Layer 2 receive checks can be performed in a more energy efficient manner. At 250 kbps, an approximate theoretical minimum bound can be calculated on the listen time per receive check from Table 3.3.

A theoretical minimum listen time for RTS/CTS strobes is 1044μs (see Table 3.3). Assuming a receive current of 20 mA, this equates to receive check energy of approximately 63 μJ per operation. The theoretical figure of 63 μJ per listen operation falls between existing MAC protocols that use Layer 1 and Layer 2 checks.

Table 3.3: **Theoretical RTS/CTS Length Example**

Function	Time
RTS Frame Length	10B 320μs
CTS Frame Length	8B 240μs
TX/RX Turnaround time	192μs
RX/TX Turnaround time	192μs
Microcontroller Overhead	100μs
Total	1044μs

Table 3.4: **RTS Packet Structure**

Bits	0-15	16-31	32-39
Value	Preamble	Sync Word	Length
Bits	40-47	48-55	56-71
Value	Destination	Source	CRC

Table 3.5: **CTS Packet Structure**

Bits	0-15	16-31	32-39	
Value	Preamble	Sync Word	Length	
Bits	40-47	48-55	56-63	64-79
Value	Destination	Source	Reserved	CRC

3.3.2.2 RTS/ CTS Packet Structure

IX-MAC proposes RTS/ CTS strobes with 8-bit addressing (can easily be extended to 16 or 32). Excluding additional error check and preamble bytes added by the RF transceiver, RTS packets are 3 bytes in length, one length byte (necessitated by physical layer), one destination address byte and one source address byte. CTS packets, sent in response to the reception of CTS packets are 4 bytes in length, one for length, one for destination address, one for source address and one additional byte. One send RTS, listen for CTS cycle takes approximately 2 ms using the SX1211 transceiver. The transceiver adds, 2 preamble bytes and a sync word at the beginning of the packet, along with a 2 byte CRC at the end of the packet. Both packet structures are shown in Tables 3.4 and 3.5.

The standard IEEE 802.15.14 ACK frame is 5 bytes in length and does not contain destination or source address information, it does however contain a sequence number byte. *IX-MAC* contains source and destination address information in ACK packets, hence ensuring nodes cannot receive ACK packets destined for other nodes.

Address matching is utilised in the 868 MHz RF transceiver. Packets are received in a two step process, first the receiver waits for an address match interrupt from the transceiver and finally waits for a packet received interrupt. An address match interrupt is generated when the address byte of the packet matches the internally configured address in the RF transceiver.

3.3.3 Learning Neighbours' Schedules

When no explicit synchronisation is used at the MAC layer, the energy required to send a packet between two receiver duty-cycled nodes is directly proportional to the time offset between their wakeup schedules. This energy per packet will vary as oscillator drift causes the time offset between two nodes receive check schedules to drift over time, this is described in further detail in Section 4.4.1. To reduce this energy per transmitted packet, a loose time synchronisation scheme was developed. It does not require explicit exchange of scheduling information and enables ultra low energy per transmitted packet. With restrictions on the TX duty-cycle in the 868 MHz ISM band, this optimisation also enables more frequent transmissions in the network.

This mechanism learns when neighbouring nodes are expected to wakeup to perform periodic receive checks. Knowing when wakeups occur, allows a sender to begin sending just before the destination is expected to wake, thus reducing power consumption through excessive RTS/CTS streaming. Learning schedules involves *zero* additional synchronisation packets. Each node in the network maintains its own wakeup schedule, and guarantees that each receive check will occur at an integer multiple of the network wide receive check rate. Nodes *must* obey this schedule independent of their own tasks. Synchronisation only works when the data send interval is an integer multiple of the receive check interval.

Nodes measure the time offset between their wakeup schedule and those of their neighbours by counting the time taken for them to acknowledge packets. Senders must always begin sending at the same time and receive checks must always be performed at integer multiples of T_W. This guarantees that the time offset between node's wakeup schedules is constant with oscillator drift being the only factor which plays a role. The time-offset value is stored in a neighbour specific data structure, and is updated during every transaction to account for oscillator drift (Section 4.4.1). Before a transmission takes place, the sender recalls the learned time offset value and waits until T_{SYNC} before the destination is expected to wakeup. It then begins its RTS/CTS stream.

Figure 3.10 illustrates this process. It shows Nodes A and B with T_W=1 s, Node B wakes at T=1 s and Node A slightly later at T=1+δ s. τ and τ_1 represent the respective oscillator drifts which occur. Node B learns the offset between itself and A to be δ s, during the next transmission it delays a little less than δ s and begins performing RTS/ CTS before A wakes. The result is greatly reduced transceiver active time per message. This gives reduced power consumption and TX duty cycle.

Consistent scheduling of transmissions and receive checks also enables the estimation of pairwise delays between nodes, caused by radio duty-cycling.

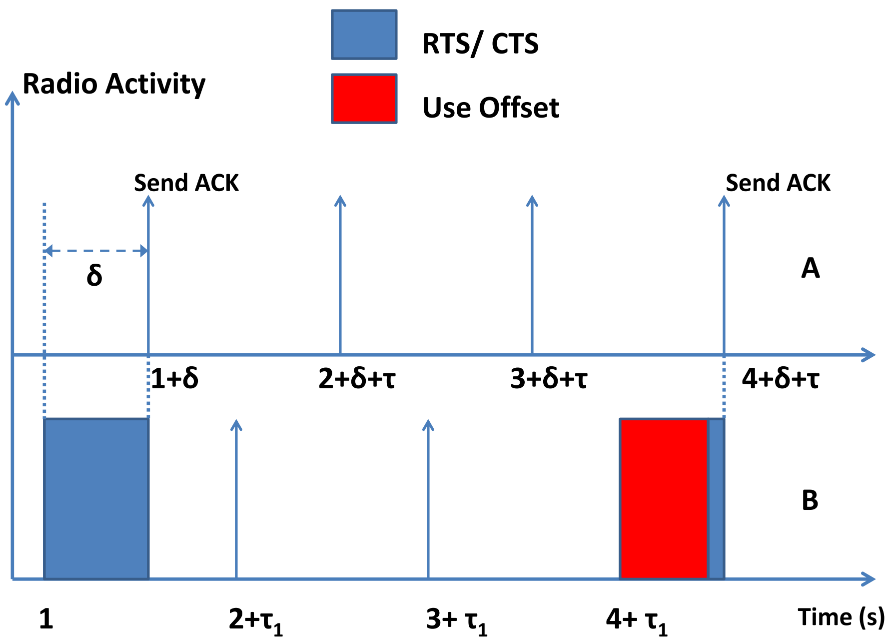


Figure 3.10: **Neighbour Schedule Learning**

Figure 3.11 shows the power consumption profile during a unicast send with neighbour learning enabled and disabled. In the case of neighbour schedule learning being disabled, the unicast send lasts from approximately 6.05 s to 6.6 s. This equates to the radio transceiver being active for approximately 550 ms. In the case of neighbour learning being enabled, the unicast send begins with a receive check at $\approx$ 6.25 s and ends with a short burst of radio-on time for a few milliseconds at 6.6 seconds. Figure 3.11 also shows the scheduled receive checks occurring at 6.05 s for the trace where neighbour learning is disabled and at 6.22 s for the trace where it is enabled.

To calculate the energy required to send a packet in both scenarios, the current consumption is integrated from t=6 to t=6.7 (times taken from 3.11). The equation used is shown in Equation 3.1.

$$Energy(J) = V * \int_{6}^{6.7} i_t \, \mathrm{d}t \tag{3.1}$$

The version without neighbour learning consumes 15.4 mJ for the unicast send. The version with neighbour learning enabled consumes 826 μJ for the unicast send operation. This equates to 18.6 times less energy to send a unicast packet using the version with learning enabled. A supply voltage of 2.8 V was used for these experiments and data was captured using an Agilent N6705B DC Power Analyser.

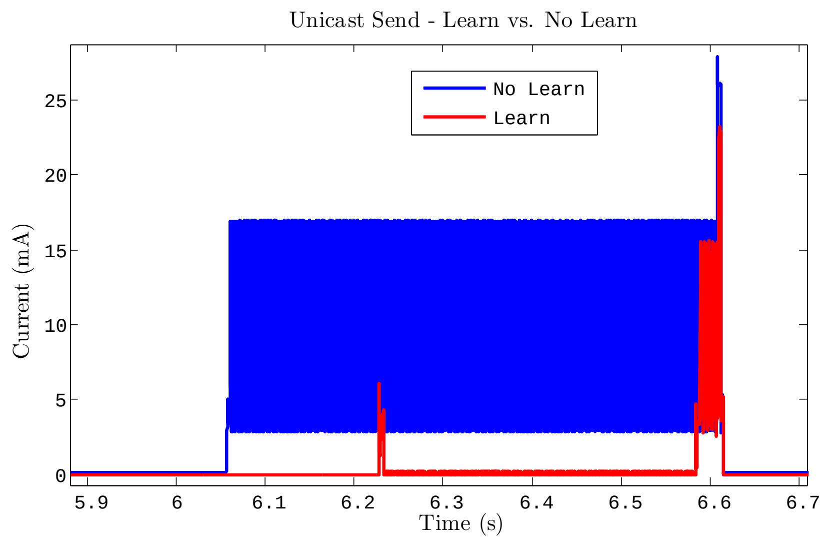


Figure 3.11: **Current Profile with Neighbour Learning enabled and disabled**

3.3.4 CSMA Implementation

The CSMA implementation in *IX-MAC* is designed to minimise the probability that collisions will occur, and maximise throughput. It does not allow multiple senders to occupy the channel during a transmission. Other protocols do however allow multiple senders to occupy the channel while attempting to send data to duty cycled neighbours, such as BoX-MAC2 [57]. However, allowing multiple senders to be able to interleave their send-listen streams has potential drawbacks.

The main issue is that because multiple concurrent senders are allowed to transmit at the same time, a clear channel check must be performed before each sent packet in the send-listen stream. This necessity produces many adverse effects. The first is lengthening the overall send-listen cycle time which inadvertently affects the length of time for which a listener must enable its receiver to guarantee reception/ detection of a packet. This increases idle listening power consumption unnecessarily.

The second and more significant effect is that when contention occurs one of the senders will need to perform a back-off delay. This back-off delay causes the length of one send-listen cycle to change, the amount it changes by will depend on the exact implementation. When the length of sender's send-listen cycle can change dynamically, this has the effect of introducing unreliabilities into the system. Recall from Figure 3.8 that a receiver is expecting a certain maximum send-listen cycle time. When this is surpassed, receivers can completely miss the sender's send-listen stream because the sender is busy performing back-off delays. A receiver which fails to detect a sender's send-listen stream, requires the sender to continue attempting to contact the receiver for another T_W interval before it can deliver an acknowledged message. This impacts the power consumption of the sender, another knock-on effect is the increased probability of contention due to the increased TX on time.

In this work, one clear channel check is performed before the general transmission attempt. Using this technique, each RTS packet in the RTS/ CTS stream can be sent without further clear channel checks. The sender guarantees the channel is clear by sampling the channel for a duration of time that is equal to two RTS/ CTS cycle lengths. If a nearby transmission is taking place, the sender can be sure it will detect the transmission by checking the channel for this length of time. The reason why the clear channel check must last for 2 RTS/ CTS cycles is explained below.

The clear channel check function first samples the wireless channel a fixed number of times (16) over a pre-defined period. These sequential checks are carried out until each check results in a value that is below the pre-defined threshold. If the initial checks result in each reading being below the threshold, the transmission will proceed. If any one of the initial checks result in a carrier being detected, the channel is deemed to be busy. At this point, the waiting sender will begin performing semi-continuous carrier checks, but this time over an interval that is equal to one RTS/ CTS cycle. The reason for using longer checks for initial checks is to allow waiting senders to gain access to the channel faster. This technique prevents new senders from skipping the queue and ensures waiting senders are given priority.

Finally if the channel is found to be clear a final check is performed before the transmission occurs. Considering that multiple nodes can be waiting for the transmission to finish, a system must be implemented to prevent contention from occurring. To achieve this, a randomised delay is performed between the phase when waiting nodes detect the channel is free and the final check.

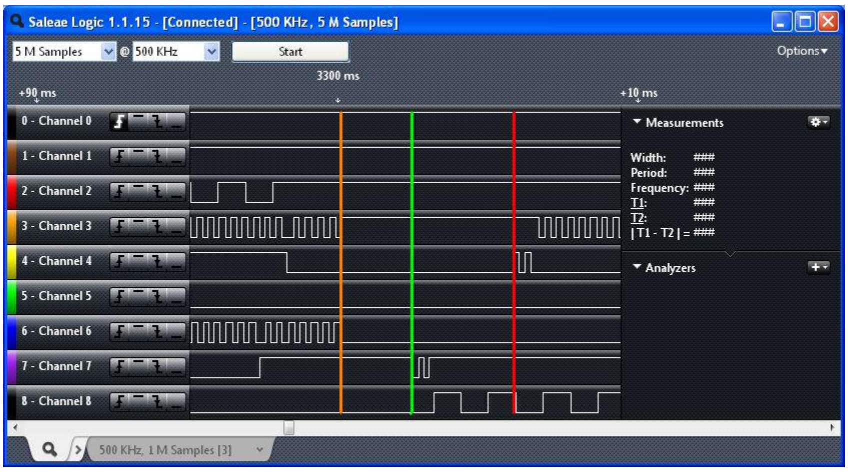


Figure 3.12: **Screenshot of 3 senders competing for channel access.**

To develop and validate the algorithm behind the CSMA implementation used in *IX-MAC*, an experiment with three senders was configured. Each sender tries to gain access to the channel to broadcast a two second long transmission. Figure 3.12 shows a snapshot of the algorithm in action. Three connections are attached to each node, Line 1 toggles when waiting for the channel to become free, Line 2 toggles during the final check after the channel appears to be free and Line 3 toggles when access has been gained to the channel and RTS/ CTS is being performed. From Figure 3.12, Lines 1,2 and 3 for Node A are channels 0 (Black),1 (Brown) and 2 (Red), Lines 1,2 and 3 for Node B are channels 3 (Orange),4 (Yellow)and 5 (Green). Lines 1,2 and 3 for Node C are channels 6 (Blue),7 (Purple) and 8 (Black).

The illustration shows that initially Node A has channel access, because channel 2 is toggling. Nodes B and C are waiting for A's transmission to finish because channel 3 and 6 are toggling. They both detect the channel to be free at the same time where the orange vertical line is shown.The final checks are indicated by channels 4 and 7 toggling briefly (and the green and red vertical lines). Node

C's randomised back-off delay happens to be shorter than B's and when it performs its final check, it finds the channel is clear and C begins transmitting (Green vertical line). Node B finds the channel to be busy during its final check because its randomised delay is longer than C's and C had began transmitting (Red Vertical Line).

This system works for multiple concurrent senders and also includes mechanisms to increase fairness. Waiting senders that have been waiting for long periods are given preference, this is implemented by changing the length of the final randomised back-off delay. This prevents a new sender from jumping in and stealing channel access even though other nodes may have been already waiting to gain access to the channel.

3.3.5 TX Power Optimization

When two nodes attempt to communicate over a large physical distance, this can result in received signal power levels which approach the theoretical link budget of the system. In this scenario, the sender must transmit at maximum output power to guarantee reliable communication. If two nodes are only separated by a few meters, it is possible to reduce the transmit power levels while still remaining well within the transceiver's link budget. This reduction in the TX power level reduces the energy per transmitted packet and also helps to reduce contention by minimising radiated power levels.

Therefore, transmit power levels are scaled dynamically based on the separation between nodes to save energy and reduce contention. RSSI data is exchanged in the MAC header and CTS packet, allowing nodes to know at which RSSI levels their transmissions are being received at. Before a sender begins its RTS/ CTS attempts, transmit power levels are dynamically changed depending on the RSSI level at which the last transmission was received at. If the destination fails to acknowledge a transmission, the transmit power level is returned to maximum at the sender to increase the chance of contacting the destination.

To maintain an absolute reference, the payload is always transmitted at full power. The payload transmission phase of the overall communication cycle is usually much shorter than the contacting phase, hence preserving the concept of reducing contention. Only RTS packets are transmitted at a dynamic level. This maintains a reference point of RSSI, against which both sender and receiver can dynamically scale other TX power levels. TX power levels are dynamically changed in an attempt to keep RSSI levels in the region of -70 to -80 dBm.

In the results Section 4.6 more details are presented on the reduction in power consumption achieved by using this dynamic TX power scaling. This saving decreases as packets are being forwarded less frequently. Lin et al. [51] present a more detailed study into TX power optimisation and a resultant algorithm. The authors in [13], [75] and [38] also present results of MAC protocols which include dynamic TX power adjustments.

3.3.6 Unicast Communication

RTS/CTS strobes are carried out for a maximum of T_W seconds. This gives the recipient one single chance of receiving the packet, i.e. as soon as the destination is contacted, the RTS/CTS stream stops. After a CTS packet has been received from the destination, the sender proceeds by hopping to a new channel and sends the data payload. The channel to which both the sender and receiver hop is based on an XOR operation of their address and the RSSI information communicated in the CTS packet.

The receiver at this point, after sending a CTS packet, hops to the new channel and waits for the data payload. If the receiver receives a data payload within a predefined timeout period, it sends a payload ACK frame to the sender. The sender waits for a predefined timeout interval for a payload ACK frame before resending the payload. The payload ACK frame is 9 bytes in length, it contains length, destination, source and additional bytes for exchanging information that will be required for routing (see Section 5.3.3). Figure 3.13 illustrates the operation of a unicast transmission. Transmitting the payload on a different channel also serves to benefit dense networks where parallel paths exist, and transmissions occur often. RTS/ CTS communication is always carried out on the main network channel. As soon as 2 communicating nodes have passed the RTS/ CTS phase the transmission the main channel becomes free for other nodes to initiate RTS/ CTS communication.

3.3.7 Broadcast and Multicast Communication

IX-MAC is designed to support broadcast and multicast communication. The difference in implementation between broadcast and unicast is that the sender continues to perform RTS/CTS for a full T_W interval to ensure all neighbours

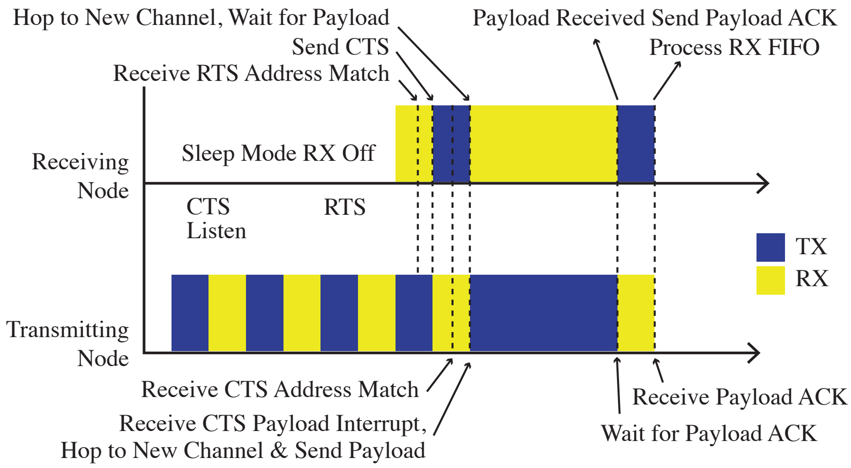


Figure 3.13: **A Unicast transmission between two nodes**

have an opportunity to receive the transmission. Multicast transmissions are also supported. This is achieved by including an extra byte in the MAC header to address specific clusters. The overall length of a broadcast transmission is equal to T_W plus a small additional safety overhead. Broadcast transmissions are also fully acknowledged unlike in Wise-MAC. Figure 3.14 describes graphically the operation of a broadcast transmission, where Node C successfully delivers an acknowledged broadcast message to nodes A and B, the expansion in Figure 3.14 shows how A and B receive C's RTS/ CTS stream. The expanded view shows nodes A and B receiving C's broadcast.

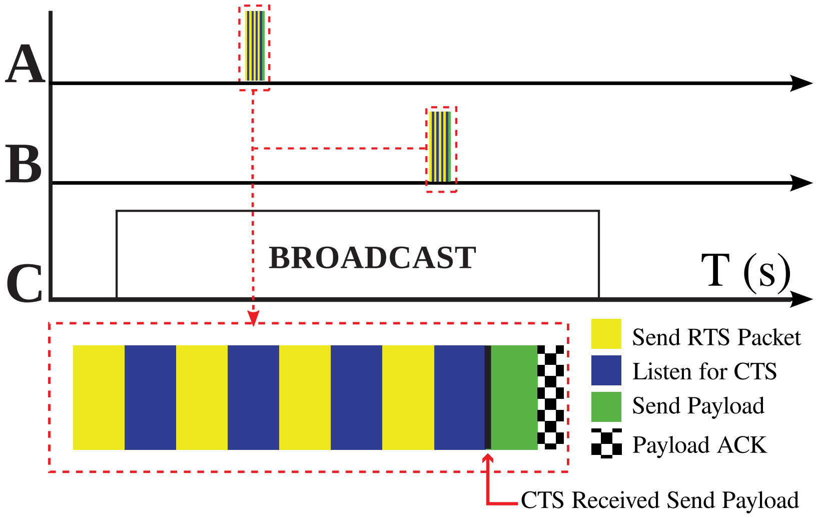
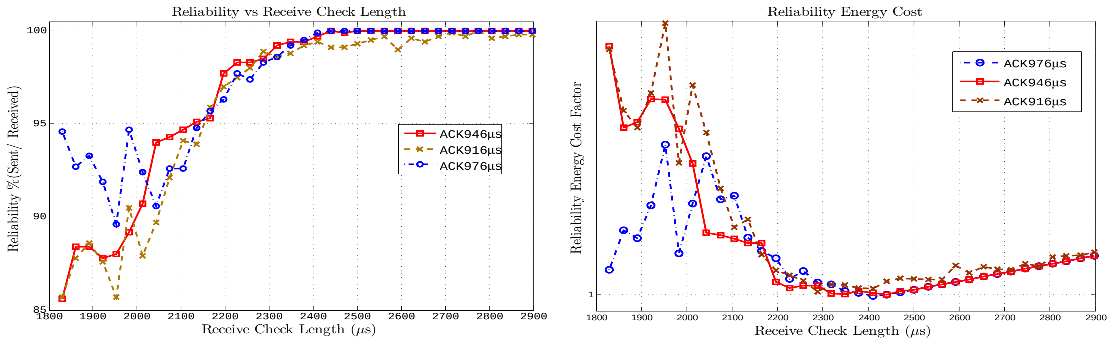


Figure 3.14: **A Broadcast Transmission**



(a) Reliability as a function of listen for ACK period

(b) Reliability Energy Cost Factor Optimisation

Figure 3.15: **Parameter optimization for optimal Energy Efficiency-Reliability trade-off**

3.4 Parameter Optimisation

Considerable testing was performed to optimize the performance of *IX-MAC*. The length of each receive check operation (T_{RC}) is the parameter that most affects reliability performance, also impacting power consumption. If receive check lengths are too long, the protocol will suffer from increased power consumption without increased reliability. If receive check lengths are too short, the protocol will suffer from increased power consumption (more resends) with decreased reliability.

The length of time for which a transmitting node listens for a CTS packet during the wakeup stream ($T_{\text{CTS Listen}}$) also has an impact on energy efficiency and reliability performance. The optimal length of T_{RC} is directly linked to the $T_{\text{CTS Listen}}$ period. Theoretical minimum values for T_{RC} and $T_{\text{CTS Listen}}$ are calculated using RX/TX and TX/RX turnaround times, and bit rate of the transceiver, and microcontroller processing times.

To investigate the optimum values for timing parameters, a two node experiment was used. One node was programmed to wake once per second and acknowledge incoming data packets, the other was programmed to send data packets every second. The receiving node began using a T_{RC} of 1.8ms. Every 10,000 packet transmission attempts, the sending node signals the receiver to increment its T_{RC} by 30μs. It counts the number of acknowledged transmissions. T_{RC} is incremented until 2900μs. The optimum $T_{\text{CTS Listen}}$ period was also investigated in this experiment. Three values (976, 946 and 916μs) were used (derived from multiples of a 32.768kHz sourced timer and derived heuristically). The 10,000 packet experiment was repeated for each of the three $T_{\text{CTS Listen}}$ periods.

To graphically illustrate these effects, a figure of reliability versus T_{RC} is defined. This is the average current that a particular T_{RC} provides, divided by the reliability squared at the sending node. The lower the figure, the better (highest reliability at the lowest energy cost).

Figure 3.15a shows reliability increases with increasing T_{RC}. 976 and 946μs CTS listen versions achieve 100% reliability at T_{RC} above 2450μs. The 916μs version fails to reach 100% reliability consistently, averaging 99.6% above 2450μs. Figure 3.15b shows three datasets on how the Reliability versus Energy Cost figure changes as T_{RC} is increased. When the optimum T_{RC} of 2400μs is exceeded, a decrease in efficiency is evident. These values were used in the final system, the receive check length is 2450 μs and the listen for CTS length is 946μs. These values achieve 100% packet delivery reliability at the lowest energy cost.

3.5 Regulatory Duty-Cycle Restrictions

Considering that the license free bands can be occupied by multiple co-existing wireless networks, transmit power levels and maximum transmit duty cycle restrictions are placed on the users. These sanctions are placed by the ISM band regulatory authorities such as ETSI [3]. Examples of such systems which can be found in residential homes, would be wireless smoke alarm systems and wireless burglar alarm systems. These restrictions are put in place to maximize fairness and reduce the probability of interference/ collisions. Operation in the license free 868˘−868.6 MHz ISM band requires maximum TX duty cycles of <1% (Class 2) and maximum TX power levels of approximately +14dBm. Other nearby bands require <0.1% (Class 1). This specific duty cycle requirement of <1% translates to a maximum of 36 seconds of TX activity within one hour and a maximum length of 3.6 seconds for any single transmission. (See Table 3.6).

These restrictions place limitations on the functionality of the network, specifically, they limit the rate at which sensor readings can be reported to the network sink. Increasing the sensor reporting rate increases the TX duty cycle. The rate at which sensor readings must be reported from each node in the

[3] http://www.etsi.org/

Table 3.6: Frequency Bands For Non-Specific Short Range Devices in Europe

Frequency Band	Max ERP	TX Duty Cycle	Bandwidth
868–868.6MHz	14dBm	$< 1\%$	No Limits
868.7–869.2MHz	14dBm	$< 0.1\%$	No Limits

network, is a largely application based requirement. Depending on the application, these regulatory restrictions may impose a limit on the desired sensor report rate and the network engineer will need to be satisfied with a lower sampling rate.

Using the techniques described in Section 3.3.3, *IX-MAC* increases the functionality of devices operating in networks that must conform to regulatory standards. The neighbour learning technique enables nodes in the network to be able to send sensor readings more frequently, hence improving functionality while adhering to legal regulatory requirements. It lowers the amount of the time the radio transceiver needs to spend in TX mode to deliver a message to a receiver duty-cycled neighbour.

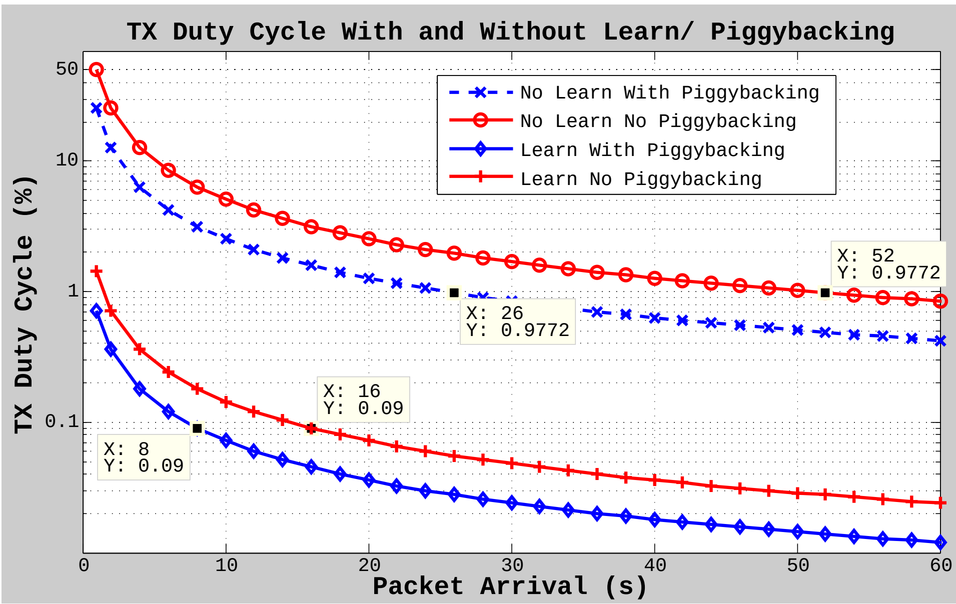


Figure 3.16: **Improvement in TX Duty Cycle to meet legal requirements**

To evaluate the reduction in TX duty-cycle, a small network was modeled to measure the reduction in TX duty cycle that the developed neighbour schedule learning scheme offers. The network consisted of a line-topology of 3 receiver duty-cycled nodes. The TX duty-cycle of node in the center of the line was measured (Node 2), Nodes 2 and 3 were configured to perform sensor readings at varying intervals and send these readings to the network sink (Node 1). Node

2 was able to communicate directly with the network sink, Node 3 was forced to route its packets through Node 2 onto the network sink.

Figure 3.16 shows the results of the simulation, only the blue traces are off interest. The upper blue trace depicts the achieved TX duty cycle with neighbour schedule learning disabled. For all tested packet send intervals (1-60 seconds) it does not achieve a TX duty cycled less than 0.1%. Forwarding packets every 26 seconds it achieves a TX duty cycle just under 1%. The lower blue trace with diamond markers depicts the simulated TX duty-cycle with neighbour schedule learning enabled. The TX duty cycle here is less than 0.1% forwarding packets every 8 seconds, it also achieves sub 1% TX duty-cycle forwarding packets every second.

These results show that using the developed neighbour schedule learning system, nodes can afford to forward one packet per second while remaining below the legal 1% duty-cycle requirement. If a 0.1% legal requirement is imposed, nodes can still forward one packet every 8 seconds while still achieving sub 0.1% TX duty-cycle. Compared to the values attained for neighbour schedule learning disabled, this optimisation offers a vast improvement in functionality of the network while still adhering to legal requirements.

3.6 Conclusion

The novel MAC protocol presented in this chapter focuses on reducing power consumption and increasing transmission reliability. While the underlying hardware provides a low-power radio module, reducing RFIC active time is key to achieving long node life and high network functionality. For applications where sensor readings must be reported frequently, the existing art cannot provide long and reliable network lifetime. *IX-MAC* is described as an optimized MAC protocol that reduces the energy required to reliably transmit data to duty-cycled neighbouring nodes. *IX-MAC* achieves this by efficiently managing the active periods of the transceiver through learning the wakeup schedules of neighbouring nodes. In doing so, it also decouples the energy required to transmit a unicast packet from T_W, the receive check interval. *IX-MAC* integrates a number of existing optimization techniques in a novel way, and delivers significant performance improvements over the existing art.

This work also provides an in depth analysis of timing mechanisms at the MAC layer and their impact on power consumption and reliability. *IX-MAC* lowers the energy requirements of layer 2 receive check compared to the state of the art, this is achieved by a carefully designed RTS/ CTS frame with precise turnaround times. *IX-MAC* provides semi synchronous operation without the need for periodic data transmissions or exchange of information, it is also capable of estimating latency when sending to duty-cycled neighbours. This will be of use to the routing protocol described in Chapter 5.

Chapter 4

IX-MAC Modeling and Comparative Evaluation

This chapter analyses the proposed novel *IX-MAC* protocol through simulation and experimental evaluation. Sections 4.1 and 4.3 investigate the effects that oscillator drift has on latency in asynchronous protocols and Section 4.4 analyses how oscillator drift affects the efficiency of the proposed neighbour schedule learning scheme. Section 4.5 presents a model for the estimated power consumption of the protocol, comparing against WiseMAC and real experimental data from this work. In Section 4.6, results are presented showing a comparison of the performance of this work against a number of other protocols. A brief discussion is given in Section 4.7 and Section 4.8 highlights the contributions of this work.

4.1 Oscillator Drift in duty-cycled WSNs

To maximise the efficiency of the neighbour schedule learning system presented in Section 3.3.3 and to be able to understand the dynamics of latency at the networking layer, this body of work was a necessary step in understanding precisely the mechanisms behind latency drift in duty-cycled multi-hop WSN deployments.

End-to-end latency in a Wireless Sensor Network (WSN) is defined as the time taken for a packet generated by a source node to traverse the network prior to its reception by the network sink. The factors that contribute to non-uniform and unpredictable latency in duty cycled WSN deployments, and the mechanisms behind latency drift (the consequence of which is imprecise synchronisation between devices), remain largely understudied in the literature. Applications of WSNs have varying requirements in terms of maximum and minimum allowable latencies (and are often traffic-type dependent).

An example of a WSN application that places a maximum bound on the allowable latency is a wireless fire alarm monitoring system. EN54 [2] regulates this industry, and places a maximum bound of 30 seconds on the time required for every alarm on the premises to sound their horns under an alarm condition.

In a star topology (i.e. single hop), contention-free WSN, latency tends to be very low. In these conditions, latency is typically determined by the following: the size of the data packet, delays introduced by MAC/CSMA protocol implementation (e.g. guard and back-off times), and the bit rate of the physical layer being used. When a multi-hop topology is used (possibly necessitated by transmission range), the latency incurred by each hop must be considered when estimating the end-to-end latency for each data transmission. In a network wherein each node is in an always-listening state, the delay incurred by each hop contributing to the overall latency remains relatively low. Some factors that contribute to overall latency in such a system are processing time at each hop, bit rate, and network contention.

Increasing the lifetime of WSN deployments is achieved by minimizing the radio on time. This is primarily accomplished by keeping the RFIC in a non-active (usually low-power mode), non-listening, state for the majority of its lifetime. This is known as radio duty cycling (RDC). While RDC can reduce the energy consumption, it has a negative impact on one-hop latency, and, by extension, on end-end latency.

The choice of the network's T_W interval has the largest influence on the per-hop latency, typically dwarfing the impact that processing and transmission times have on the latency. When a non early-exit MAC protocol[1], such as B-MAC or BoX-MAC1 [57], is used, the entire duration of T_W, plus an additional overhead, must be added to calculate the delay incurred over every hop.

On the other hand, when an acknowledgment-based MAC protocol is used[2], such as BoX-MAC2 [57] or X-MAC [8], the average per-hop delay incurred will

[1]Transmission consists of a non-interruptible sequence of packets or preamble
[2]Transmission consists of an interruptible send-listen stream

be equal to T_W divided by two. Thus, the average end-to-end latency, assuming no loss, is approximately the product of the number of hops and half of T_W. The designer of the WSN must carefully choose the value of T_W. The choice of T_W will influence the end-to-end latency and power consumption of each node in the network.

T_W the receive check interval is usually derived from a crystal oscillator source with a given specific accuracy in parts per million (ppm). Nodes used for WSN deployments typically use 32.768 kHz crystal oscillators for timing and from node to node their oscillators will be slightly mismatched and will not run at precisely the same frequency. Therefore, from node to node, timing intervals will unfortunately never be precisely equal to one another. When dealing with ACK based early exit MAC protocols, this accumulated drift between nodes causes the per-hop latency to drift over time and likewise the energy to send a packet between them. Figure 4.1 demonstrates how relative oscillator drift between two nodes causes their individual receive check instances to drift over time. At one instant in time the offset is 1s, during the next measurement this offset has drifted to 0.5s. This time also determines the latency and energy per transmission when sending a packet from node to node. This phenomenon is identical to the beat-frequency principle of two waves interacting with one another. The rate at which two node's receive checks drift is proportional to the difference in frequency of their crystal oscillators.

These concepts are shown in Figure 4.2. Depending on the base frequency of these oscillators, a desired T_W of 1 second can inadvertently become 1 $\pm\Delta\tau$ seconds. Figure 4.2b shows how Node A experiences minimum latency and energy sending to C, but maximum sending to B.

Mathematically, periodic drift can be described as the beat frequency of the two independent oscillators. Let T_1 be the reference period of events for Node 1. Let T_2 be the period of events for Node 2. Consider an instant in time where Nodes 1 and 2 wake at precisely the same instant in time, how can the next time when a rendezvous will occur be calculated?

$$\Delta\tau = |T_1 - T_2| \tag{4.1}$$

$$\Delta\tau = |OSC1_{PPM} - OSC2_{PPM}| \times RCI \tag{4.2}$$

$\Delta\tau$ is the absolute time difference between the two wakeup events, during one

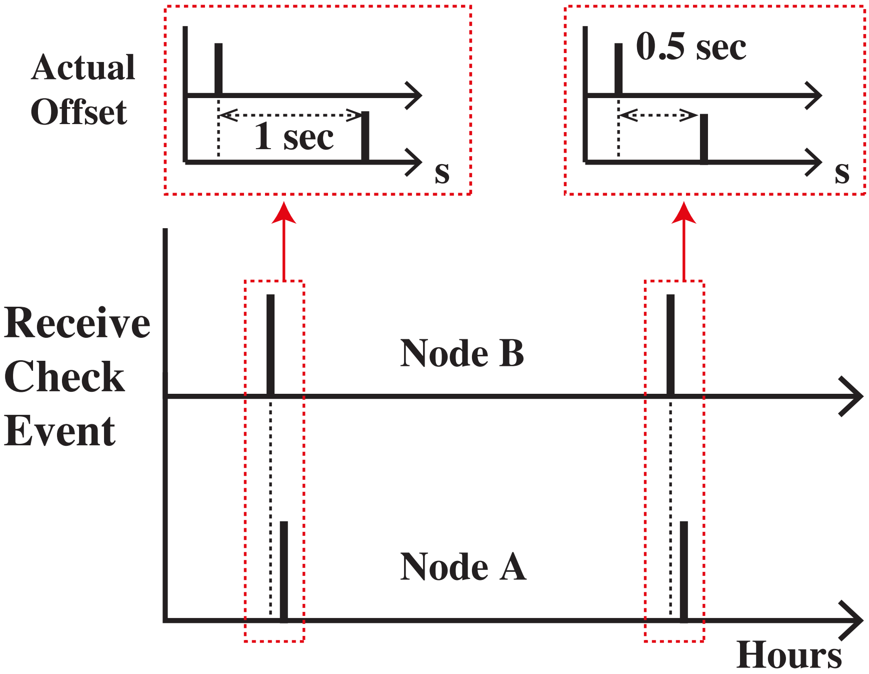
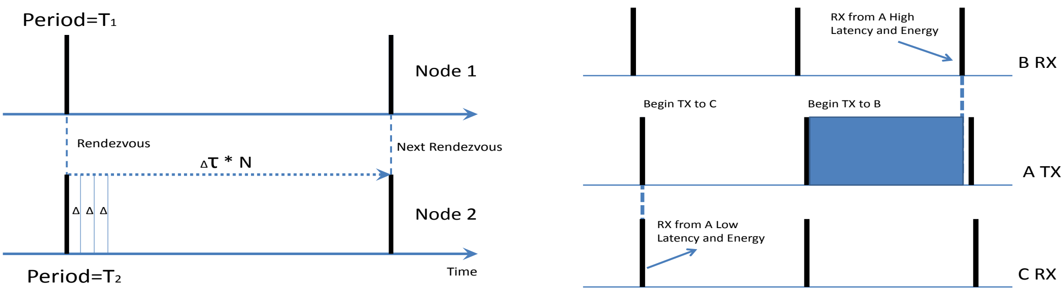


Figure 4.1: **Latency Drift over time**



(a) Oscillator beat frequency, Time for rendezvous to occur

(b) Latency Energy Dependency

Figure 4.2: **Latency and Energy Concepts**

RCI interval.

$$N = \frac{\frac{T_1+T_2}{2}}{\Delta\tau} \tag{4.3}$$

N is the number of wakeup events required before the next rendezvous will take place. During each wakeup event a total of $\Delta\tau$ seconds worth of drift will occur. $\frac{T_1+T_2}{2}$ is the average of both their periods and is the amount of drift required to accumulate before another rendezvous occurs.

$$t = N \times \frac{T_1 + T_2}{2} \tag{4.4}$$

t is the time taken before the next rendezvous will occur. Because $T_1 \approx T_2$, this is replaced by the symbol T.

$$t = N \times T \tag{4.5}$$

$$\therefore t = \frac{T^2}{\Delta\tau} \tag{4.6}$$

Figure 4.2a graphically illustrates the process described mathematically above. Node 2 drifts by $\Delta\tau$ seconds every wakeup interval. It therefore requires N wakeup events before the next rendezvous occurs.

4.2 Optimum T_W for non schedule learning systems

This section focuses on modeling and evaluating how a non-synchronised acknowledge based MAC protocol can minimise power consumption by selecting an optimal T_W interval depending on how frequently packets are being forwarded.

The choice of the network's T_W interval impacts the performance of the WSN in many ways. It impacts significantly the end-to-end latency and power consumption. Power consumption is affected by not only the frequency of receive checks, but also the impact that receive check frequencies have on the energy required to send a packet. In a network where data is sent very infrequently (i.e. hourly or daily), it would appear intuitive to increase the T_W interval to minimise the energy consumed by periodic receive checks. Increasing the interval between periodic receive checks also has its drawbacks, however. Primarily, it has the effect of increasing the energy required to transmit data to neighbouring nodes before they will respond with an acknowledgment (ACK) packet.

When packets must be forwarded more frequently, it would seem rational to reduce T_W, thus reducing the energy required to transmit data packets. Reducing T_W interval has the real effect of increasing the average power consumption. Considering the drawbacks in either case, this motivates an investigation into the optimum T_W interval value for specific packet inter-arrival rates.

4.2.1 System Modeling

The system was modelled to find the optimum T_W for specific packet forward rates. Equation 4.7 to 4.12 below, are used to model the system. The values for the packet inter-arrival time, or data send interval, (T_D), receive check interval and T_W are swept. The equations simulate a generic acknowledge and early exit type MAC protocol. These protocols contact neighbours by sending a stream of packetised preamble pulses. These preamble pulses basically consist of variations of *send packet, listen for ACK* streams. As soon as the destination of a packet wakes and detects the incoming packet, it responds with an ACK packet; causing the sender to cease transmission. Table 4.1 gives all of the constants used for simulation.

The basic factors that influence power consumption when forwarding data packets are presented mathematically in the following. The total amount of energy consumed by periodic receive check is:

$$E_{RCheck}(J) = E_{RC} \times \frac{T_D}{T_W} \tag{4.7}$$

Receiving data packets constitutes sending a CTS, an ACK, receiving an RTS, and a payload. This is described by $E_{RX\text{-}PKT}$:

$$E_{RX-PKT}(J) = V_B[I_{RX}(T_{RTS} + T_P) + I_{TX}(2 \times T_{CTS})] \tag{4.8}$$

V_B is the battery voltage. The next influence on the overall power consumption per data send interval is the energy required to send a data packet to a duty cycled neighbour. Factors that affect this are: CCA check, time spent performing RTS/CTS until node wakes, sending the payload and receiving an ACK. T_{Offset} represents the time required to contact the destination. This is the wakeup offset. The total energy cost for transmission is E_{Send}, given by 4.11:

$$E_{Contacting}(J) = E_{CCA} + (\frac{T_{Offset}}{T_{RTS} + T_{CTS}} \times E_{RTS+CTS}) \tag{4.9}$$

$$E_{Payload}(J) = V_B[(I_{RX} \times T_{CTS}) + (I_{TX} \times T_P)] \tag{4.10}$$

$$E_{Send}(J) = E_{Contacting} + E_{Payload} \tag{4.11}$$

The total energy consumption of the system is thus:

$$E_{Total}(J) = E_{RCheck} + E_{RX-PKT} + E_{Send} \tag{4.12}$$

Table 4.1: **Simulation Constants**

Constant	**Abbr.**	**Value**
Receive Check Energy	E_{RC}	30μJ
Receive Current	I_{RX}	3mA
Transmit Current	I_{TX}	25mA
Receive Check Interval (RCI)	T_W	100ms
Data Send Interval	T_D	20s
Drift	γ	10ppm
CTS Time	T_{CTS}	1.2ms
RTS Time	T_{RTS}	1.2ms
CCA Energy	E_{CCA}	15μJ
Energy RTS/CTS Operation	$E_{RTS/CTS}$	65μJ
Payload Length	T_P	2.4ms
Battery Voltage	V_B	3V

This is the sum of the energy costs required by a repeater node in a network. It receives a packet and forwards it to a duty-cycled neighbour.

The simulation assumes that the sender spends an equal amount of time in both TX and RX modes, trying to contact the destination node using a *send packet, listen for ACK* stream. The time taken for the destination to respond, and thus the energy required to send a packet, varies as the time offset between their respective wakeup schedules changes. The simulation measures the energy required to send data packets across different (T_W). The model includes the energy cost of receiving data packets for repeater nodes in multi-hop networks, although the time taken to receive and acknowledge a data packet (1-2ms) is insignificant compared to the cost of sending data packets to duty cycled neighbours. Sending data packets to duty cycled neighbours can take as long as the entire T_W interval (i.e. seconds).

4.2.2 Results of Optimum T_W Interval Simulation

Figure 4.3 presents one of the results attained from the simulation (The non-smooth characteristic of the plot is a result of a quantised simulation). Therein, the optimum T_W versus packet inter-arrival rate is plotted. The optimum T_W interval, being the time value at which the lowest energy was required to forward data packets at specific packet inter-arrival rates over a given time-frame. As the packet inter-arrival rate decreases, so too should the receive check rate. The exact figures will be MAC protocol dependent, and depend on the length of the receive check operation itself.

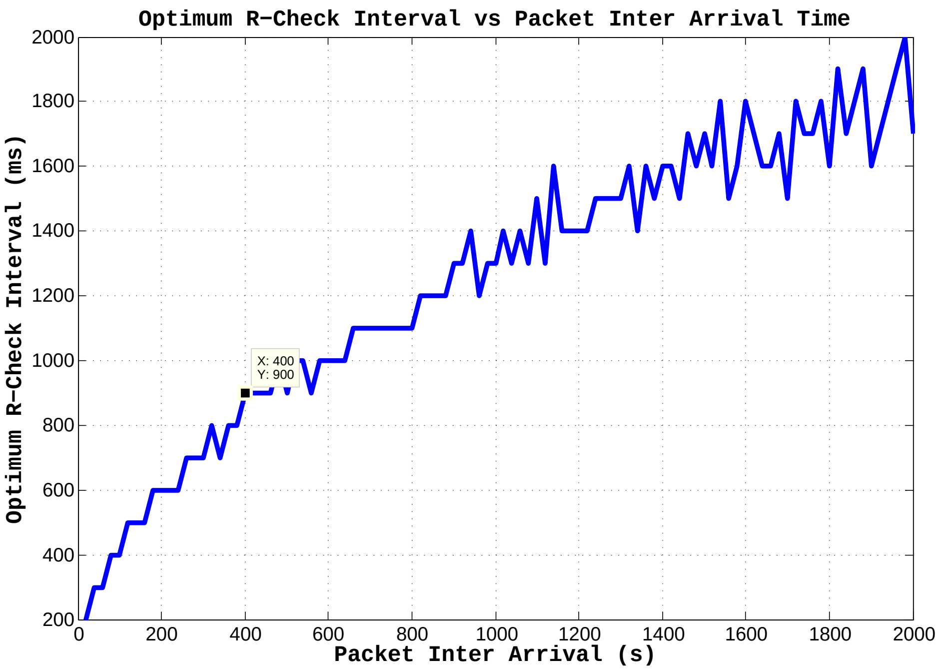


Figure 4.3: **Optimal RCI (T_W) graphed against Packet inter arrival time**

Figure 4.4 shows a specific example where data packets are forwarded at two specific intervals (5 minutes and 40 minutes). The energy required to forward data packets at these rates for a total of 116 days across different T_Ws is graphed. The energy values produced for each T_W are normalised against the lowest energy that was achieved at a specific T_W. Forwarding packets every 5 minutes, the most efficient T_W, that equates to a normalised energy of 1, occurs at 700ms. At 700ms, the lowest energy was required to forward data packets at this rate. This value was used to normalise all of the remaining energy values recorded. The simulated power consumption values were normalised to emphasise the savings which can be made when the correct T_W is used. Also included in Figure 4.4 is empirical test data, which demonstrates the accuracy of the developed model.

For a packet inter-arrival time of 5 minutes, reducing the RCI to under 700 ms results in an increased overall power consumption. This is due to an unnecessary increase in the overall radio duty cycle, caused by over-activity on the part of periodic receive checks. Increasing the RCI above 700 ms also results in increased power consumption. For example, when the RCI is 2 seconds, an increase of 58% is observed in the power consumption. Forwarding data packets every 40 minutes, the most efficient RCI is 2000ms. For this packet inter-arrival

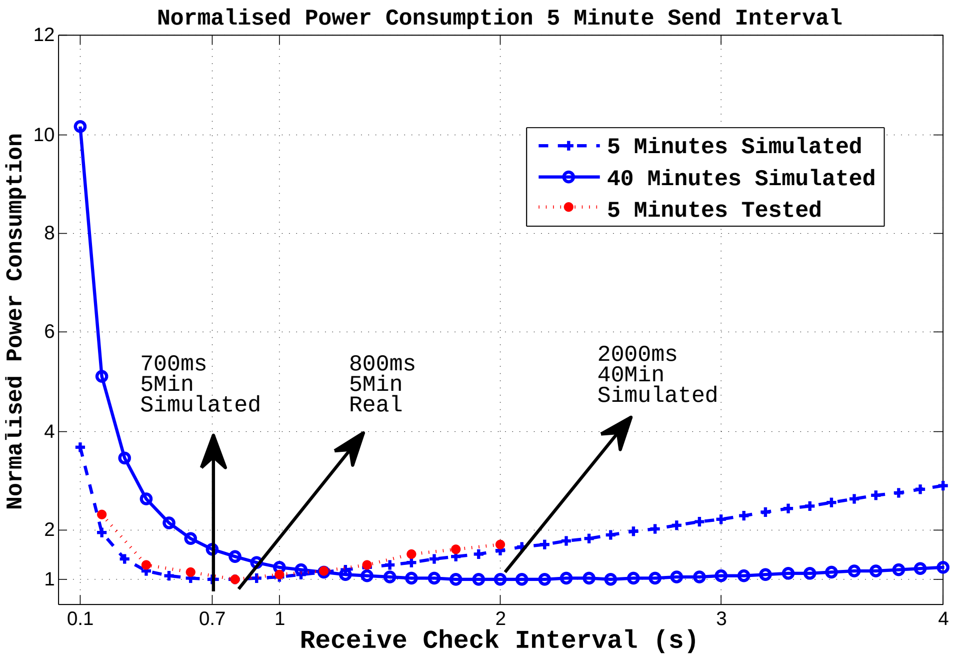


Figure 4.4: **Optimal RCI (T_W) tested and simulated curves**

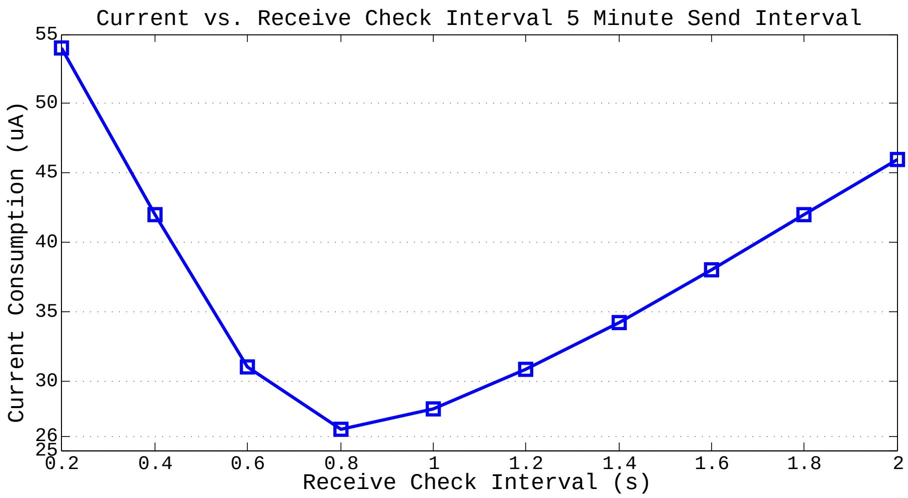


Figure 4.5: **Optimum RCI (T_W) empirical test data**

rate, reducing the RCI under 2000 ms results in an increased power consumption.

In Figure 4.5, real test data is graphed. Here an identical network was implemented and operated for a total of 4 hours. A drift configuration of 190ppm was chosen for the sending node for this experiment. This was done to create a very short drift cycle of 1.46 hours to average the power consumption. If the drift cycle had been longer than the duration the experiment, this would result in final power consumption figure which is not a true representation of the average power consumption over time. The power consumption of the forwarding node was measured using an Agilent N6705B DC power analyzer, with a data send interval of 5 minutes. The RCI was swept as granularly as possible (200ms steps, 0.2s-2s) to facilitate a reasonably time-efficient experiment. The lowest quiescent current consumption of 26.5μA was achieved when the RCI was 800ms. This is in close agreement with the simulated values of 700ms (100ms offset). Across the tested range of RCI values, there is close agreement between simulation and hardware implementation.

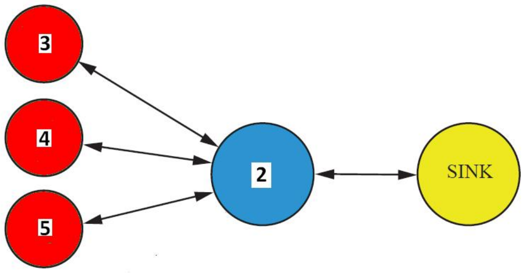
The results from the simulation and real experimental data clearly show that for each specific data send interval there is an optimal value for the RCI (T_W). From Figure 4.4 it can be seen that when data packets are generated and forwarded every 40 minutes, choosing a value for T_W that is greater than the optimum results in a less efficient implementation in terms of power consumption. On the other hand, choosing a value for T_W that is less than the optimum, results in a significantly inefficient implementation (100ms T_W results in ten times higher consumption when T_D is 40 minutes). The slopes of the curves either side of the optimum values are different. In general, choosing a value for T_W that is greater than the optimum, results in a more efficient implementation compared to a value that is lower than the optimum T_W (has less of an impact on the overall radio duty-cycle).

4.3 Pairwise Latency and Latency Drift

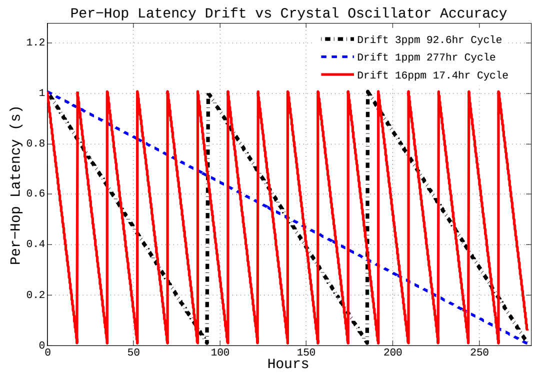
This experiment looked at the latency dynamics of a multi-hop network, namely how pairwise latency changes over time. A simple 5-node network was used wherein one node is programmed as the network sink, and always listens. The remaining 4 nodes are receiver duty-cycled. The second node is located within transmission range of the network sink. The remaining 3 nodes are placed out of range of the sink, but within transmission range of the second node (illustrated in 4.6a). Messages are generated at a fixed rate by the 3 nodes within RF range of the second node. These generated messages are received by the second node, and forwarded to the network sink. Recall, due to oscillator drift differences in the exact rates at which nodes 3, 4, and 5 perform periodic receive checks relative to node 2, this will cause the relative time difference between their respective receive check instances to differ. The 3 sending nodes were populated with 3 different load capacitor configurations to yield oscillator

Table 4.2: **Accuracy Table**

Capacitors	Accuracy
22pF 1%	1ppm
22pF 5%	3ppm
18pF 5%	16ppm
15pF 5%	46ppm



(a) 5 node topology, Latency drift experiment



(b) Per-Hop latency experiment results

Figure 4.6: **Latency Drift Experiment**

accuracies of 1, 3 and 16ppm (Table 4.2).

To calculate the per-hop latency, and the drift in latency over time, the time taken for the packets generated by nodes 3, 4 and 5 to arrive at the network sink was measured. This delay includes the offset between their wakeup schedules and node 2's, and the processing time at 2. Before nodes 3, 4 and 5 begin transmitting their packets to node 2, they signal this by driving an I/O high. Upon arrival at the network sink, after relaying through node 2, the sink also activates an I/O. The time difference between these two pulses is the packet latency, and thus can be measured to calculate the per hop latency. The sink was always listening, therefore removing duty-cycling latency. The time difference between the two pulses (latency) was logged along with a time-stamp. Packets were generated every 5 minutes by each node, and the test operated for a total of 14 days. The RCI (T_W) was configured to be 1 second.

Figure 4.6b shows the results of this experiment. Specifically, it shows how the per-hop latency drifts over time (using 3 different crystal oscillator accuracies). As expected, the maximum per-hop, or pairwise, latency is equal to $T_W + T_{Process}$ (1.008s). The minimum value observed was 8 ms and occurs when the receiving node's receive check coincides with the beginning of the sender's transmission. Also, as predicted, the period of the latency drift cycle is directly related to the crystal oscillator accuracy being used. The higher the accuracy the longer the period of the latency drift cycle. The 1ppm accuracy sender requires 277 hours for the latency drift to complete one full cycle. All 3 experiments yield similar results in terms of the average per-hop latency. The 1ppm sender averages 508 ms of latency. The 3ppm sender averages 508.3 ms of latency and a drift cycle time of 92.6 hours. The 16ppm sender averages 510 ms of latency and a drift cycle time of 17.4 hours.

Statistically, the average per-hop latency is approximately equal to the RCI divided by 2. The accumulated oscillator drift between nodes 3, 4, 5 and 2 causes the time offset between their respective receive checks to drift from 0 to T_W seconds apart, and the latency will change accordingly. This drift process will repeat indefinitely. Over time, this latency can range from the minimum of ($2^* T_{PKT} + T_{Process}$) to the maximum of ($T_W + 2^* T_{PKT} + T_{Process}$). If the crystal oscillators on nodes 2 and 3 both have accuracies of 20ppm (or +-10ppm), then the maximum relative drift between them is equivalent to 20ppm. During 1 second, this equates to a drift of 20μs between them. This is 7.2 ms per hour, or 1.728 s per day. Therefore, if the RCI was set at 1.728 seconds, the latency, data packets experience would vary from the minimum to the maximum over the course of one day.

4.4 Modeling Oscillator Drift's Impact on Neighbour Schedule Learning

The loose time synchronisation scheme used in this work (described in Section 3.3.3), functions by learning the relative time offset between node's wake-up schedules. The instant after a node has learned the relative time offset between its wakeup schedule and that of its neighbour, oscillator drift begins to affect how efficiently the protocol operates. The synchronisation period (T_{sync}) is the length of time a node performs RTS/CTS operations before the destination node is expected to wake-up and perform its receive check. As the time interval between communications becomes larger and larger, the accumulated drift can become comparable to the chosen synchronisation period. At the point when the accumulated drift during a communication interval becomes equal to or greater than the synchronisation period, the energy per transmission increases drastically. Instead of the transceiver remaining active for just a few ms, it must remain active for almost a full T_W seconds.

There are 2 types of drift which can occur between nodes which are time synchronised. A) the relative drift between the node's oscillators causes the receive checks to drift apart, and B) the relative drift between the node's clocks causes the receive checks to drift towards one another. These terms are coined *positive* and *negative relative oscillator drift* (PRD & NRD), respectively. Both of these cases are unique, and both need to be considered when designing an algorithm to counteract oscillator drift.

When two nodes experience *NRD* the effect is the following, let us consider two nodes A and B where node A sends data packets periodically to node B. During node A's first encounter with node B, it learns that node B wakes up Xms later. Gradually, as node A begins sending periodic data packets to node B, the initial Xms offset between their respective receive check schedules becomes X-Yms. Y is proportional to the accuracy of the crystal oscillators being used, and the elapsed time. The same logic applies for the other scenario where the *relative drift* is positive. Negative relative drift causes the sender to need fewer and fewer attempts to contact the destination in terms of RTS/CTS cycles. Positive relative drift when not counteracted, causes the sender to need more and more attempts to contact the host. In Section 4.4.1, it is explained in further detail how both of these scenarios are counteracted.

When data packets are transmitted very infrequently over the wireless channel, large drifts can accumulate between node interactions. Consider a network where a T_{sync} of 20 ms is chosen, T_W of 1s, and a given relative oscillator drift of 20ppm. In the case of *NRD* node's receive checks can drift toward one another at a rate of 20μs/s or 1.2ms/minute, within 20 minutes a relative drift of 24 ms can have accumulated. The sender, having already learned the time offset of its neighbour from a previous encounter, now waits for the normal learned offset time (20ms) and begins attempting to contact the host. In this scenario the sender will miss the destination's receive check and incur almost a full T_W worth of RTS/CTS attempts. A full second worth of RTS/CTS cycles consumes a significant quantity of energy. This occurs because of *NRD*, whereby the receive check drifted from where it was normally supposed to occur.

In the case of *PRD* the normal 20 ms synchronisation period turns into a period of 44ms; 20 minutes after having learned the wakeup offset from the first encounter. This positive drift will accumulate and cause excessive energy per transmission as this figure goes into the hundreds of ms range, if not counteracted against. To investigate these effects, the system was modeled in Matlab to investigate the following system properties:

- Initial synchronisation period (T_{SYNC}) and its effects on power consumption
- Data send interval vs Power Consumption.
- Oscillator drift and its influence on selection of the initial synchronisation period (T_{sync})

The model simulates sending a fixed number of data packets using different synchronisation periods and data send intervals. The simulation also models the effects of *NRD* and *PRD*.

The simulation sends 100,000 packets at each data send interval from 2 seconds to 2048 seconds. The simulation was performed for each synchronisation period from 4 ms to 100 ms. The energy required to send the 100,000 packets across different data send intervals is summated for each synchronisation period. In Figure 4.7, the effects of *NRD* and selection of the synchronisation period on the total energy required to send 100,000 packets are graphed. The energy to send the 100,000 packets across different send intervals is summed for each T_{SYNC}. A summary of the results is given below:

- When data send intervals are large, *NRD* has a large negative impact on the power consumption
- The optimum value for T_{sync}, the synchronisation period, is 24 ms when negative relative drift is occurring and data is being forwarded less than every 34 minutes.
- When the synchronisation period is too low, this results in more periodic re-synchronisations which increase power consumption
- If only *PRD* occurs then the optimal synchronisation period is the lowest one

The T_{sync} interval that gives the lowest total energy to send 100,000 packets across send intervals is used to normalise all of the other summed values for different T_{sync}s. This optimum value occurs at a T_{sync} of 24 ms (depicted in Figure 4.7). Figure 4.7 also shows that when a non optimal T_{sync} period of 10 ms is chosen, the energy required to send 100,000 packets across different data send intervals increases by a factor of 5.5 for relative oscillator drifts of 10ppm and 20ppm.

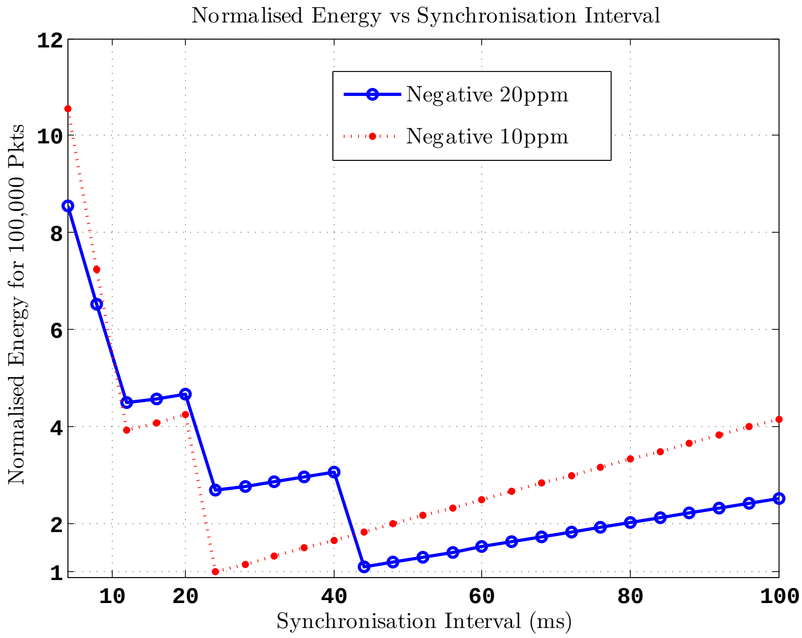


Figure 4.7: **Effect of T_{SYNC} choice on the power consumption**

4.4.1 Counteracting Oscillator Drift

4.4.1.1 Hardware Measures Taken

The first step which was taken to maximise the effectiveness of the loose time synchronisation scheme was to investigate the sources of oscillator drift. The most fundamental factor which affects the oscillator drift is the Q factor of the quartz crystal oscillators in use. The second factor is the value of the oscillator's load capacitors. A number of other factors also affect the accuracy of the oscillator such as temperature (2^{nd} order effect) [72], aging and mechanical vibration (3^{rd} order effects).

Standard surface mount 32.768 kHz oscillators with a high Q factor and an accuracy of 10ppm were chosen. Low tolerance 1% 22 pF load capacitors were used and basic calibration was performed on each node. The gains from using 1% tolerance load capacitors alone was excellent. The following experiment was performed to find the mismatch between the node's oscillators; One node was selected at random and was used as a reference, the relative oscillator drift between the nodes under test and the reference node was measured. The relative drift was measured over a 31 hour period. The graph below in Figure 4.8 shows the results, measured across 20 nodes. The maximum measured drift relative to the reference node was $\pm$ 6 μs.

The measured drift values for each node were documented and the EPROM of each node was programmed with the individual oscillator calibration values. The microcontroller in the Firemote (developed as part of this project in Chapter 2) has an internal Real Time Clock peripheral which contains a calibration register. The peripheral adds or subtracts the value stored in the calibration register, enabling user calibration of the oscillator speed. On startup, nodes read the stored EPROM calibration value and program this value into the Real Time Clock calibration register. This calibration values reduces inter-node relative drift to approximately ±2 ppm.

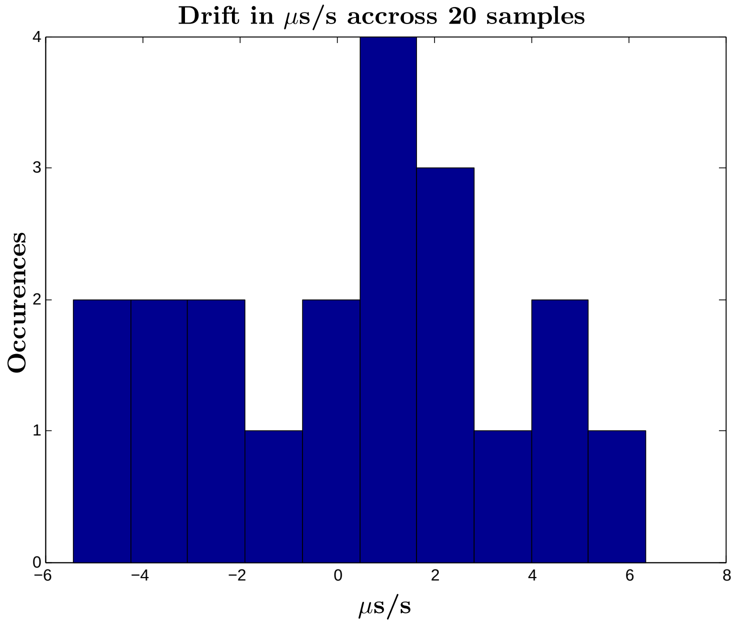


Figure 4.8: **Oscillator drift distribution**

4.4.1.2 Software Measures Taken

After carefully analyzing the results from Section 4.4, it was possible to devise an algorithm in software to minimise the impact oscillator drift has on the effectiveness of the loose time synchronisation scheme proposed. The function of this algorithm is to minimise the number of required resynchronisations to keep two nodes synchronised, and to update the time offset between node's receive check schedules.

The three implemented concepts are:

- T_{sync} is varied depending on how often packets are being forwarded. The amount of drift which can occur in time is directly proportional to time elapsed between transmissions. If a node is forwarding packets to parent every second, it can be sure minimal drift has occurred. In this scenario the sending node needs only 1 RTS/ CTS cycle to contact the destination. As the time between transmissions increases, so too must the T_{sync} interval. Based on the analysis which was done on the oscillator drift, a safety guard band of approximately 1 ms/ minute is added.

- Identify when errors have occurred and re-learns offset. If for whatever reason the number of RTS/ CTS attempts required to contact the destination differs from the number of expected attempts, the sender is forced to re-learn the offset of the destination during the next transmission. This is achieved by setting a bit in the neighbour specific data structure. This mechanism identifies when resynchronisations need to occur.
- Update the time-offset (time to receive ACK) during every communication to account for drift. This counteracts the drift which can occur between transmissions and ensures the sender is able to track the drift and maintain very low energy per transmitted packet.

4.5 Power Consumption Model

To predict the operational behavior of the developed MAC protocol under varying workloads, an analytical model was developed. To estimate the average power consumption mathematically, the operation with the longest period is chosen to average over. In this work, the operation which has the longest period is the packet forwarding period. All contributors to the overall energy consumption which occur during this period must be summed. The operations that impact power consumption are listed below:

- Periodic Receive Checks
- Receiving Data Packets
- Sending Data Packets
- Clear Channel Assessments
- Oscillator Drift

The workload of a wireless node in a mesh network is directly proportional to the number of child nodes which depend on it. For each packet which a node forwards, it must receive and acknowledge an incoming packet and forward this packet to a parent node. This parent node can be a duty-cycled neighbour or an always listening base-station. Both of these scenarios result in different energy quantities per forwarded packet and both are reflected with slightly adjusted mathematical models. To estimate average power consumption under various loads, these elements must be summed.

The energy used by each receive check operation is denoted E_{RCheck}. The total energy per periodic cycle is represented by $E_{TotalRCheck}$:

$$E_{TotalRCheck}(J) = E_{RCheck} \times \frac{T_{SendInterval}}{T_W} \tag{4.13}$$

Receiving and acknowledging data packets (E_{RX}) is calculated below, it also includes a small overhead caused by the microcontroller:

$$E_{RX}(J) = V_{BAT} \times [I_{RX}(T_{RTS} + T_{Payload}) + I_{TX}(T_{CTS} + T_{PayloadACK}) + I_{UC}T_{UCRX}] \tag{4.14}$$

The energy required to forward a data packet to a duty cycled neighbour (whose schedule is known - E_{Send}) is given by the formula below. It also includes the inter-packet drift which causes two node's receive check schedules to drift over time.

$$E_{Send}(J) = E_{CCA} + (2 + (\frac{\gamma \times T_{SendInterval}}{T_{RTS} + T_{CTS}}) \times E_{(RTS+CTS)}) \tag{4.15}$$

$$+V_{BAT}[(I_{RX}(T_{CTS} + T_{PayloadACK}) + (I_{TX} \times T_{Payload}) + I_{UC}T_{UCTX}] \tag{4.16}$$

Periodic re-synchronisations occur between nodes when their oscillators drift one full receive check interval apart. At this point the sender misses the receiver's receive check and is forced to send for one full T_W interval to contact the packet recipient. The energy expended during this operation, E_{RESYNC}, is calculated:

$$E_{RESYNC}(J) = (T_{SendInterval} \times \gamma)(\frac{T_W}{T_{RTS+CTS}} \times E_{RTS+CTS}) \tag{4.17}$$

The total energy per periodic cycle (E_{Total}) is then:

$$E_{Total}(J) = E_{RESYNC} + E_{Send} + E_{RX} + E_{TotalRCheck} \tag{4.18}$$

Table 4.3: **Variable Nomenclature**

$\mathbf{T_{RTS}}$	Time for one RTS Packet
$\mathbf{I_{TX}}$	Current in TX Mode
$\mathbf{T_{CTS}}$	Time for one CTS Packet
$\mathbf{E_{RTS+CTS}}$	Energy for one RTS Send plus CTS Listen
$\mathbf{E_{CCA}}$	Energy for a CCA Check
$\mathbf{V_{Bat}}$	Battery Voltage
γ	Oscillator Drift
$\mathbf{I_{RX}}$	Current in RX Mode
$\mathbf{T_{Payload}}$	Payload Length (ms)
$\mathbf{T_{SendInterval}}$	Packet Send Interval
$\mathbf{E_{RCheck}}$	Energy for one receive check
$\mathbf{P_{SLEEP}}$	Power Consumption in Sleep Mode

Table 4.4: **Model Parameters**

$\mathbf{T_{RTS}}$	1.1 ms	$\mathbf{I_{TX}}$	25mA
$\mathbf{T_{CTS}}$	1.1 ms	$\mathbf{E_{CTS}}$	6.6μJ
$\boldsymbol{T_W}$	0.5s	$\mathbf{E_{RTS}}$	55μJ
$\mathbf{E_{CCA}}$	15μJ	$\mathbf{V_{Bat}}$	2V
$\mathbf{I_{RX}}$	3mA	γ	10ppm
$\mathbf{T_{Payload}}$	2.4ms(30 bytes)	$\mathbf{T_{Send}}$	1-1000s
$\mathbf{E_{RCheck}}$	30μJ	$\mathrm{P_{SLEEP}}$	2μW

The average power, $P_{Average}$, is thus calculated as:

$$P_{Average} = \frac{E_{Total}}{T_{SendInterval}} \tag{4.19}$$

A summary of the parameters used is shown in Table 4.4 and an explanation is given in Table 4.3. This model is based on the SX1211 transceiver. Results of the analytical power consumption model are depicted in Figure 4.9. Here, the power consumption versus packet inter arrival rate is plotted for a single forwarding node.

Langendoen and Meier (2010) [46], identified WiseMAC as the strongest MAC protocol from a number of tested protocols. They modeled a number of MAC protocols and determined WiseMAC to be the best performer in terms of energy efficiency versus latency. To compare the results against the state of the art, the modeled results from WiseMAC are also included in Figure 4.9. Values for WiseMAC are shown in blue, this work simulated is in black and tested is in red. This work outperforms WiseMAC by a factor of 1.85 (48/26) when packets are forwarded frequently due to more efficient RTS/ CTS preamble vs. WiseMAC's TX only wakeup phase.

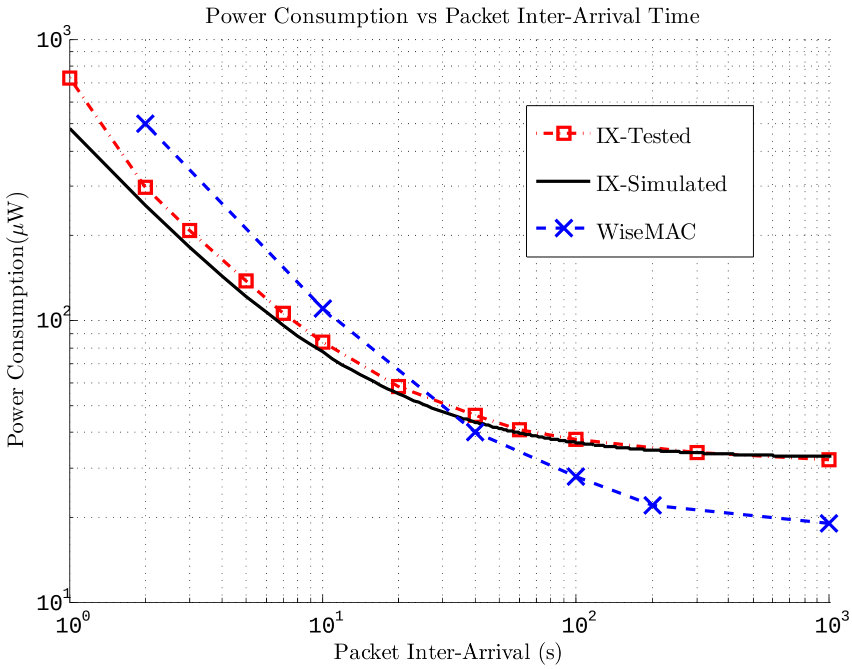


Figure 4.9: **IX-MAC vs. WiseMAC's simulated values**

The theoretical model of this work outperforms WiseMAC's model up until packet inter arrival times of 40 seconds. Thereafter, WiseMAC's extremely low-power physical layer and low energy per receive check operation allow WiseMAC to excel when the packet inter arrival rate is low. Also included in Figure 4.9 is an adjusted model (red trace) which includes the real timing values which were needed to make the protocol work on the hardware testbed[3]. The values modified were values such as E_{RCheck} and $T_{RTS}+T_{CTS}$; the latter two values being adjusted to include microcontroller overheads. Real tested values are also included to show the accuracy of the modeled values. *IX-MAC*'s experimental values show very close correlation with the simulated values achieved from the adjusted model. They also outperform WiseMAC's modeled results for packet inter arrival times of below 15 seconds.

4.6 Comparative Empirical Evaluation

A number of experiments were devised to test and compare the MAC functionality of this work against the current state of the art protocols. The

[3]Timing values were adjusted which included realistic implementation overheads

results of these experiments are given in Sections 4.6.1 and 4.6.2. The 3 protocols which were chosen for experimental comparison were BoX-MAC2, the TinyOS *de facto* standard, ContikiMAC (the standard in Contiki) [17] and X-MAC, also implemented in the Contiki operating system[4]. All of the above mentioned protocols were tested on the Crossbow TelosB platform [1]. Unfortunately with no TinyOS or Contiki version of WiseMAC available, it was not possible to directly compare against WiseMAC. In [37] the authors evaluated WiseMAC on an 868MHz radio and presented some basic results. Using a super capacitor as the power supply for their nodes under test, they estimated the power consumption based on the lifetime of the node.

During the preliminary testing of BoX-MAC2 in TinyOS2.1, it was observed that without using CTP the reliability of communication was very poor (20-30% packets received). To provide reliable communication without CTP activated, ACK_WAIT_DELAY was changed to 58 instead of 256 and MAX_LPL_CCA_CHECKS from 420 to 400.

Testing ContikiMAC, several problems were encountered. The first most fundamental bug was observed when a sender was programmed to send 100 packets to a receiver. At the receiver, these transmissions were counted, often the receiver received more than 100 packets. This was occurring because the sender was failing to receive the ACK packet, and retransmitted the payload. To overcome this, a simple data sequence counter was implemented to reject duplicate packets. The second bug was noticed in ContikiMAC's phase optimization. The problem arose when drift had occurred between two nodes which were phase optimized. Drift had caused the sender to miss the receiver's active cycle. Instead of trying to contact the receiver for T_W seconds, ContikiMAC continues to try a fixed number of times using the learned offset figure; failing every time. Only after a fixed number of failures does it try for the full receive check duration. To overcome this problem, the amount of time a node waits before sending to a phase locked neighbour was reduced and the max number of allowed attempts to contact the neighbour was increased. This modification proved to be less susceptible to drift, and was more reliable.

[4]www.contiki-os.org/

4.6.1 Power Consumption

Not only does this test measure how much energy the node under scrutiny requires to send data packets to a duty cycled neighbour, but it also includes receiving the packets to be forwarded, carrying out CCAs and periodic receive checks. The results of this experiment determine the length of time for which a node could deliver useful sensor readings in a real WSN deployment.

4.6.1.1 Experimental Setup

To carry out a comparative analysis of the power consumption of *IX-MAC* versus the state of the art, an experiment was designed to measure the power consumption versus packet inter arrival rate. T_W=0.5s was used for this experiment for all protocols and nodes. The *IX-MAC* protocol was implemented on two different hardware platforms, the 868 MHz Firemote platform and a 2.4 GHz platform containing a CC2520 transceiver and an MSP430F5437 microcontroller.

A 4 node indoor network was deployed which consisted of 2 senders, 1 routing node and one sink node. The distance between the routing node and the network sink was 20m. The 2 senders were configured to send at specific rates such that when both are sending, the overall arrival rate at the routing node was the desired one. The routing node was programmed to perform periodic receive checks and forward any received data packets to the network sink (which was also duty cycled to execute the same T_W). The routing node was connected to an Agilent N6705B DC Power Analyser. This DC power analyser samples/logs the current which the node consumes at a rate of 5 kHz. The fastest packet inter arrival rate at which the node under test was forwarding packets was 1 pkt/s, this rate was decreased in regular steps until the node under test was forwarding data packets at a rate of 0.0002pkt/s or one packet every 5000s. Data was logged for two hours per test. The results of this experiment are depicted in Figure 4.10 and summarised in Table 4.5.

4.6.1.2 Results

From Table 4.5 and Figure 4.10, it can be seen that all 3 versions of *IX-MAC* outperform the others across all data send intervals. This is due to the reduction in radio duty cycle provided by *IX-MAC*. Using similar hardware to

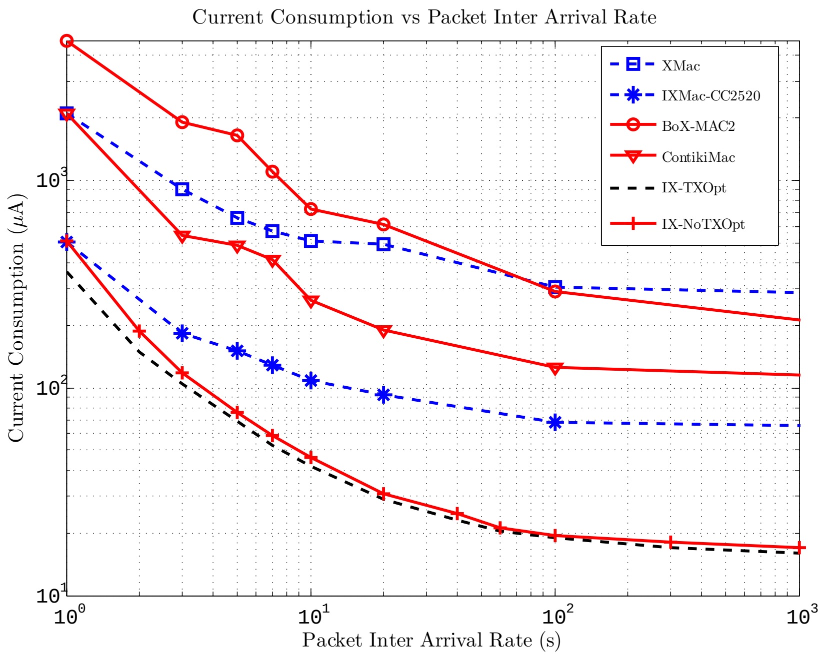


Figure 4.10: **Measured Current Consumption vs. Packet Inter Arrival Rate**

Table 4.5: **Current Consumption results T_W=0.5s, values shown are in μA. 2nd column represents forwarding every 1s, 3rd every 10s etc.**

Protocol	1s	10s	600s
IX868-TXOpt	317	58	29
IX868-NoTXOpt	502	70	29
IX-2.4Ghz	504	102	65
BoX-MAC2	4700	720	220
Contiki	2100	264	115
X-MAC	2100	512	291

the TelosB with the CC2520 2.4 GHz platform, the 2.4 GHz implementation outperforms the state of the art. Figure 4.10 also proves that *IX-MAC's* TX power optimisation is a secondary effect and only provides savings when packets are being forwarded frequently. As packet inter-arrival rate decreases, the time the radio spends sending data to duty cycled neighbours has increasingly less of an impact on the overall radio duty cycle of the system. Reducing the radio active time during transmissions with duty cycled neighbours is the key to reducing power consumption when the packet inter arrival rate is relatively high i.e.$\geq$ 1 pkt/minute.

Interestingly *IX-MAC* also outperforms ContikiMAC when packets are being forwarded infrequently. This is counter intuitive, as the layer 1 dual stage receive check mechanism used in ContikiMAC should provide lower power consumption compared to *IX-MAC*'s layer 2 receive check (see Table 3.1 on page 49). To investigate the reason for the discrepancy between the idle power consumption of the power consumption figures present in [17] and the measured values in the Contiki operating system, an experiment was performed. This experiment should also serve as direct comparison between *IX-MAC* and ContikiMAC in an idle listening only state on the same hardware platform.

The TelosB was chosen as the hardware platform for comparison and receive check intervals of 1 and 2 seconds were used. Best efforts were taken to implement precisely the dual stage receive check described in ContikiMAC [17]. Between the two short CCA checks, the CC2420 is transitioned to an idle state where the oscillator is still running but all RF blocks are disabled, additionally the MSP430 is put in a low-power mode between CCA checks. Figure 4.11 shows the current consumption profile of how ContikiMAC's dual stage receive check was implemented.

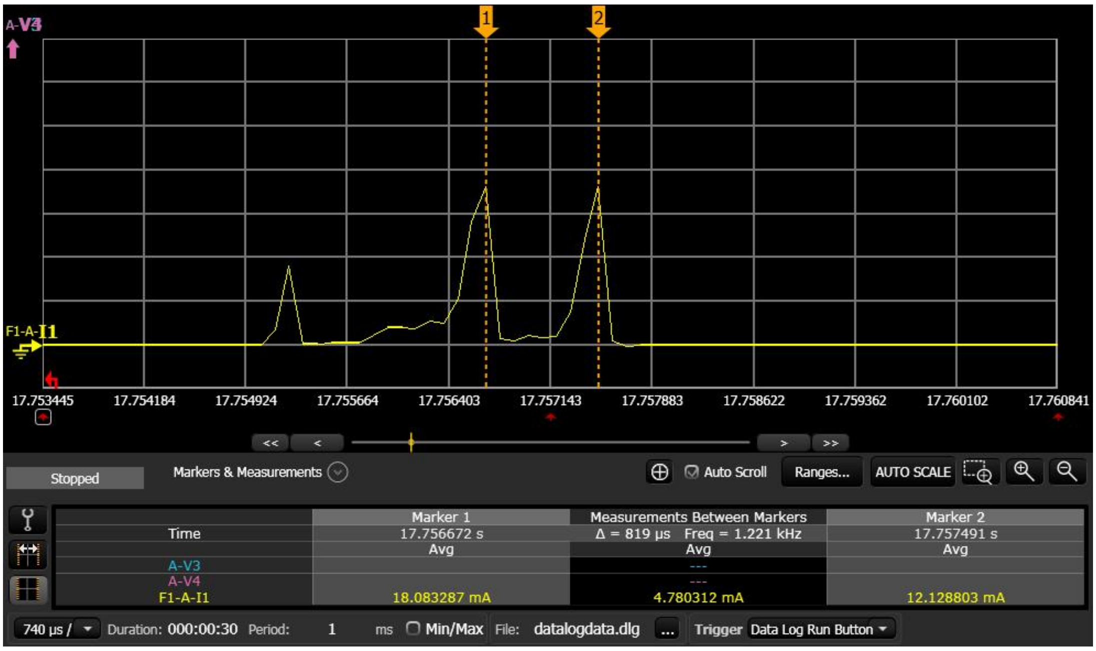


Figure 4.11: **Contiki Receive Check Implementation**

The same Agilent DC power analyser was used to measure the idle power consumption of ContikiMAC and *IX-MAC* on the TelosB platform. The results are given in table 4.6. This implementation of ContikiMAC achieves lower power consumption than the *IX-MAC* protocol. Both protocols achieve identical sleep current of 10 μA between receive checks, therefore the reduced power consumption exhibited by ContikiMAC is due to the shorter and lower energy layer 1 dual stage receive checks. The version of ContikiMAC

Table 4.6: **Idle Power Consumption Comparison**

Protocol	**Current (μA)**
ContikiMAC 1 s	20
ContikiMAC 2 s	15.5
IX-MAC 1 s	38
IX-MAC 2 s	24

implemented in the Contiki operating system resulted in a much higher power consumption (see Figure 4.10). Using a receive check interval of 0.5 s a current consumption of 115 μA was measured, this was measured forwarding one packet every 10 minutes (this is infrequent enough to not impact on average current consumption). The discrepancy between the ContikiMAC version in Contiki and this implementation can be explained by a number of factors. 1) The Contiki operating system may not achieve the same sleep current for the TelosB. 2) The Contiki operating system may have periodic timer interrupt events occurring, waking the MSP430. 3) The implementation of the dual stage receive check may not have been implemented as per the description in [17].

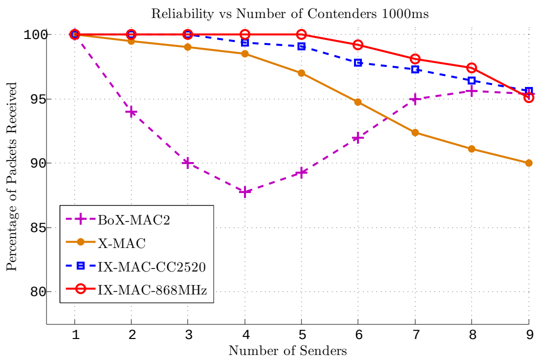
This implementation of ContikiMAC achieves lower idle average current consumption than the Contiki operating system version. The reduction in current consumption only applies to periodic listening events and this reduction would be subtracted across all measured values in Figure 4.10. This change would only affect the performance of the protocol when packets are being forwarded infrequently and the performance increase of *IX-MAC* at frequent packet forward intervals would still remain.

4.6.2 Reliability and Scalability

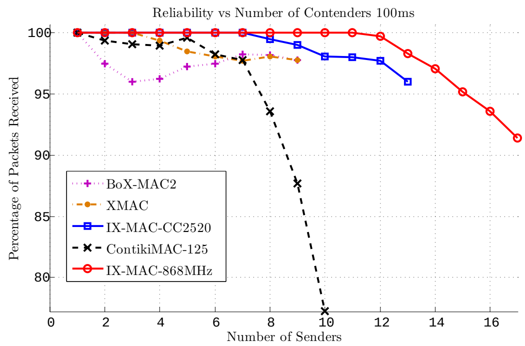
This experiment was devised to measure the performance of *IX-MAC* under contention and to compare its performance against the state of the art. This experiment simulates the conditions which would be present in a dense network scenario, it measures the reliability of packet delivery under contention.

4.6.2.1 Experimental Setup

An experiment similar to that in [57] was conducted to measure the reliability and scalability of *IX-MAC* against the current-art MAC protocols. This experiment tests the following features of the protocols: *Reliability, Scalability,*



(a) Reliability under Contention, T_W= 1s



(b) Reliability under Contention, T_W= 100 ms

Figure 4.12: **Reliability vs. Number of Concurrent Senders**

CSMA Fairness and *Collision Detection/Avoidance*. Two (common) receive check intervals (100 ms and 1 s) were chosen for this experiment. All sending nodes were programmed to send 1000 unicast packets, at a rate of 1pkt/s, to a single node. The receiving node was configured to perform receive checks at the same rate as the sending nodes.

4.6.2.2 Results, Reliability under Contention

The results from these tests are presented in Figures 4.12b and 4.12a. For T_W=100ms, *IX-MAC*, operating on the 868MHz platform, achieves 100% *reliability* (defined here as the number of packets received/sent) for up to 11 concurrent senders, and slowly degrades to 91.4% with 17 senders. *IX-MAC* on the 2.4GHz platform achieves 100% for 7 senders, and degrades to 96% for 13 senders. The best of the alternatives tested is X-MAC, which provides 100% for only 1 sender but remains above 95% for 9 senders. BoX-MAC2 also provides 100% for 1 sender, but 96% for 3 senders before returning to 97.4% for 9 senders. ContikiMAC remains above 95% for 7 senders, but degrades to 77% for 10 senders.

For T_W=1s, *IX-MAC* operating on the 868MHz platform achieves 100% reliability up to 5 senders, degrading to 95.1% with 9 senders. *IX-MAC* on the 2.4GHz platform achieves similar performance, degrading to 95.6% for 9 senders. The best of the alternatives is X-MAC, providing 100% for 1 sender, and remaining above 90% for 9 senders. BoX-MAC2 provides 100% for 1 sender, dipping to 88% for 4 senders, before returning to 95.4% for 9 senders.

4.7 Discussion

IX-MAC outperforms the current art in terms of reliability and scalability for ultra-low power operation. It provides 100% packet delivery for up to 11 concurrent senders, while the best of the known, available alternatives provide 100% for 1 sender. This is due to the CSMA implementation in *IX-MAC*, and disallowing multiple transmitters to occupy the channel simultaneously. Optimal parameters obtained from optimization work in Section 3.4 ensure turnaround times are precise and predictable. Increases in reliability can also be explained by failsafe mechanisms built into the protocol, including: frequency hopping for payload delivery, ultra-efficient RTS/CTS with payload ACK, and reduced transmission lengths.

IX-MAC improves upon the current art using a number of power saving techniques, some of which were optimised from previous work. These techniques include: Neighbour Schedule Learning, TX power optimization, and optimized receive check lengths. It improves on power consumption by a minimum factor of 6.6 and a maximum of 16.6. This can extend to an order of magnitude improvement over other protocols under similar operational conditions. It outperforms WiseMAC (the gold standard in low-power MAC protocols for wireless sensor networks) in situations where packets are forwarded frequently. The neighbour schedule learning scheme presented in this work also greatly simplifies the version presented in WiseMAC.

IX-MAC's time synchronization protocol does not incur large overheads in microcontroller resources or increased radio transmissions. These are normally required for time synchronization protocols. Also shown is that *IX-MAC* successfully decouples the energy required to transmit data to a sleeping neighbour node and the network's receive check interval.

4.8 Conclusions

At the outset of this chapter, a number of theoretical issues were presented and analysed. Sections 4.1 and 4.3 investigated how oscillator drift affects duty-cycled MAC protocols that do not use receive check scheduling techniques. Section 4.4 described how oscillator drift affects the efficiency of the novel neighbour schedule learning scheme proposed in this work. Optimal operating parameters were derived and the hardware and software measures taken to counteract oscillator drift were described. In Section 4.5, a power consumption model of *IX-MAC* was given. The modeled version of *IX-MAC* outperforms WiseMAC for packet inter arrival times of less than 15 seconds.

In Section 4.6, *IX-MAC* was compared against the state of the art in terms of power consumption and performance under contention. It was shown empirically that *IX-MAC* provides excellent reliability and robustness, balancing the need for ultra-low power operation and reliability. Due to its efficient CSMA implementation and channel hopping, it outperforms the state of the art under contention in dense networks and shows it enables improved reliability in dense deployment scenarios. It outperforms the state of the art in terms of power consumption due to its semi-synchronous operation, whereby nodes can learn the receive check schedule of neighbouring nodes without direct exchange of scheduling information.

Chapter 5

Design and Evaluation of a Latency Aware Routing Protocol for Low-Power Duty-Cycled WSN

This chapter describes the design and implementation of a new networking protocol for low-power and high reliability WSN applications. It features a novel weighted parent selection process that considers multiple parameters when selecting parent nodes. Another novel feature is a latency aware parent selection mechanism where nodes consider end-to-end latency when selecting parents.

Section 5.1 gives a brief introduction into routing protocols and their function in WSN. In Section 5.2 an overview on the development of the existing state of the art routing protocols for low-power WSNs is given. This Section also highlights the strengths and weaknesses of existing approaches. Section 5.3 describes the development of the latency aware routing protocol developed in this work and explains its functionality. Results from a 52 node deployment are presented in Section 5.4. The chapter is concluded in Sections 5.5 and 5.6.

5.1 Introduction

Layer 3, or networking layer protocols are an integral part of the OSI (Open Systems Interconnection) model of computer networking. This layer leverages the MAC layer to receive and send data packets. Its main functions are route discovery and reliable forwarding of packets in the network. In the context of this work, the network topology used is a converging tree based multi-hop topology with a single sink node. The traffic pattern for data collection is converge-cast, as is typical for WSN data gathering applications. It uses a Destination Oriented Directed Acyclic Graph (DODAG), similar to RPL [36]. All nodes in the network are capable of sensing and extra routing nodes are not used. A diagram of a convergent tree based single sink tree topology is shown in Figure 5.1.

In this work and low-power WSN applications in general, the main responsibilities/ tasks of the networking layer are described below:

- Route Discovery: The networking layer is responsible for discovering potential parent nodes and deciding which parent/ parents would provide the most reliable path/ paths to the network sink. An example of this is a node that is within transmission range of the network sink, the task of the route discovery engine is to detect this direct link and address all traffic to the network sink. For robust reliable operation in dynamic and harsh RF environments, the route discovery is responsible for finding multiple potential parents for the case that the radio link to the first preference parent becomes lossy. When no parents can be found, nodes must continue to periodically search for potential parents.

- Loop Avoidance: The networking protocol must avoid creating routing loops. Routing loops occur when a node which previously forwarded a packet receives the same packet.

- Packet Forwarding: On a per node basis, packet forwarding is responsible for reliably delivering data packets to parent nodes. It leverages the information gathered during the *Route Discovery* phase about potential parents. Based on this data, it attempts to deliver packets to the chosen parent. If the chosen parent fails or is unreachable, the packet forwarding engine can either try the same parent again, or it can choose the next preference parent if one exists. The packet forwarding engine is also responsible for carrying out multiple MAC level transmission attempts. One attempt is equal to T_W seconds worth of RTS/ CTS attempts due to the receiver duty-cycled nature of this work.

- Route Maintenance: Due to the unpredictable and dynamic nature of the wireless medium, link qualities will vary over time [6, 55]. The task of the route maintenance engine is to keep track of these changes and to dynamically choose better quality links, as they become available. The scenario where nodes are placed in/ removed from the network also exists and this must be dealt with accordingly. Route discovery is an expensive process in terms of energy consumption and a balance between truly dynamic behavior and low-power operation must be met for long life deployments.

For example, CTP [28] uses periodic beacon transmissions to maintain the network, these consist of broadcast transmissions which last for one full T_W interval. Using the CC2420 transceiver and a T_W of 1 second, each beacon costs a minimum of 60 mJ (assumes I_{RX}=I_{TX}=20 mA, 3 V operation).

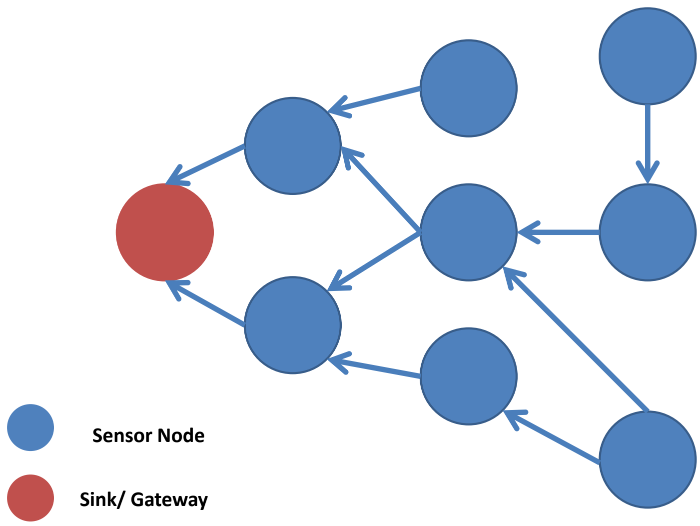


Figure 5.1: **A convergent tree topology with a single sink node**

5.1.1 Routing Protocol and its Impact on Network Performance

The policies of the underlying routing protocol impact greatly upon the characteristics of the deployed network. In general, optimising for one desired performance characteristic may impact the other desired performance traits. The following is a brief overview of the important performance characteristics, with attention paid to how they are intertwined:

- **Latency**: *End-to-end latency* in WSNs is defined as the time taken for a packet generated by a source node to reach a destination node. Typically, latencies experienced are non-deterministic in real RDC WSNs. Many potential applications for WSN's are sensitive to end-end latencies. Some specific examples of these time sensitive applications would be intruder detection systems for building security and smoke/ CO2 detection systems.

  For both of the aforementioned examples, it is critical that the network sink is notified of any events within an application specific time bound. The maximum delay between detection of an event and reception at the network sink must include the per-hop latency incurred by each hop in the network. In a multi-hop WSN deployment with a maximum of 10 allowed hops and a bound of 1 second on end-end latency, the maximum per-hop latency would be 0.1s (100ms) if a packet was to require 10 hops to arrive at the network sink. The most common approach to solve such an issue is to deploy a network which performs frequent receive checks. The impact of this is increased power-consumption due to frequent receive checks.

  To date there has only been one approach to lowering latency at the routing layer. This opportunistic routing approach is explained in Section 5.2.

- **Reliability**: In the context of the primary application for this work, the reliability of the deployed network is of utmost importance. Each node is required to report back to the network sink on a periodic basis. If one of these periodic updates is missed, an alarm is raised that a device has malfunctioned. A malfunctioning smoke/ CO detector can pose a serious risk to the occupants. On the other hand, a fully functional smoke/ CO detector that is deemed to be non-functional because of an under performing wireless network, is also problematic. In this work, reliability is measured by examining the timely arrival of expected updates from nodes in the deployed network.

  The task of the routing protocol is to guarantee high levels of reliability/ quality of service. It should choose paths to the network sink which guarantee a high probability of successful delivery. Bearing in mind that the strongest link quality paths to the network sink may not be those which offer the lowest latency/ least traffic, a compromise between reliability/ low latency and traffic balancing must be struck when selecting parents. In [71], the authors present a detailed study on the reliability of wireless links based on link quality measurements at the MAC layer.

In the event of communication failures between child and parent nodes, the routing protocol must respond quickly by attempting to deliver the packet to the next preference parent. If the network sink is expecting an update from a particular node and it does not arrive on time, a fast response is needed to comply with strict timing requirements on packet inter-arrival times. There also exists a close relationship between the reliability of a deployed network and the power consumption, recall from Figure 3.2 on page 40 that each transmission attempt consumes excessive energy in a duty-cycled system. A reliable network is usually one which also exhibits low-power consumption.

- **Power Consumption**: The power consumption of the deployed nodes will be influenced by a number of factors: 1. Data Rate: The rate at which nodes must report back sensor readings to the network sink will strongly influence the power consumption. This will dictate the amount of time that the radio transceiver remains active for. 2. Load Balancing: The distribution of traffic in the network will impact the power consumption of the nodes. Their power consumption will be directly proportional to the number of packets which they must forward per unit time. There also exist natural imbalances between nodes, depending on their position in the network. Leaf nodes do not forward packets for other nodes and only send one packet per data send interval. Nodes located more centrally in the tree topology must potentially forward packets for multiple child nodes, this means they perform multiple transmissions per data send interval. Sink neighbours may be required to forward multiple packets per data send interval, but each packet they forward consumes low energy due to the fact that the network sink is always listening. 3. Reliability: As mentioned previously, expensive failures will increase the power consumption dramatically. 4. Route Maintenance: The rate at which nodes wish to update their knowledge on prospective neighbours will also impact the power consumption. A trade-off between latency/ power consumption performance must be found here.

The ideal routing protocol for a low-power duty-cycled WSN deployment should exhibit excellent performance in all three of the aforementioned characteristics. It should provide the low-latency performance of an always on 100% CSMA network, close to 100% reliability, at a power consumption level which guarantees a 10-year battery life. As will be seen in Section 5.2, previous work in routing protocols for low-power duty-cycled WSNs also targets these performance traits. Certain protocols address low-latency operation, others reliability and power consumption. In this work a new protocol was developed that performs well against all of these performance metrics.

5.2 State of the Art Review

In previous years there have been many advances in the area of networking protocols for low-power multi-hop WSNs. Listed in approximate chronologically descending order are the major contributions from previous years:

- Let the tree Bloom: scalable opportunistic routing with ORPL [18]
- Broadcast Free Collection Protocol [68]
- Low-power, Low delay: opportunistic routing meets duty cycling [44]
- Low-power Wireless Bus [25]
- RPL [36]
- DISSense [12]
- Collection Tree Protocol [28]
- Dozer [9]

Dozer, developed at ETH in Zurich in 2007 by Burri, Rickenback *et al.*, was designed for ultra low-power operation. It achieves a radio duty cycle of 0.2% and was designed for environmental monitoring. This radio duty cycle results in very low quiescent power consumption and makes it suitable for long-life applications. The authors present results from a 40 node deployment using the TinyNode 584 hardware [34]. It uses a TDMA type MAC protocol and is a cross-layer approach to routing. Nodes maintain two schedules, one for their parent and their own independent schedule, which their child nodes must follow. Parents are selected on the basis of only two parameters, depth in network and child count (load).

All nodes wake every 30 seconds and transmit a short beacon packet, after the beacon is sent they remain in receive mode, waiting to receive data from potential child nodes. When a node wishes to join the network it first goes into an always listening state, listening for beacons from potential parents. After selecting a parent, the child node synchronises to the beacon interval period of their parent and wakes up in time to hear their parent's beacon. A parent can receive data from multiple child nodes after sending its beacon, thanks to a time slotted approach. Each child node is given a time-slot during which it should send its data to its parent.

Dozer does achieve very low radio duty cycle and power consumption due to its large in-active sleep periods. When using a beacon interval of 30 seconds, a node is on average able to sleep for a full 15 seconds before it must wakeup. During this 30 second beacon interval, it must both send its own beacon and it must forward any data to its parent. This approach gives very high end-end packet latencies. Worst case scenario, a node wishing to send data, must wait 30 seconds for its parent node to wake and send a beacon. Over multiple hops this does not scale well and can lead to minutes between a packet being generated by a leaf node before it's received at the network sink. This approach is also not very suitable for event detection type applications such as, smoke/ CO alarms due to it's long periods of inactivity.

CTP (Collection Tree Protocol) published in 2009, remains to be a very popular routing protocol for sensor networks, as identified in [27]. CTP was originally written for the TinyOS operating system but has since been ported to work with Contiki. CTP creates a convergent tree type topology and can work with single or multiple network sinks. CTP uses an ETX metric (first mentioned in [15])to aid in parent selection. ETX stands for the expected number of transmissions and is a measure of the probability with which a packet can be delivered to a particular parent and the number of transmissions to the network sink.

All nodes transmit periodic beacons, when a node hears a beacon from a neighbouring node it looks at the sequence number of the transmission. If the next beacon that is received from the same node has a sequence number that is not equal to the old number plus 1, a beacon was missed and the ETX number for that node is decreased. If all beacons are received that node is deemed to provide a very stable link. The ETX metric increases linearly along each level in the network tree, nodes closest to the sink have the lowest ETX and leaf nodes the highest. CTP does not explicitly consider link quality of RSSI data of potential parents, it does however consider this when two parents have the same ETX score. Beacons are transmitted less and less often as the network becomes more stable, this is governed by Trickle [48]. Beacons are transmitted when a change in network topology is detected, these beacons also trigger neighbouring nodes to transmit beacons.

CTP does not consider latency, as it does not factor per-hop latencies (of the links along its path to the sink) into the calculation of the ETX metric. Due to the duty-cycled nature of low-power WSNs, the one-hop latencies between nodes will be randomly distributed across the network with an average value of $T_W/2$. Nodes are not allowed to elect parent nodes that are at the same level in the tree as themselves (loop avoidance). Lower latency paths to the sink could often be achieved by selecting nodes at the same level in the tree which may have low-latency paths to their parents etc.

DISSense published in 2011 created by Colesanti, is designed for ultra-low duty cycle operation. The basic principle of it is the following: if nodes are supposed to report sensor readings on a 10-minutely basis, all nodes wake every 10 minutes and switch to a 100% active mode. During this 10-minute active period, nodes can send their sensed data to the network sink over a network that is 100% active, removing the transmission and reception problems associated with duty-cycled approaches. All nodes must be time synchronised to share a common notion of time and schedule to wake every data send interval. To facilitate this, beacons are transmitted to indicate the time remaining until the next active collection phase will begin. Between active data collection phases, nodes switch to duty-cycled operation waking every few seconds to check for incoming packets. DISSense is very well suited to applications where only periodic sensor readings are required, but it cannot provide event detection capabilities because of its long inactive periods. The authors quote a radio duty cycle of 1.09% with a collection interval of 1 minute and 0.22% for 15 minutes.

LWB (Low-Power Wireless Bus) developed by Ferrari, Zimmerling et al. in 2012, is a flooding based approach. It is based on earlier work in Glossy [24]. All nodes in the network are synchronised to wake and perform receive checks together, this approach of all nodes waking simultaneously is also used in [86]. If a node has data to send, it does so at the beginning of a wakeup interval. Neighbouring nodes will be awake and listening at this point, upon reception of a data packet they retransmit the same packet concurrently, if these concurrent transmissions are close enough in time they create constructive interference and this improves packet reception probability for neighbouring nodes. Using this technique a node can flood the entire network with a single transmission. A transmission using this flooding technique is illustrated in Figure 5.2

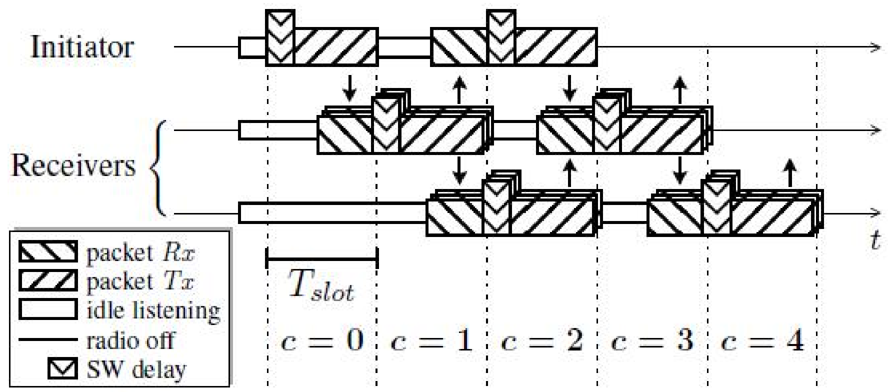


Figure 6: Example of a Glossy flood with $N = 2$. *Nodes always transmit packets with the same relay counter c concurrently.*

Figure 5.2: **A flooding transmission, taken from [24]**

Flooding latency is the average time taken for a node's transmission to reach all other nodes in the network with its transmission. LWB achieves very low average flooding latency of 1.77 ms with a network of 5 hops. LWB requires relatively long receive check lengths due to the fact that any node can send a packet to be flooded during every wakeup event. If a network has a maximum of 8 hops and maximum length payloads must be catered for, then nodes closest to the network sink must remain in receive mode for 32.768 ms(assumes 128 byte payload 802.15.4) during each receive check event. The advantages of a flooding approach such as this are ultra-fast event detection, low duty cycle operation for applications where frequent reliable sensor readings are required. The disadvantage of flooding approaches is that they are not suitable for low-rate networks where sensor readings are required perhaps every 5 minutes or more. Here nodes would need extra synchronisation packets to maintain time synchronisation, this is due to large time gaps between transmissions where oscillator drift can accumulate and result in time offsets between nodes' receive check events.

RPL is a routing protocol that was designed to operate on top of the 802.15.4e MAC, it was created by a work-group of the IETF [1]. RPL is designed for low-power and lossy WSNs and it supports IPv6 based routing providing low-power devices with internet connectivity. It is optimized for operation in networks where nodes are predominantly required to periodically report measurements to a small number of collection points. RPL labels this type of communication as Multi-Point-to-Point (MP2P). RPL also provides basic

[1]http://www.ietf.org/

functionality for configuration traffic, this traffic is from the collection points to the networked nodes and is typically infrequent. RPL labels this type of traffic as Point-to-Multi-Point (P2MP).

RPL supports P2MP communication by each node appending their ID to the Reverse Route Stack of forwarded messages. Network sinks are known as low power and lossy network Border Routers (LBR). RPL uses a gradient based system, whereby nodes determine their *Rank* in the DODAG. RPL is largely based on CTP and borrows many of its underlying mechanisms. RPL is advantageous in that it supports P2MP traffic.

ORW developed in 2012 by Landsiedel, Ghadimi et al. is an opportunistic routing protocol designed for low latency, low duty cycle operation. ORW is opportunistic because it does not use a traditional neighbour discovery phase. ORW replaces the traditional 16-bit destination address field in the MAC header of 802.15.4 packets with 2 values. The first is called EDC, it is a measure of the number of potential forwarders a node has and the quality of these links, the second value is the required routing progress. Using the second value, nodes which receive packets decide if they can forward them or not. When a node running ORW wishes to send its sensed data to the network sink, it begins broadcasting its packet and the first node which happens to wake and can progress the packet becomes the forwarder by acknowledging the transmission. This anycast approach has two functions, it gives low one-hop latency and it results in a low radio on-time per transmitted message.

ORW offers low radio duty cycle and latency thanks to its opportunistic routing approach. It is also robust to link failure as it can try multiple potential forwarders within a single T_W interval. ORW does not offer load balancing and may also select sub-optimal parents in terms of end-end latency. For example, Node A happens to wake up just after Node B begins sending its anycast message, it has sufficient Link Quality to acknowledge B's message and can progress the packet toward the sink and it becomes B's parent. Unfortunately Node A's only possible forwarding options happen to wake up a very long time after A begins forwarding B's message ($\approx T_W$). In this scenario ORW will provide sub-optimal end-end latency for Node B's messages. Different options may have offered lower end-end latency if Node B had chosen them. Unfortunately in ORW, nodes are not aware of the latency of downstream nodes. ORW also potentially suffers from duplicate packets being received at the network sink, this occurs when a node fails to receive an ACK from a parent but the parent genuinely received the packet. This results in the same packet taking multiple routes to the network sink. The drawbacks associated with opportunistic approaches are potential duplicate packets, difficulties in detecting routing loops and shallow neighbourhood knowledge.

BFC (Broadcast Free Collection) protocol was developed in 2012 by Daniele Puccinelli, Giordano et al. Its primary aim as the title suggests is to create a broadcast free routing protocol. Broadcast beacons are typically used during neighbour discovery phases, here nodes broadcast their position and status in the collection tree. Neighbouring nodes which hear these broadcasts consider the nodes who broadcasted these beacons as being potential parents. An example of this is a node which has a direct path to the network sink that would advertise this periodically by transmitting broadcast beacons. For example, nodes which hear this broadcast and do not have a path to the network sink would choose this node as their parent.

Broadcasts are expensive in terms of energy per broadcast, as a broadcast typically lasts for the duration of one T_W interval. BFC relies on snooping/ overhearing on messages being transmitted in its vicinity. Node A that does not have a path to the network sink may overhear a neighbouring Node B's transmission. If this packet is addressed to the network sink, then Node A will assume that B has a direct path to the network sink and B will become A's parent. Node A will now forward its sensed data to node B which will in turn forward the packet to the network sink. The same applies to nodes further removed from the network sink, only here parents may no longer have a direct path to the network sink.

When nodes wish to learn parents available to forward their data towards the network sink, they switch to an always listening mode and attempt to snoop on overheard transmissions. Only after a node has transmitted a certain number of successfully acknowledged packets to its parent, does it set a Viability bit in the MAC header to let overhearers know the link is stable. Nodes look at the source, destination and viability fields of overheard transmissions and select parent nodes based on these. BFC does achieve broadcast free tree construction and collection, but it does so at a cost. BFC only improves over CTP when receive check intervals are above 5 seconds (broadcast lasts for one T_W). This reason for this is because of the costliness of broadcasts with increasing T_W. BFC must listen for long periods to find parent nodes. If parent nodes are potentially only sending once every 5 minutes and 5 nodes in the neighbourhood have stable paths, does this mean a node needs to listen for a few minutes before it will overhear a transmission? This fact also makes nodes slow to react to changes in network topology.

ORPL is an extension of the ORW routing protocol. It extends ORW by adding any-any packet transmission. Any-any transmissions differ from traditional traffic where all nodes send data to a single network sink node. Any-any traffic allows the network sink to be able to reach any node in the network (endpoints), this type of traffic is useful when a user may wish to remotely actuate a load connected to a low-power wireless node in the mesh network.

5.2.1 Delay Aware Routing Protocols for Multi-Hop Low-Power WSNs

This section gives an overview of delay-aware, low-power protocols. The need for low-power operation in WSN's results in long in-active sleep periods, this comes at the expense of increased latency. Certain WSN applications have strict delay requirements on the arrival of sensor readings, these applications include industrial control loops where sensor readings must be as 'real time' as possible. For delay critical applications, TDMA synchronous MAC protocols are typically used [74].

In [74], the authors present GinMAC, it is a MAC protocol designed to deliver sensor readings over multiple hops within application specific time limits. In the evaluation, a fixed topology is used with a maximum of 3 hops, 100% of the sensor readings arrived within the required 1 second reporting interval. GinMAC uses predetermined time slots as part of its TDMA operation, synchronisation overhead traffic is also required. It contains a number of reliability enhancing failsafe features to ensure packets arrive within 1 second at the network sink. Using the application specific relatively high sample rate of 1 Hz, the average per node duty-cycle was 0.76% and a maximum of 2.48%. Where sensor readings must be provided frequently and within specific time bounds as in [74], TDMA approaches function well.

In [73], the authors present Dwarf, a cross-layer, multi-hop, delay-aware, low-power contention based protocol. It was designed for use in similar applications as the applications described in this thesis. Dwarf aims to reduce end-end delays in multi-hop networks using an asynchronous MAC protocol, it is not able to provide a guarantee on the end-end delay without a time synchronous approach such as in [74]. The MAC protocol used in Dwarf is based on WiseMAC. It reduces end-end delays by nodes selecting next hop parents based on the proximity of their wakeup schedules and their level in the tree topology. Dwarf is delay aware, however Dwarf does not consider end-end latency when selecting parent nodes and this can lead to sub optimal performance over multiple hops. This occurs when a node selects a particular parent due to its wakeup schedule proximity, without considering how much time the parent requires to forward the packet to its parent.

PEDAMACS [21] is a TDMA protocol, it guarantees contention free end-end delay guarantee and low-power operation. PEDAMACS uses the network sink to coordinate the entire network, the network sink is also responsible for allocating time-slots for transmissions. Based on information gathered by each node in the network during the learn phase, the network sink executes a scheduling algorithm, determining transmission scheduling in the network. This is done by the network sink analysing the connectivity map of the network, one of the main tasks of the scheduling is to avoid contention by assigning geographically close nodes with different transmit schedules. The scheduling algorithm now knows precisely which route each will select enabling a guarantee on the end-end delay. PEDAMACS is based around a network sink that is capable of communicating with all nodes in the network over a single hop (direct link). They say this is feasible as the network sink is usually connected to a mains power source and it can afford to use higher TX power levels. In reality, due to legal restrictions, this approach is unlikely to be suitable for industrial or commercial applications where the frequency bands used are regulated.

5.2.2 Summary and Insight, State of the Art Review

Each of the aforementioned protocols have strengths and weaknesses. The flooding based protocols such as Glossy and LWB give very low latency data and reliable operation, but suffer on power consumption. Others such as Dozer and DiSSense provide ultra low-power operation, but provide poor flexibility and are not suitable for applications where event detection capabilities are required. CTP provides good performance in highly dynamic networks with its periodic beacon transmissions, but gives increased duty cycle and high latency sensor readings.

It is interesting to look at the parameters which each of the above protocols use

Table 5.1: **Parent Selection Parameters**

Protocol	LQI	Hop Count	Latency	Traffic	Energy
Dozer	×	✓	×	✓	×
CTP	✓	✓	×	[2]	×
DiSSense (Uses CTP)	-	-	-	-	-
ORW	×	✓	✓	×	×
BFC	✓	×	×	×	×

when selecting parent nodes. Gomez *et al.* in [29], present a detailed study of the various parent selection parameters used by different popular routing protocols. They explain the parameters used in the respective objective functions for parent selection. Of the protocols they studied, the parameters used for parent selection are hop-count, Link Quality Indicator (LQI), RSSI or packet delivery-based link quality metrics. None of the studied protocols consider the latency of potential routes, which is an important metric when duty-cycled networks are used. In Table 5.1 a brief summary of the parameters which each of the major protocols consider is given. Best efforts were taken to include as accurately as possible all of the parameters which these protocols consider. LWB and Glossy are not listed as these are flooding based protocols.

Except for Dozer considering load-balancing, none of the other studied protocols directly consider energy or load-balancing for parent selection. Load-balancing is the process of distributing workload evenly throughout a network, it has two primary functions: 1) It ensures even power consumption amongst nodes in the network. 2) It increases reliability when network traffic is high. Recent work in 2013 by Liu, X et al. extended RPL to include load-balancing [53], Arbutus [69] in 2008 also considers load-balancing. The authors of [18] mention this as being a topic for future work.

The aforementioned routing protocols do not consider remaining energy when selecting parents. Energy based selection constraints steer network traffic away from nodes with depleted energy resources. This helps to ensure well balanced power consumption and reliable maintenance free operation.

The only non-flooding based protocols which consider latency are ORW and ORPL. Because these protocols route opportunistically they achieve low-latency operation by default without considering latency in parent selection. ORW and ORPL also perform well under lossy links, their anycast style forwarding can

[2] CTP does mention a congestion bit, parents set this when they are dropping packets due to congestion

Table 5.2: **Routing SoA, Performance Metrics Overview**

Protocol	**Reliability**	**Radio Duty Cycle**	**Comment**
CTP	95%	2.2%	5 minute interval, T_W=2 s
ORW	95% + (est)	1.1%	5 minute interval, T_W=2 s
BFC	Unknown	1%	No reliability data provided, T_W=2 s
LWB	99.97%	1.69%	1 minute interval, T_W=0.5 s

reach multiple potential forwarders with a single T_W interval. Protocols that use a neighbour discovery phase such as [28, 36] do not consider the latency of the paths which they select.

In CTP, the exchange of information required for routing parent selection is based around the reception of beacon transmissions. A node wishing to join the network must somehow trigger its neighbouring nodes to send beacons, or wait for them to send one to be able to join the network. CTP broadcasts a beacon on startup, any node which hears this beacon schedules its own beacon transmission. The original node wishing to join the network then receives these triggered beacons and looks at the information contained within. Based on this information it will choose a parent and join the network.

This system is inefficient in that neighbouring nodes are forced to transmit a beacon just because another node wishes to join the network. A better system would be one where routing information can be exchanged more efficiently between nodes. This work proposes a system whereby nodes exchange routing information in payload acknowledgment packets. This removes the need for parent nodes to transmit beacons to allow new nodes to join the network.

Table 5.2, gives a brief overview of the performance metric presented in the various publications associated with the different bodies of work. In ORW no precise reliability data is presented and in BFC no reliability data at all. However, the authors of CTP do provide detailed reliability data. For 100% on CSMA only approaches, CTP achieves close to 99.9%, as soon as a duty-cycled approach is used the authors present figures of around 95% reliability. The figures presented in LWB are very impressive, 99.97% reliability at 1.69% radio duty cycle. However, the depth of the network was quite limited (3 hops) and the duty cycle of LWB will scale directly with increasing network depth.

In terms of delay aware routing protocols based around CSMA contention based protocols, Dwarf is one of the only protocols which actively selects parent nodes based on delay. However it does not consider the end-end latency that each parent would provide if selected, nor does it contain a mechanism by which nodes can calculate and advertise the end-end latency of their chosen paths to the network sink.

To summarise, having studied the capabilities of the existing art, it was felt that improvements could be made in the following key areas:

- Smarter Exchange of Routing Information: The way in which the route discovery phase is carried out in the existing art can be improved. The majority of the existing art use beacon broadcast transmissions to perform route discovery. A node cannot join the network until it has heard a beacon transmission from a neighbouring node, or in the case of BFC, until it has overheard a unicast transmission taking place in its neighbourhood. This work proposes a smarter more energy efficient technique for route discovery in duty-cycled WSN's.

- Latency Awareness: Of the aforementioned protocols, only the flooding based routing approaches achieve very low end-end latency over multiple hops. The low end-end latency achieved in flooding based protocols comes at the cost of additional synchronisation overhead and increased power consumption. The opportunistic routing approaches such as [44, 4] achieve low latency in some scenarios by selecting the first parent which happens to acknowledge the packet (and can progress it). They do not use a traditional route discovery phase and are therefore not aware of the latency their neighbouring nodes would provide if selected. This work aimed to develop a routing protocol whereby nodes are provided with end-end latency estimates during its neighbour discovery phase. This improves on the approach used in Dwarf [73] whereby nodes are only aware of the node-node delay.

- Multi-Parameter Consideration: This work aims to improve on the current art by considering multiple parameters when selecting parent nodes. After reviewing the parameters used by the current art in routing protocols, improvements could be made by considering multiple parameters during the parent selection phase. Existing approaches tackle specific areas, attempting to maximise or optimise performance in that respect. However these issues are intertwined and therefore should be handled together. This was a goal of this work.

5.3 Latency Aware Protocol Design

5.3.1 Design Goals

The design goal of the new routing protocol was to overcome the shortcomings of the existing art. The proposed novel routing protocol should also help to meet the system level performance requirements. It is called LARP (Latency Aware Routing Protocol). Listed below are the areas in which improvement was sought after, and how it was proposed to be achieved.

- **Latency**: Without using an opportunistic or flooding routing approach, the only way in which a low-latency protocol for duty-cycled operation can be created, is by having latency aware parent selection. The first step to achieving this is to enable latency measurement at the MAC layer. The cross layer approach used in this work, achieves latency measurements at the MAC layer by careful receive check and transmission start scheduling (explained in Section 3.3.3).

  To facilitate this, nodes need to be able to estimate the cumulative end-end latency which they would provide to upstream nodes if they were to be selected. If nodes wishing to join the network only consider the latency of their one-hop parent (instead of the entire path), this will result in sub-optimal parent selection. Therefore nodes must consider the latency of the entire path to the network sink over multiple hops. During the neighbour discovery phase, nodes should factor in the latency which each potential parent would provide into a weighted selection process. Careful consideration and thorough testing needs to be carried out to find the optimum weighting factor. Too much emphasis on selecting low-latency paths could result in unreliable and lossy links.

- **Load Balancing**: As node/ traffic density increases, the importance of load balancing increases. The number of packets a single receiver duty-cycled node can forward is proportional to the T_W interval of the MAC layer. If a node has a T_W=1 second, in the best case scenario it can only receive/ forward one packet per second. This places a fundamental bound on the maximum throughput of a duty-cycled WSN deployment.

  Load-balancing attempts to share workload evenly throughout a network of distributed nodes. In an unbalanced network where the majority of the workload is carried out by a minority of nodes, energy consumption tends to be unbalanced and reliability tends to decrease due to collisions. If a particular node must forward packets frequently from downstream child nodes, there is an increased chance of contention. In the context of this work and low-power networking for duty-cycled WSNs, workload is defined as the number of packets a node forwards per unit time. For convenience, the unit chosen is the sensor sample interval (SSI), this is the period of sensor readings in the network. A leaf node which does not forward packets for neighbouring nodes has a workload of 0, a parent node with one dependent child node has a workload of 1, etc.

Dynamic estimation of a node's workload is a trivial task, nodes must simply track the number of packets they forward and calculate the workload periodically.

- **Energy Awareness**: Energy awareness and load balancing go hand in hand. A network which has well balanced traffic loads, also has an even power consumption distribution, however even with a best-effort load balanced network, the power consumption of individual nodes will vary. Transmission failures for example will cause excessive energy wastage (illustrated in 3.2). The only convenient way for nodes to account for their own local power consumption is by having a software based power consumption estimator, alternatively an additional IC can be used to facilitate this.

- **Smart Exchange of Routing Information**: As explained previously in Section 5.2, traditional routing protocols use beacon transmissions to exchange routing information and to update changes in network topology. Dynamic changes in network traffic patterns and topology will affect how reliably sensor readings can be gathered from the network.

  In an ideal world, nodes would broadcast any detected changes to their immediate neighbourhood. These changes may be a new child node it has inherited or lost, a sudden reduction or increase in its link quality. Unfortunately, in duty-cycled networks, it is extremely costly to broadcast/ advertise this information. Therefore, it is proposed to exchange this information in a more energy efficient manner.

5.3.2 Tree Formation and Neighbour Discovery

The very first step in forming the converging single sink tree, is for nodes with a direct path to the network sink to discover this link. Nodes with a direct path to the network sink will be referred to as sink neighbours. On startup, nodes attempt to send a packet to the network sink. If an acknowledgment is received within 10 RTS/ CTS attempts, the node updates a variable in RAM to reflect this direct path (*Hop Position*=0). If no path to the network sink is detected, nodes schedule a *Distress* transmission in the hope of finding neighbours which do have established paths.

Distress transmissions are the mechanism by which nodes try to identify and learn potential parents. They consist of a broadcast transmission which lasts for one T_W interval. Any nodes within transmission range will receive the packet and respond with a CTS and payload acknowledgment. The data contained in the payload acknowledgment contains all of the necessary routing information and is analysed by the sender of the broadcast. If it detects a node with its *Hop Position* field less than 31, it knows this node has an established path to the network sink and knows its position in the tree hierarchy.

Starting from a default value for *Hop Position* of 31, nodes will set this value equal to the value of *Hop Position* in the received acknowledgment + 1, if the received value is less than its own value. If a node A does not detect a direct path to the network sink but detects Neighbour B which has a *Hop Position* of 0 (sink neighbour), this means A is at level 1 and it sets its *Hop Position* to 1. Further nodes wishing to join the network that hear a response from both A and B during their learn phase, would establish their level in the tree as being 1, not 2. Using this scalable technique, nodes learn their position in the tree relative to the network sink. This concept is portrayed graphically in Figure 5.3.

5.3.3 Latency Aware Routing

This aspect of the routing protocol was designed to reduce the typical end-to-end latencies present in asynchronous duty-cycled WSNs. Due to the duty-cycled nature of low-power asynchronous WSNs, the per-hop latency is inherently high when compared to always listening/ synchronous approaches. *End-to-end latency* in WSNs is defined as the time taken for a packet generated by a source node to reach its destination. The worst case scenario end-end

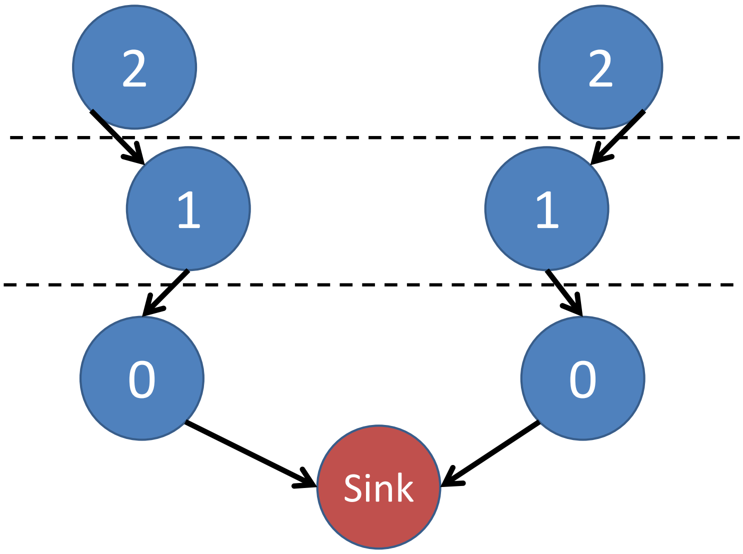


Figure 5.3: **Depiction of distinct levels in the tree topology.**

latency for a multi-hop WSN deployment is $\approx T_W$ seconds per hop. The best case scenario end-end latency is proportional to the bit rate of the transceiver in use, the processing time at each hop and the number of hops.

The initial body of work carried out that resulted in the creation of this Latency Aware Routing was presented in [62], wherein the relationship between clocks, latency and energy efficiency in multi-hop WSNs was analytically and empirically evaluated. It was hypothesised that coupling this information with a routing protocol could be exploited to develop enhanced communications protocols.

The proposed Latency Aware routing mechanism introduces a novel latency metric in the specification of the *objective function* used to select routes towards the destination node. Thus, nodes select parents not only based on traditional parameters such as link quality, hop count, or workload, but also on an estimate of the end-end latency provided.

Given the relative lack of latency consideration in objective function computation for parent selection in multi-hop WSN routing protocols, this work investigates the potential to develop a latency-aware routing protocol for low-power, reliable data transfer. To achieve this, latency, LQI (Link Quality Indicator) and other information from potential parents are gathered during the learn phase. Therefore, it can make better-informed decisions on which parent provides the best trade-off between latency and link quality, whilst retaining low-power operation. Recent approaches only leverage LQI type information from the MAC layer.

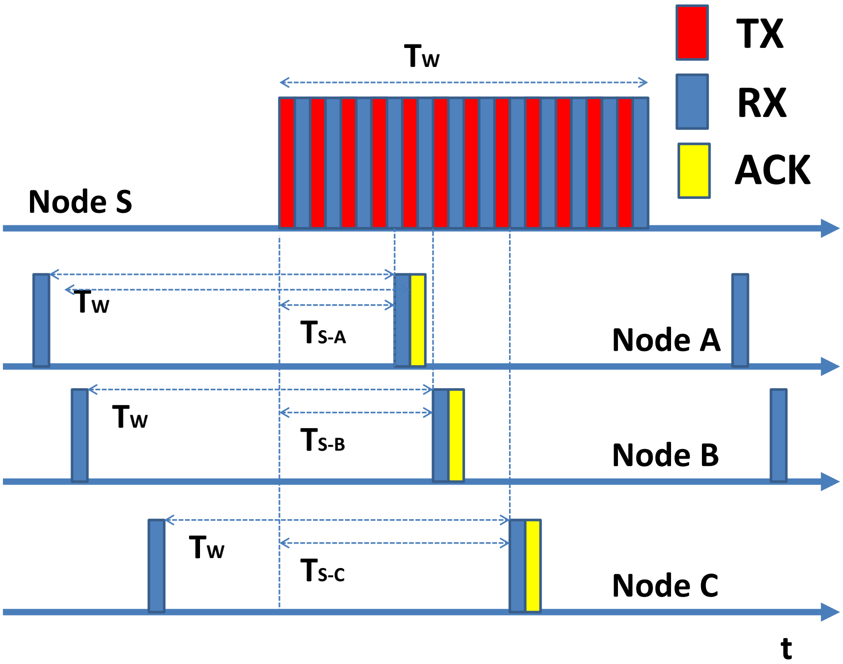


Figure 5.4: **Reaching multiple nodes with one broadcast**

The acknowledgment-based MAC protocol in this new routing protocol always begins transmitting at specific times and counts the time taken for neighbouring nodes to respond (acknowledge). It stores this time value in a neighbour specific data structure, and updates it during each exchange to account for drift (see Section 3.3.3). The rules, governing receive check and transmission start scheduling, enable the measurement of the one-hop latency available through each neighbour. Figure 5.4, shows how a sender can learn multiple one-hop neighbour latencies with a single broadcast transmission. The time taken for an ACK to be received is the one-hop latency.

In this latency aware routing protocol, a system was developed whereby each potential parent provides an estimate for the end-end latency which it would provide if chosen during the neighbour discovery phase. In most multi-hop duty-cycled WSN's, the sink node always listens and always responds immediately. Therefore, nodes communicating directly with the sink incur minimal latency. Nodes with a path to the sink are at Level 0. Nodes that have a path to these, but not to the sink, are Level 1 etc (CTP uses same approach). Level 0 Nodes advertise their *Latency Constant* to be 0. Level 1 Nodes which have chosen Level 0's as their parents, advertise their *Latency Constant* as being the time offset between their wakeup-schedule and those of their Level 1 parents. Nodes at Level 2 advertise a *Latency Constant* that is the sum of the latency to their Level 1 parent and the *Latency Constant* of their Level 1 parent. Using this scalable technique, each parent advertises a cumulative estimate of the end-end latency that it would provide if chosen. This cumulative latency technique and the proposed hierarchal structure are illustrated in Figure 5.5. Arrows indicate the one-hop latency and the value inside the Nodes is the cumulative latency. Other recent approaches do not offer a latency estimate during neighbour discovery phases.

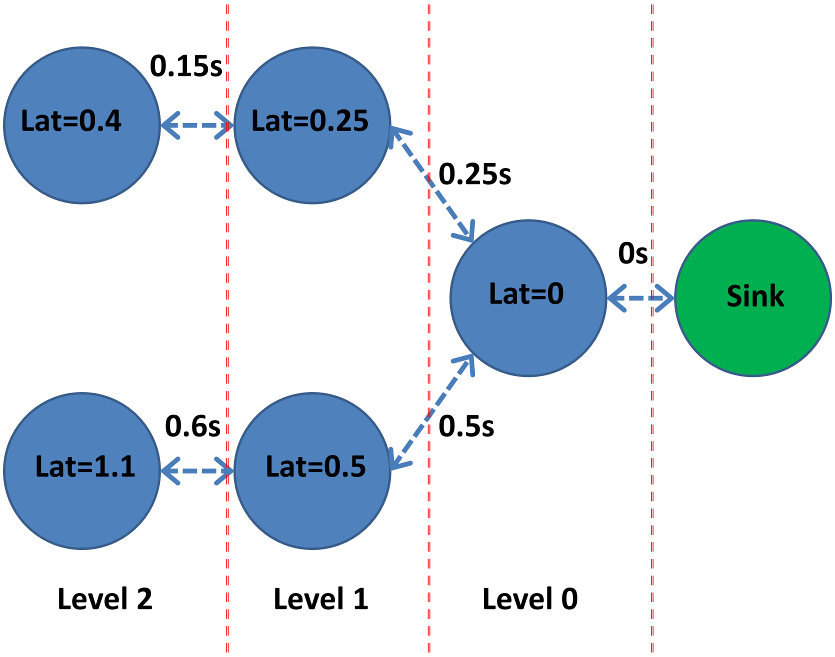


Figure 5.5: **Cumulative Latency**

5.3.4 Beaconless exchange of Routing Information

The aim of this technique is to create a mechanism by which nodes can exchange routing information in an energy efficient manner. To avoid the requirement for a node to trigger neighbouring nodes to send beacons before joining the network, a novel method to exchange routing information is proposed. The technique used in this work leverages the payload acknowledgment packet to exchange information needed for parent selection. When a node wishes to join the network, it sends a broadcast message for T_W seconds. Neighbouring nodes which receive this broadcast, respond with their routing information in the payload acknowledgment. The reception of the broadcast does not trigger these nodes to send any of their own beacons.

Using this method, nodes wishing to join the network can now gather the routing information of potential parents in one T_W interval. Within the length of one broadcast transmission (T_W seconds), it can collect the routing information of all nodes in its RF vicinity. Previous techniques require nodes to wait for the triggered beacons of parents to be received and it increases the power consumption of neighbouring nodes unnecessarily.

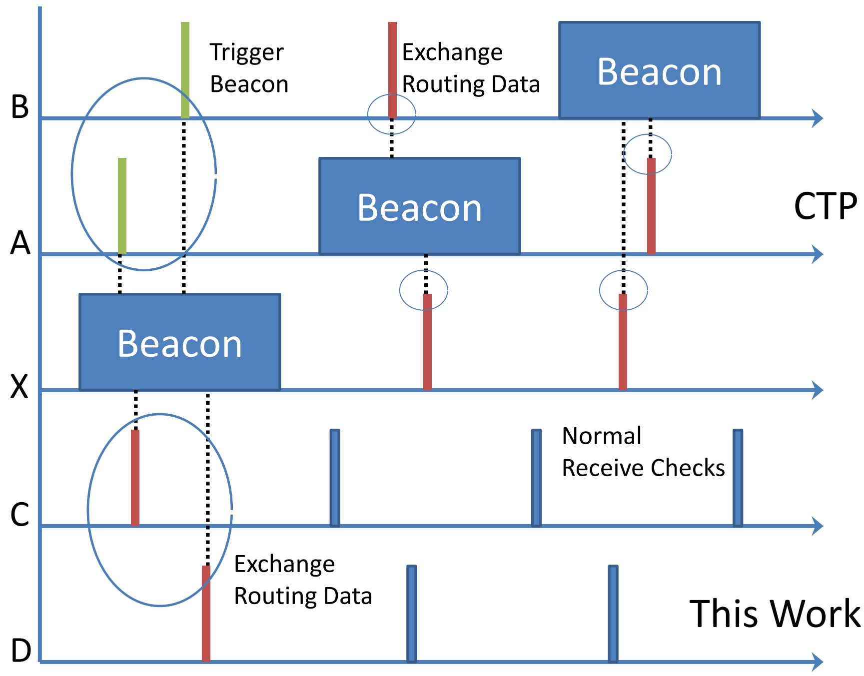


Figure 5.6: **Routing Information exchange Comparison**

Figure 5.6 shows the above described technique graphically and compares its operation to that of CTP. Node X wishes to join the network, neighbouring nodes for CTP are A and B, neighbouring nodes for this work are C and D. In CTP, Node X transmits a beacon that is received by A and B. A and B schedule individual beacon transmissions to exchange their routing information with X. Node X receives A and B's routing information during their beacon transmissions. The entire operation requires 3 energy hungry beacon transmissions.

In this work, Node X transmits a beacon, C and D receive it and respond by supplying node X with routing information in the payload ACK. C and D are not triggered to transmit subsequent beacons. This work achieves exchange of routing data from C and D to X with 1 broadcast opposed to CTP's 3.

Nodes can learn the routing information of neighbouring nodes during reception, or transmission of a packet. A node that has recently joined the network will advertise its position and status in the network by sending a single broadcast. This broadcast is sent after it has chosen a stable parent and delivered a number of successful unicast transmissions. Neighbouring nodes which receive this broadcast message examine the contents of the MAC header of the message payload and assign the node with a routing score. If this score is greater than the score of their current parent, they will switch to the new node which advertised itself.

5.3.5 Parameters Considered During Parent Selection

The 4 factors on which the parent selection process is based are listed in Table 5.3. Each parameter is assigned a weighting factor, this weighting factor determines what percentage of the overall score that each potential parent is assigned with. The weighting factors used were derived heuristically by deploying numerous multi-hop networks and analysing the routes selected in the network. The weighting factors were modified to the point where the network performance was satisfactory. This section justifies the selection of the parameters that were chosen.

LQI is the parameter that is given the largest weighting factor. It is given the largest weighting factor due to goal of achieving ultra-high reliability. LQI is used as a parameter in the parent selection process to ensure nodes select routes which are stable and unlikely to result in transmission failures. Links that offer strong LQI levels result in a lower probability of transmission failures. If the TX power is 0 dBm and the receive sensitivity -90 dBm, the link budget is 90 dB. If a receiver receives a signal at a level of -60 dBm, then it falls within 30 dB of the overall link budget. This 30 dB margin offers a level of protection where received signal strength levels can vary dynamically due to changes in the environment or multi-path fading.

Latency is the parameter that is given the second largest weighting factor. Latency is used as a parameter in the parent selection process for two unique reasons. 1) The end-end latency estimate provided by potential parents during the neighbourhood discovery phase allows nodes to consider the end-end latency that each potential parent would provide if they were chosen. This results in lower end-end delays for transmissions in receiver duty-cycled multi-hop networks. For applications where end-end delay management is critical, end-end delay based route selection is critical. 2) When the end-end latency parents would provide is factored into the parent selection process, this impacts on the length of the selected routes. Shorter routes with fewer hops will inherently offer lower end-end latency, therefore these routes will be preferred.

Child Nodes is the parameter that is given the third largest weighting factor. It is used for load balancing purposes to ensure a more even distribution of the workload throughout the network. Load balancing helps to create an even power consumption distribution between nodes. Without load balancing there is a risk that many nodes will choose the same parent to forward their packets. This creates an imbalance between nodes' power consumptions with certain nodes meeting the target battery life of 10 years, while others may fall short of that due to excessive workload.

Battery voltage is the final parameter that is considered during parent selection, it is given the lowest weighting factor. This parameter is considered in an attempt to help nodes meet the 10 year battery life requirement. The instantaneous battery voltage is a good indicator of the remaining energy in the node's battery.

5.3.6 Assigning Parents with Routing Scores

During the neighbour discovery phase, each neighbour which responds and has a *Hop Position* field of <31, is assigned a score/ *Attractiveness Factor*. It is designed to give parent nodes with high LQI, low latency, low traffic, high available energy a high score and parent nodes with low LQI, high latency, high traffic, low available energy a lower score. The neighbouring node with the highest *Attractiveness Factor* is selected, this is achieved by performing a modified bubble sort on the first 20 stored routing options. The *Attractiveness Factor* is 11 bits in length, representing a maximum score of 2047.

During the transmission of a broadcast message, the values used to assign an *Attractiveness Factor* to each neighbour which responded with an acknowledgment, are taken from the payload acknowledgment and locally from the MAC layer. During the reception of a packet, the values used to assign an *Attractiveness Factor* to the sender are taken from the MAC header of the payload frame. Listed in Table 5.3, are a list of the parameters (and their approximate relative weighting) which are considered in assigning each neighbour with an *Attractiveness Factor*. The distribution of the weighting

Table 5.3: **Considered Parameters**

Parameter	**Weighting A**	**Weighting B**
LQI	50%	40%
Latency	30%	50%
Child Nodes	10%	10%
Energy	10%	0%

factors shown in Table 5.3 was derived heuristically. Intuitively, the LQI must retain a high weighting in order to achieve reliable links. Results are presented in Section 5.4 where network performance vs. weighting factors is studied.

The exact objective function used to assign each neighbour with a routing *Attractiveness Factor* is described mathematically in the following pages.

5.3.6.1 Link Quality

Link Quality considerations account for 40% of the overall score. The function that is used to calculate the score does not consider the average Link Quality of the entire path, but only the Link Quality of the local path to the parent and the Link Quality of the parent's chosen parent. The logic behind this is to avoid the possibility that a very poor link quality of one link along a path could be disguised by stronger link qualities around it.

To emphasize this concept, consider the following example; a Link Quality metric varies from ≈ 20-100, 100 being the best possible Link Quality and 20 the minimum at which a packet can be received. Imagine two paths, A and B, both are 3 hops in length. Path A offers one-hop Link Qualities of 100,100 and 25, path B offers 75, 75 and 75. Both when summated give an identical average Link Quality of 75 (225/3), but path B would provide a much more reliable link because all of its links have a minimum quality of 75. Path A offers 2 hops at 100 and one very weak hop at 25, this hop at 25 is much more likely to cause transmission failures than 3 hops at 75 each.

To overcome the issues associated with averaging Link Quality information over multiple hops, only the Link Quality of the direct path to the parent and that of the parent's path to its parent are considered in this work. Bi-directional Link Qualities are considered, the received Link Quality level of the parent's ACK/ CTS packet and the Link Quality level at which the parent received the sender's RTS packet. A simple non-linear function is used to discourage the use of weak links, it is applied before the Link Quality values are passed to

Equation 5.1. Low Link Quality values are penalised so as to make these routes even less attractive, if the Link Quality is below a certain threshold, a further fixed number is subtracted to lower the overall score. This helps to avoid choosing low-latency links which offer poor Link Quality.

In this work, the actual value used to represent Link Quality varies from the 868 MHz implementation to the 2.4 GHz one. The 868 MHz implementation uses RSSI and a history of previous transmission failures to assign each link with a Link Quality estimate. The 2.4 GHz implementation combines the automatically appended end-of-frame LQI byte, RSSI, and a history of previous link failures. For every link failure (failure to receive an ACK) a fixed value is subtracted from the score. This technique leverages some of the ideas presented in [71], where the authors use a combination of RSSI and LQI to assign an overall link quality score. This work uses a weighted combination of RSSI and LQI in assigning parents with link quality scores.

$$LQI_{\text{Score}} = \frac{LQI_{\text{TX}} + LQI_{\text{RX}} + LQI_{\text{Parent}}}{3 \times LQI_{\text{MAX}}} \tag{5.1}$$

5.3.6.2 Latency

The portion of the overall score assigned to the latency consideration is calculated cumulatively. If the parent for which a score is being calculated is a sink neighbour, latency is not considered as these nodes all have the same zero latency path to the always listening sink. The score is the sum of the individual latencies of the hops along the path and the equation to calculate it is shown in 5.2. Each node advertises the latency of their parent plus the latency to that parent.

$$Latency_{\text{Score}} = 1 - \frac{\sum_{i=1}^{HopCount} Latency_{\text{i}}}{HopCount * T_{\text{W}}} \tag{5.2}$$

5.3.6.3 Load Balancing

In the context of this work, the workload of a node in a multi-hop network is the number of packets it forwards per data send interval. Load balancing is achieved by nodes being aware of the their own local workload and using this knowledge to avoid it from inheriting new child nodes. There are multiple possible ways to achieve load balancing. Some of these are summarised below:

- **1:Workload of their Immediate Parent**: The advantage of this approach is the simplicity of its implementation. The disadvantage of this approach is that it achieves only superficial local load balancing and not global balancing.
- **2:Workload of their Parent and Parent's Parent**: The advantage of this approach is improved global load balancing. The disadvantages of this approach are a slightly more complicated implementation and still sub-optimal global load balancing.
- **3:Nodes Consider the Workload of the Node at the Root of their Sub-Tree**: The advantage of this approach is almost optimum global load balancing. The disadvantages of this are a more complex implementation and sub-optimal local load balancing. Nodes no longer know which immediate parent in the same sub-tree has the lowest load if all advertise the load of the node at the root of the sub-tree.

To achieve optimum load balancing this work uses a combination of techniques 1 and 3 listed previously. All nodes advertise the workload of the node at the root of their sub-tree and their own local workload. Load balance tracking is achieved by nodes counting their local workload. If the data send interval is 5 minutes and a node forwards 5 packets in 5 minutes, its load is 5. When nodes exchange routing information during 'Distress' transmissions, they exchange both the workload of the root node of the sub-tree and their local workload.

The formula used to calculate the portion of the overall score given to load balancing consideration is shown in Equation 5.3.

$$Child_{\text{Score}} = \frac{RootLoad}{MaxLoad} \tag{5.3}$$

The local workload of immediate parents is considered when two parents have extremely similar routing *Attractiveness Factors*, depending on the optimisations which are enabled, a post-sort can push the scales in favour of the immediate parent with the lower local workload.

5.3.6.4 Energy

The portion of the overall score assigned to energy awareness is calculated semi-cumulatively. It is based on the node's battery voltages. 4 bits are used to represent the battery voltage, representing 16 distinct levels. Nodes advertise the average of their battery level and that of their parents. This technique also helps to highlight nodes with depleted battery levels, instead of masking it by averaging over multiple nodes. The equation for this is given in Equation 5.4.

$$Energy_{\text{Score}} = \frac{Parent_{\text{Voltage}}}{Max_{\text{Voltage}}} \tag{5.4}$$

The final equation used to assign each parent with a routing score is given in Equation 5.5.

$$\begin{aligned} Routing_{\text{Score}} = {} & (Energy_{\text{Score}} \times Energy_{\text{Weight}}) + (Child_{\text{Score}} \times Child_{\text{Weight}}) \\ & + (Latency_{\text{Score}} \times Latency_{\text{Weight}}) + (LQI_{\text{Score}} \times LQI_{\text{Weight}}) \end{aligned} \tag{5.5}$$

5.3.6.5 Nonlinear Final Selection Steps

With overall reliable packet delivery being the main goal of the routing protocol, a number of non-linear steps are taken to ensure reliable links prevail.

- A final sanity check is performed on the stored values for the first preference parent. This final check ensures that the link quality of the chosen parent is strong enough to ensure reliable packet transmissions. If it isn't and the second option does provide stronger link quality, the second and first options are swapped.
- If a node has detected it is a sink neighbour, it will only address its packets directly to the network sink if it deems the Link Quality to be strong enough. If a weak path to the network sink is detected, the node will learn further routing options by broadcasting a distress message. If one of these options provides a more reliable link to the network sink, it will be chosen over sending directly to the network sink. This comes at no real extra cost per transmitted message because of the delay before send neighbour schedule learning scheme used.
- In an attempt to reduce the number of hops, the final selection steps ensure that a parent node that is at a higher level in the tree topology is

not selected. This step also helps minimise the chance of creating routing loops.

- A number of other post steps can be applied, these include sorting nodes with similar scores based on their local workload and LQIs.

5.3.7 Message Piggybacking

Message Piggybacking is used to reduce the number of overall transmissions in the network. The result is lower radio duty cycle, lower probability of contention occurring and higher reliability. The primary task of each node in the WSN is to report periodic sensor readings to the network's sink. To reach the network sink, nodes at the outer edges of the deployment may have to route through several nodes, depending on the density and RF environment of the deployment.

The underlying idea of this piggybacking optimization is as follows: nodes that happen to lie in a path that has neighbouring nodes generating or forwarding data packets will piggyback their sensor readings into messages which are being forwarded. This process is described graphically in Figure 5.7. Traditionally, this is done differently with each individual node generating its own periodic data messages. Additionally, nodes that are forwarding data packets are smart enough to still only perform one sensor reading per required sensor sample interval.

To accommodate piggybacking, the payload is partitioned into different blocks. Each node which forwards the packet, adds its sensor readings to the payload in a specific position. The position is dependent on the hop number. The leaf node that generated the packet, adds its sensed data to position 0. The next node to interact with and forward the packet adds its sensed data to position 1. In this implementation and application, each node adds a total of 8 bytes to the payload and the length of the variable data packet. Additionally, packets contain a routing header. Forwarding nodes increment a *Hop Count* value, and add the ID and LQI of the last hop to the routing header. Packets that originate from nodes which have a direct RF path to the network sink are short. Payload lengths grow linearly as the hop count increases.

When the network sink receives a data packet, it first examines the hop count of the packet in the routing header. Depending on this value, it knows how many nodes have interacted with the packet and included sensor readings. It is also

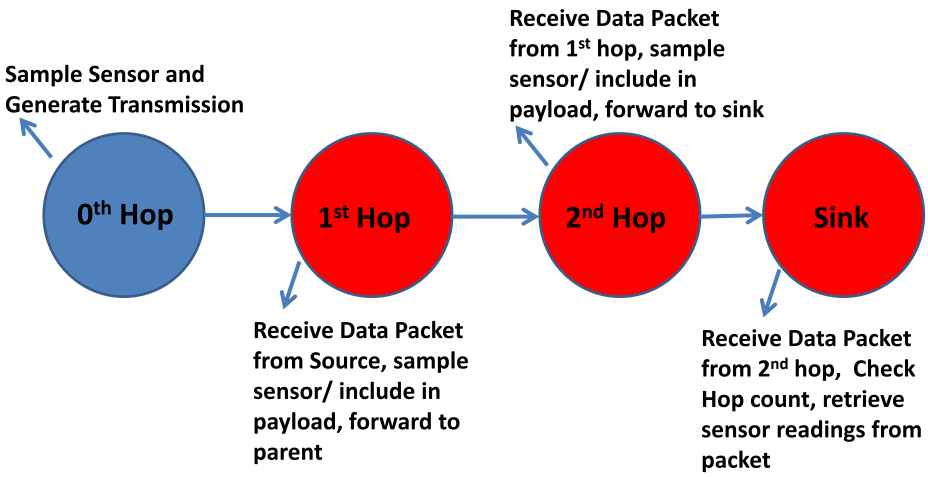


Figure 5.7: **Data piggybacking. Each node along the path adds their sensor readings to the payload of incoming messages. Reduces extra wasteful transmissions**

aware of the position in the payload where to find sensor readings from the N^{th} hop.

With transceivers such as the CC2420 and CC2520 supporting 128 byte payloads, these piggybacking technique can comfortably scale to 10 hops where each node adds 8-10 bytes to the payload. This technique would work especially well on transceivers which support longer payload lengths or infinite payload lengths.

5.3.8 Loop-back Avoidance

Loop-backs occur when a packet which a node generated/ forwarded traverses through the network and returns to the same node. The node that receives this packet with which it already interacted, must decide whether to attempt re-sending the packet (at the risk of further loops), or to kill the packet. In this work, a number of steps are taken to ensure loop free routing. The first precaution taken to avoid loop-backs, is by parent nodes not sending packets back to the last hop node. It is only allowed in extreme cases, if the child node has other routing options and the parent's other options have all been exhausted. During the parent selection process, nodes are disallowed from selecting nodes at higher levels in the tree if nodes at lower or equal levels exist and offer above threshold link quality.

A final precaution is the use of a routing header at the beginning of packets. Nodes which forward a packet append the ID of the last hop to the routing header. Using this method nodes can avoid forwarding packets to nodes that have already interacted with the packet. When this is detected, nodes choose another stored option to progress the packet. This process is described in Figure 5.8 where **X** denotes a lossy link, S is the network sink. When the link from A to E fails, A sends to B. B must notice A is its parent, it sends to C, C does not send the packet back to A because A's ID is in the packet, C instead sends to D, providing a loop free route.

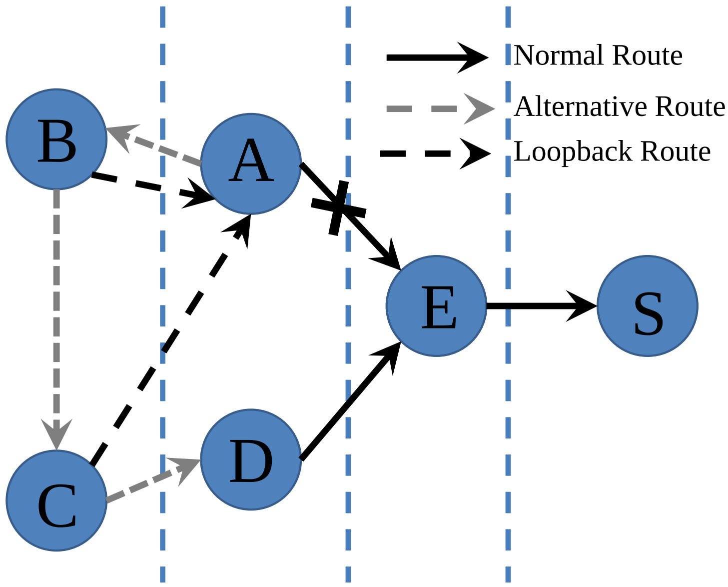


Figure 5.8: **Loop-back Avoidance**

5.3.9 Routing Information in Payload ACK and MAC Header

The payload ACK packet is 9 bytes in length, it is used to exchange routing information. The structure of the packet is given in Table 5.4. The exact structure of the payload ACK changes depending on the node's position in the tree hierarchy, sink neighbors use the structure given in Table 5.5, non sink neighbours use the structure given in Table 5.4.

Table 5.4: **ACK and MAC Header Structure, non Sink Neighbour**

Bits	0-7	7-15	16-23	24-31	32-34	35-39
Value	Length	Dest	Src	RTS LQI	Local Load	Hop Pos
Bits	40-47	48-51	51-54	55	56-63	64-71
Value	Parent LQI	Root Load	Battery	Options	Latency	Parent Score

Table 5.5: **ACK and MAC Header Structure, Sink Neighbour**

Bits	0-7	7-15	16-23	24-31	35-39	
Value	Length	Dest	Src	RTS LQI	Hop Pos	
Bits	40-47	48-51	51-54	55	56-63	64-71
Value	Sink LQI	Root Load	Battery	Options	Reserved	Reserved

5.4 Evaluation

This deployment was designed to measure the performance of the designed routing protocol in a multi-hop scenario. The three metrics of interest were radio duty cycle, reliability and end-to-end latency. In this evaluation, the 868 MHz platform was used. 52 nodes were deployed in a building spanning 3 storeys, 1 single sink node was deployed on the 3rd and top storey. 17 nodes were deployed on the ground floor, 17 on the 1^{st} floor, and 17 on the 2^{nd} floor. The dimensions of the building are: L:60m, W:70m, H:20m, that is comparable to popular testbeds, e.g. [32] and [16]. Two versions of the protocol (Version A & Version B) were experimentally tested and verified, Table 5.3 shows the weighting factors used for A and B. Each experiment was conducted for 3 days. Version A was optimised for reliability, and does not consider the latency of its paths. Version B was optimised for latency reduction, it considers the latency of its routes, it also transmits periodic beacons every 30 minutes. Creating these two versions was achieved by slightly modifying the weighting factors assigned to the parent selection criteria.

Each node includes a 1-byte estimate of the one-hop latency of its parent in packets which are generated/ forwarded. This allows for the end-to-end latency of packets to be calculated at the network sink. Nodes also include a software estimation of the radio duty-cycle. A T_W of 1 second was used and nodes were configured to provide temperature readings every 4 minutes.

5.4.1 Metrics Under Test

5.4.1.1 Latency

The end-end delay recorded by each node is used for comparing the latency performance of the 2 versions of the protocol. This measurement is broken down into average number of hops at each node, and the end-to-end latency.

To verify the designed latency reduction algorithm, it was necessary to devise a system to measure end-end latencies for a multi-hop WSN. In testbeds such as [16, 80, 32], each node has a physical USB connection, allowing for latencies to be measured. Without easy access to such a testbed, the payload is modified to include an estimator of the latency for each hop. Recalling from Section 3.3.3, each node knows the latency to its parent by counting the number time taken before an ACK is received. This figure is accurate to ±2 ms for the 868 MHz implementation. Nodes count the time-offset (latency) of each transmission and include this information in the payload. This is done at each hop, and allows the end-end latency of the packet to be calculated at the network sink with adequate accuracy.

5.4.1.2 Reliability

Reliability is measured by analyzing the timely arrival of sensor readings from each node. If nodes are configured to sample their sensors and report these readings every 3 minutes, a packet is judged to have been missed if it is not received within 10% of the allowed sample interval.

5.4.1.3 Power consumption

Each node in the network is capable of monitoring its radio duty-cycle. This information is included in each transmission and can be used to translate the radio duty-cycled into a power consumption estimate for the nodes. The lower the achieved radio duty-cycle, the lower the node's power consumption.

5.4.2 Per Node Comparison

In Figure 5.9 and Figure 5.10, results of the 3 day long experiment are presented. In Figure 5.9, averaged results for each node are depicted in terms of achieved duty cycle and end-end latency. Figure 5.10, presents the reliability

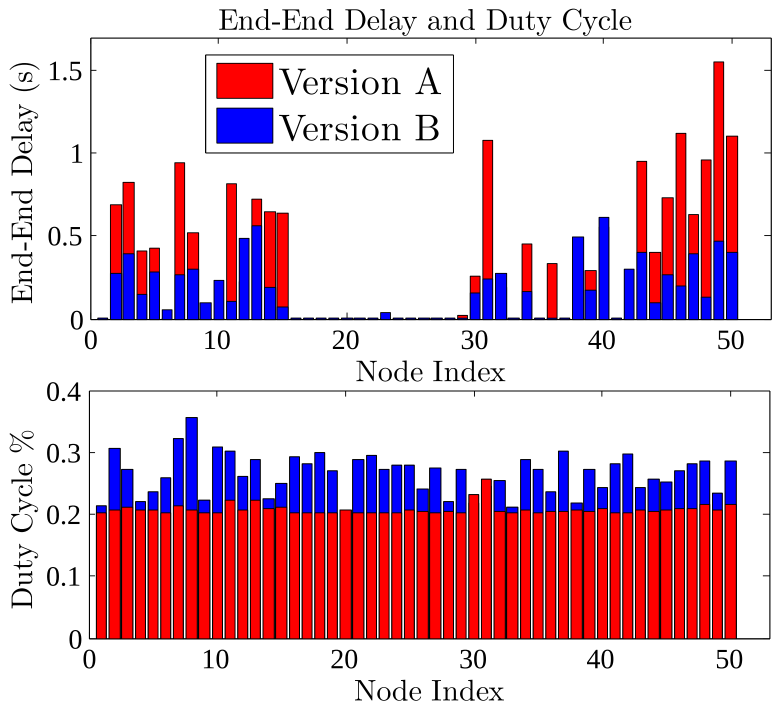


Figure 5.9: **End-End Delay Top and Radio Duty Cycle Bottom. 52 node deployment**

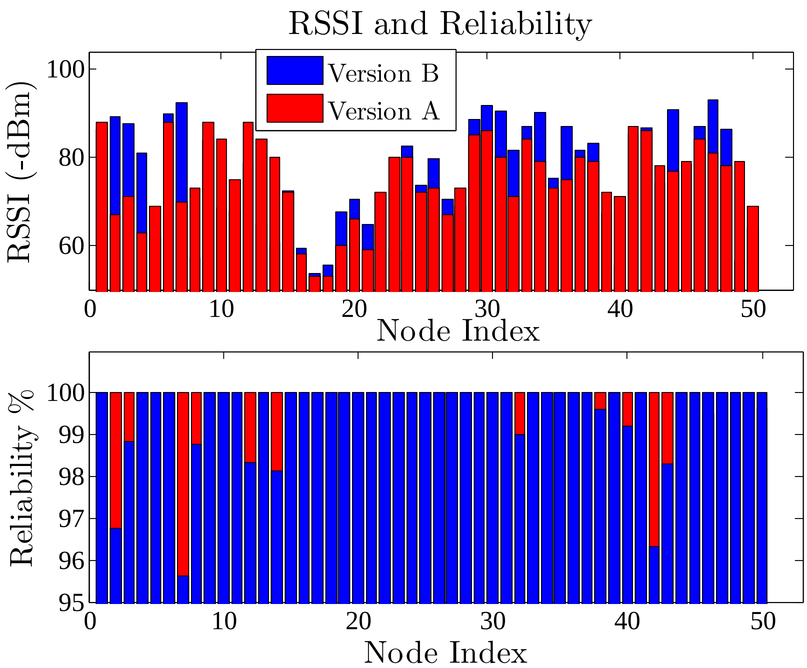


Figure 5.10: **Average RSSI of node's parent top, Reliability on bottom. 52 node deployment**

achieved by each node on bottom, and the average RSSI of their chosen parents on top. The average RSSI of the chosen parent(s) reflects the reliability of the

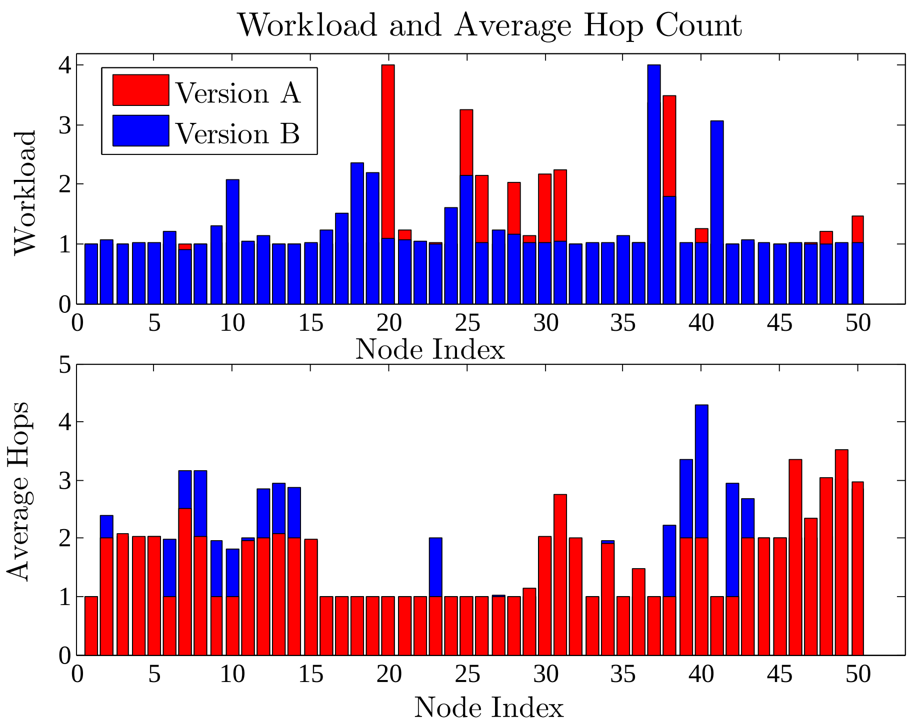


Figure 5.11: **Workload Distribution top and Average Number of Hops bottom. 52 node deployment**

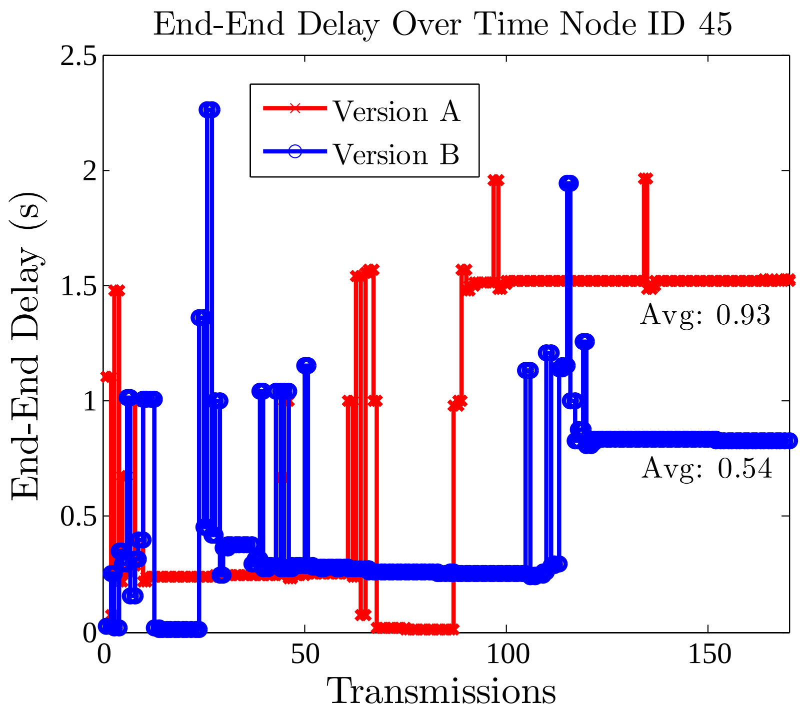


Figure 5.12: **End-to-End delay Node ID 45. 52 node deployment**

protocol.

Figure 5.11 shows how the topology of the network changes, when Version B of the protocol is in operation. In the upper section of 5.11, the workload of each node is plotted. This alternates between versions, illustrating the change in parental selection. The lower graph shows how the average number of hops per node changes from version to version. Figure 5.12 plots how the end-end latency for an arbitrarily chosen node changes over time for each version of the protocol. Node 45 achieves an average end-end latency of 0.54s using Version B and 0.93 using Version A.

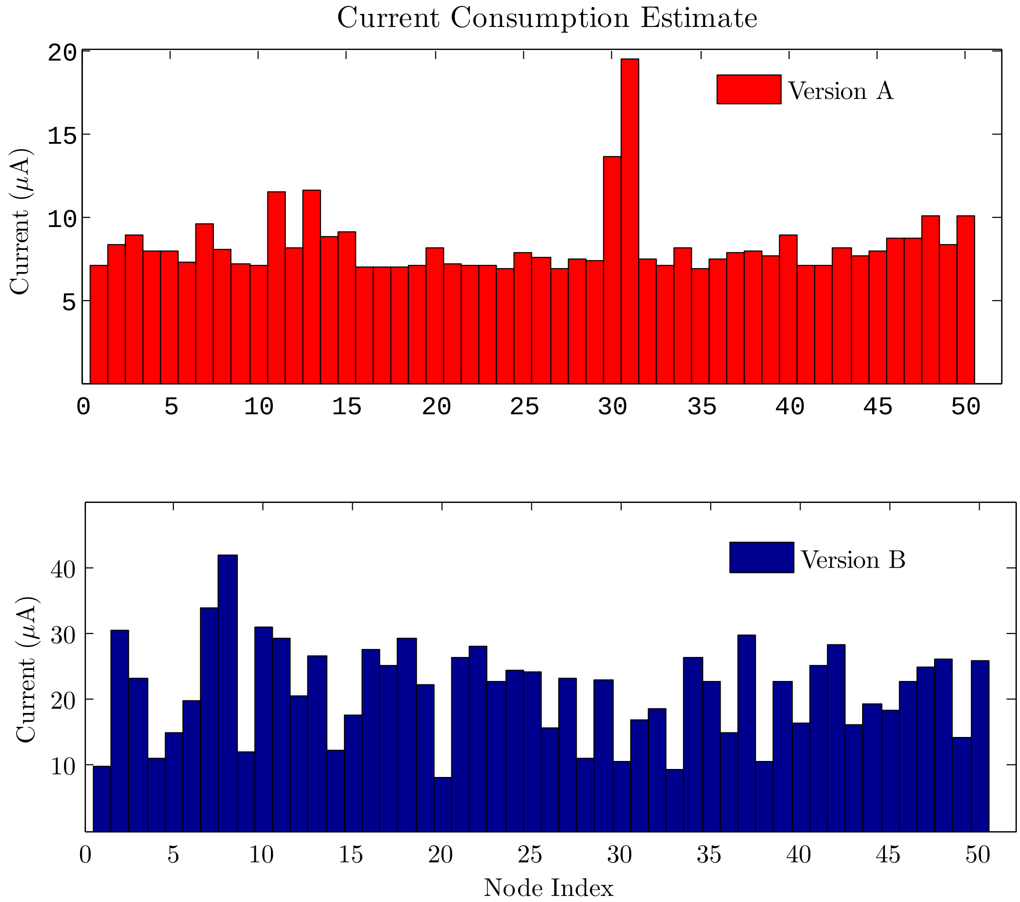


Figure 5.13: **Estimated Power Consumption of Networked Nodes**

Figure 5.13 shows the estimated average current consumption of the networked nodes over the duration of the deployment. It is based on the software radio duty-cycle estimation included in each radio transmission. The mean value for Version A is 8.3 μA with a standard deviation of 2 μA, the mean value for Version B is 20.1 μA with a standard deviation of 7.3 μA.

5.5 Discussion and Analysis of Results from Section 5.4

The goal of the new routing protocol presented in this chapter is to provide low end-to-end latency, to minimise the power consumption and to improve the end-to-end reliability. These factors influence each other and optimising for one has a detrimental impact on the others. The protocol is built around a customisable routing metric that allows factors weights to be easily modified for a different result. Two versions of the protocol were tested - reliability and low-power have higher priority in version A, while version B optimizes for E2E latency.

Version A of the protocol achieves an average reliability of 99.983%, Version B of the protocol achieves an average reliability a fraction lower at 99.58%. The drop in reliability seen in Version B can be explained by Figure 5.10. This graph shows how the average RSSI of chosen parent nodes decreases for Version B. Choosing parents with lower latency but weaker RSSI, results in lossier links. Version A of the protocol prefers stable links over low-latency links.

In terms of radio duty cycle, Version A achieves a duty-cycle of 0.207% that is 22% lower than that of Version B. This reduction in duty-cycle can be explained by the lack of periodic beacons and the more stable links provided by Version A, which optimises for reliability.

Interestingly Version A provides a very uniform distribution of the duty-cycle of all nodes under test, this would provide a more uniform lifetime for nodes in a deployment.

The improved latency performance of Version B makes it an attractive option considering that there is only a slight impact on the radio duty-cycle and reliability. The 47% reduction in average end-end latencies comes at a cost of only 0.4% in terms of reliability and a fractional increase of 0.058% on the duty-cycle (0.265-0.207).

One of the hypotheses of this work is that the network should provide 10 years of functionality while reporting data on approximately a 10 minute basis. In terms of the current consumption of the deployed nodes reporting every 4 minutes, Version A offers a lower average current of 8.3 μA versus 20.1 μA for Version B. Version A also provides a more evenly distributed workload and achieves a standard deviation of just 2 μA in the power consumption across 51 networked nodes, B manages 7.3 μA. To estimate the lifetime, the specific figures for the battery used in this work must be known. The capacity is 2,000 mAh and the self-discharge is quoted at 0.5% per year at room temperature. In 10 years the capacity of the battery will have dropped to 1900 mAh or 1800 mAh worst case scenario (double the self-discharge). An 1800 mAh battery would provide a theoretical lifetime of 24 years with an average current drain of 8.3 μA and a lifetime of 10 years with a drain of 20.1 μA. Both versions of the designed protocol are capable of providing the required 10 year battery life, Version A would have a much higher probability of achieving the 10 year goal if other worst case scenarios were considered (Temperature Variations etc).

For a node to always choose the optimum parent in terms of latency/ link quality and load balancing, it must be well informed on the states of its potential parents. Oscillator drift will affect various end-to-end latencies which potential parents offer, moreover multiple non-uniform oscillator drifts interacting over multiple hops make this difficult to predict.

A specific parent which offered the lowest latency path at a specific point in time, may no longer still offer the lowest latency route some time later. For this reason, nodes must periodically update their outlook using beacon transmissions. Version B did this by send beacons on a 30 minute basis, but further investigation would be required to find the optimum refresh interval. This optimum interval would provide the best trade-off between power consumption and latency performance.

5.6 Conclusions

This section highlights the main contributions and ideas presented in this chapter and the differences between this work and the current art. The main contributions of this work are summarised below:

- The routing protocol displayed an ultra low average current consumption figure in the region of 10 μA, as specified in the desired energy budget. This figure will help guarantee a 10 year battery life and improves upon the state of the art.
- As a result of a reliable underlying MAC layer and multiple failsafe mechanisms to improve quality of service, this work achieves excellent reliability. The average reliability of the 52 deployed nodes was 99.983%.

- Latency Aware Route Selection: This work achieves latency aware route selection by using a highly cross-layer approach, with an underlying MAC layer which enables one-hop latency measurements. One of the fundamental drawbacks with using low-power duty cycled WSN deployments, is the poor latency performance due to the underlying duty-cycled MAC operation. The routing protocol designed and implemented in this work, offers best effort low-latency operation, while respecting reliability constraints. The average end-to-end latency of all deployed nodes was 160 ms for Version B and 350 ms for Version A.
- Beacon free exchange of Routing Information: Routing Information is exchanged in payload acknowledge frames to avoid the use of power-hungry beacon transmissions. This allows for routing information to be transferred in a more energy efficient and real-time manner. Child nodes are now informed in real-time of changes to their parent's routing information, and update the score assigned to their parent after each successful transmission.

Chapter 6

Comparison against the State of the Art in a Real Deployment

This chapter describes a number of experiments which were carried out to compare and evaluate the performance of this work versus the existing art. A-MAC running over the CTP routing protocol was chosen as the standard to compare against [1].

Section 6.1 describes how this work was implemented on the TelosB hardware platform. In Section 6.3, the first of the comparative results are presented, wherein the packet delivery reliability was measured under contention. Section 6.2 shows the current profile of the basic MAC functions of this work on the TelosB platform, also included are some of those from A-MAC for comparison. Section 6.4 presents power consumption testing results from a controlled deployment. Results are presented for nodes in different positions in the network, power consumption is graphed against packet send interval. In Section 6.5 results from a side-by-side deployment are given. To finish this chapter a discussion of the results attained is given in Section 6.6 and Section 6.7 concludes the chapter.

6.1 Migration to the TelosB Platform

To carry out a fair and thorough comparison of the performance of this work against the current state of the art, the TelosB hardware was chosen as the

[1] At the SenSys 2013 Doctoral Colloquium, it was the opinion of the panel of experts that A-MAC-CTP is the standard to compare against

development platform for comparison. The TelosB platform is the best supported hardware platform in TinyOS, allowing for TinyOS developed protocols to be evaluated on it. With the majority of WSN research being carried out on the TelosB platform, this also allows for easier comparison against existing results reported in the literature. The TinyOS implementation of A-MAC is known as HotMAC [2].

The *IX-MAC* and LARP protocols for the 868MHz hardware presented in Chapters 3 and 5, were ported to work on the TelosB hardware. Code Composer Studio V.5 was used for all code development[3], code was written in C and was debugged using the standard USB MSP-FET430 programmer/ debugger [4]. The hardware designers of the TelosB platform connected the JTAG debug pins to test points on the PCB, this accelerated the debug and development process significantly. Figure 6.1 shows the TelosB platform connected to the FET debugger.

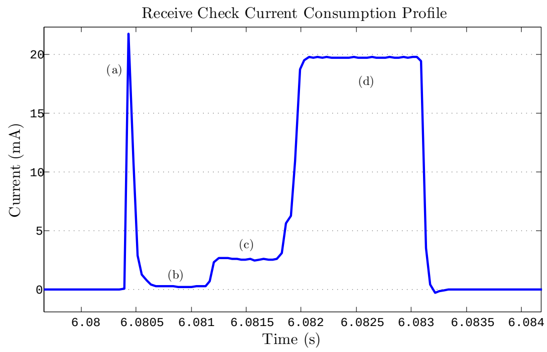
Figure 6.1: **TelosB hardware connected to the MSP-FET430 USB debugger**

[2]https://github.com/tyll/tinyos-2.x-contrib/tree/master/berkeley/hotmac
[3]http://www.ti.com/tool/ccstudio
[4]http://www.ti.com/tool/msp-fet430uif

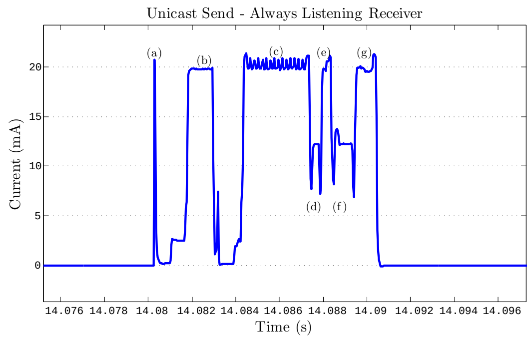
The two fundamental differences between the platforms are: 1) Different microcontroller (MSP430 on TelosB vs. PIC24F on 868 platform), 2) Different radio transceiver (CC2420 2.4 GHz vs. SX1211 868 MHz). There are also subtle differences in the software implementation due to incompatibilities between the two hardware platforms. Given below, is a list of the differences:

- Hardware address filtering is used on the 868 MHz transceiver, the transceiver rejects packets which are not destined for it. The 2.4 GHz implementation uses software address matching, the transceiver decodes all packets and the microcontroller decides which packets should be acted upon.
- In the 868 MHz implementation, the receiver waits for an address match interrupt to signal the beginning of a packet reception, the 2.4 GHz implementation uses the 802.15.4 Start of Frame Delimiter (SFD) signal pin to signal the beginning of a packet reception.
- Because of differences in the way interrupts are handled in the MSP430 processor [5], the 2.4 GHz version does not achieve the sleep current level of deep sleep ($\approx$ 10 μA) while performing delay before send, waiting for receivers to wake. Instead it achieves $\approx$ 200 μA.
- The TelosB platform contains additional ICs that provide USB programming capabilities, these hinder the minimum achievable sleep current for the platform. The multiplexer IC has a quiescent consumption of $\approx$ 8 μA and this dictates the minimum sleep current for the platform, this implementation achieves 10 μA sleep current for the TelosB platform.
- The 2.4 GHz transceiver of the TelosB platform has a bit rate of 250 kbps, this 2.5x greater than the bit rate of the 868 MHz transceiver. The 2.4 GHz transceiver can therefore achieve lower radio duty-cycles due to the increased bit rate and resultant reduced time required to transmit data packets.
- The MSP430's peripherals only support 8-bit wide operations and are not buffered. The PIC24F's peripherals support 8-bit and 16-bit wide operation and are buffered. The flexibility of the PCI24F's SPI peripheral allows for less processor intervention during data transfer between the microcontroller and the radio transceiver.

[5] Interrupt priorities are not modifiable and nesting is not supported



(a) Receive Check Current Profile, 2.8 ms Overall Duration: (a) Regulator enable and inrush current, (b) Regulator Stabilisation, (c) Oscillator Stabilisation, (d) 1.2 ms Listen Period.



(b) Unicast Send to Always on Neighbour: (a) Regulator Enable, (b) Receive Check, (c) Clear Channel Check, (d) Send RTS, (e) Receive CTS, (f) Send Payload, (g) Receive and Process payload ACK.

Figure 6.2: **MAC Primitives for TelosB Platform 1**

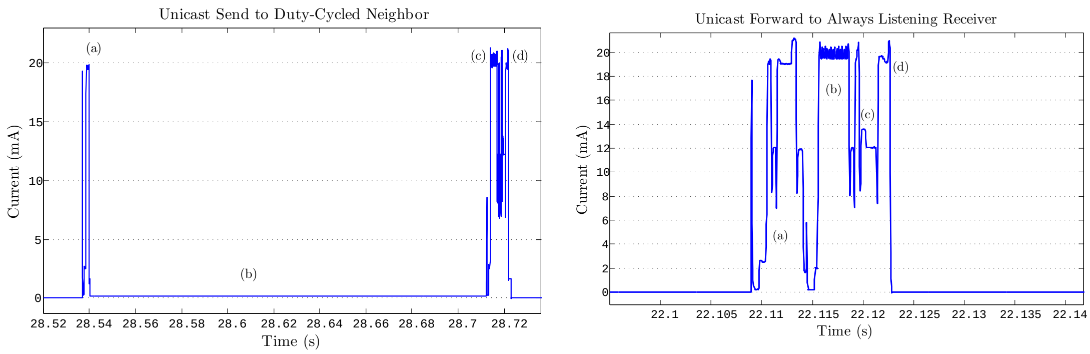
6.2 TelosB Current Consumption Profiles for *IX-MAC* Primitives

This section will present results from experiments which were carried out to measure the current consumption profile of MAC only functions of this work running on the TelosB platform. The current consumption profile was measured using a calibrated DC power analyser N6705B, the N6705B supplies the device under test with a 3 V supply and measures the current consumption with 10 kHz sampling frequency and ±10 nA resolution. Light Emitting Diodes (LEDs)s were disabled during the experiment, as these consume a few mA per LED on the TelosB platform.

The following MAC primitives were measured: Receive Check (1.2 ms in length), Unicast Send to always listening Receiver, Unicast Send to duty-cycled Receiver, Unicast Forward to always listening Receiver, Broadcast.

Figure 6.2, depicts two of the primitive MAC functions, Figure 6.2a shows the current consumption profile during a receive check operation and Figure 6.2b shows the profile during a unicast send to an always listening neighbour (sink). Both figures have the individual phases labeled and the transmit power was changed to -10 dBm so a clear step can be observed between RX and TX modes.

Figure 6.3, depicts two different primitive MAC functions, Figure 6.3a shows the current consumption profile during a unicast send to a duty-cycled



(a) Unicast Send to duty-cycled neighbour: (a) Receive check, (b) Low-Power delay for learned neighbour offset of 170 ms, (c) Clear Channel Check, (d) RTS/ CTS and payload transmission.

(b) Receive Packet and Forward to always listening neighbour : (a) RTS/ CTS and payload reception (b) Clear Channel Check, (c) CTS received send payload, (d) Receive and process payload ACK.

Figure 6.3: **MAC Primitives for TelosB Platform 2**

neighbour (whose offset is known), Figure 6.3b shows the profile during a unicast packet forward to an always listening neighbour (sink). Both figures have the individual phases labeled and the transmit power was changed to -10 dBm so a clear step can be observed between RX and TX modes.

Figure 6.4 depicts the current consumption profile during a broadcast, while performing RTS/ CTS the current consumption changes from the -10 dBm TX current level of 10 mA to the RX current level of 20 mA. Overall length is ≈ T_W + 40 ms.

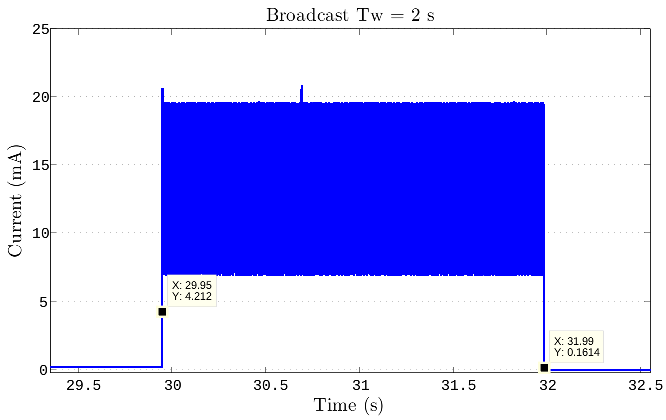


Figure 6.4: **Broadcast Current Profile**

Figure 6.5 depicts a direct comparison on the TelosB of *IX-MAC's* receive check current profile, versus A-MAC's probe. The overall length of the receive check which includes the voltage regulator stabilisation and oscillator stabilisation phase, is 8 ms for A-MAC and 2 ms for this work.

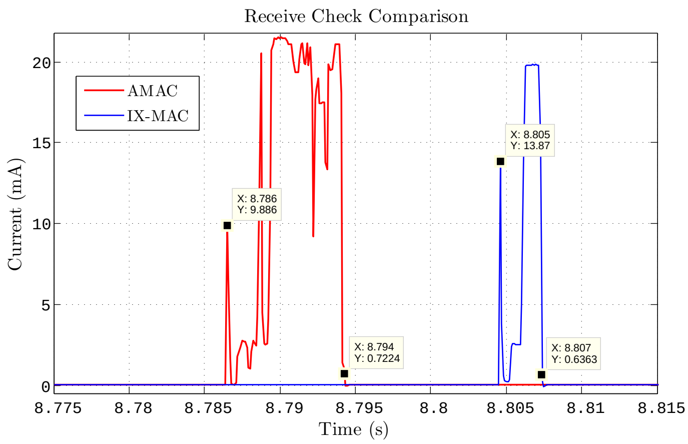


Figure 6.5: **IX-MAC vs A-MAC Receive Check Direct Comparison on TelosB**

6.3 Emulating Performance at Scale, Reliability under Contention

A similar experiment to that which is described in Section 4.6.2 was performed using the TelosB hardware. Its purpose was to measure the performance of the protocols under test when network contention occurs. To emulate a dense network and to create channel contention, multiple nodes were configured to send unicast packets to a single duty-cycled receiver. Nodes attempted to transmit for one T_W interval per transmission attempt and no resends were performed.

Sending nodes were configured to send 1000 packets at a rate of 1pkt/s. This equates to an overall experiment length of 16 minutes and 40 seconds. The receiver kept a log of packets received from all sending nodes. At the end of the test the stored values in the receiver could either be read from the in-circuit debug interface or printed to a terminal program over USB. A receive check interval of 100 milliseconds was used at the receiving node. In theory, with this receive check rate the receive should be able to receive 10 packets per second (1/100ms). Equating this back to the senders, 10 senders sending packets at a rate of 1pkt/s should be able to deliver packets to the receiver before channel contention occurs. It is not possible to deliver more than 10 packets/ second to a receiver that only listens ten times per second.

A TinyOS application was written to perform the same test on the A-MAC protocol. The results of the experiment are shown in Figure 6.6.

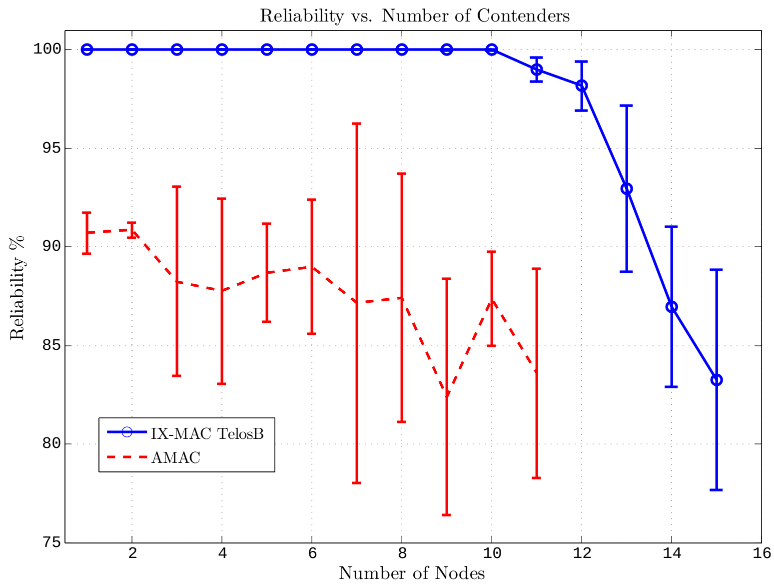


Figure 6.6: **Reliability under Contention TelosB**

IX-MAC on the TelosB platform performs as predicted, up as far as 8 eight sender it provides 100% packet delivery, for 9 and 10 senders it achieves 99.98% and 99.99% respectively. At 11 concurrent senders there is a visible decrease to 98.9%, from 11 onwards there is a steep decrease in packet delivery reliability, 15 senders manage to deliver 83% of packets. The same experiment in TinyOS for A-MAC turned out to be a non-trivial task. When nodes were programmed to send 1000 packets and stop sending after 1000 packets had been reached, unpredictable behaviour resulted in more than 1000 packets being received from certain nodes. Finally nodes were configured to send 1 packet per second and stopped after 100 seconds. This approach solved the problem of receiving more packets than were sent, but it created another problem. Now instead of receiving 100 packets in the 100 second period, the receiver received approximately 90.

Figure 6.6 shows that 1 and 2 senders achieve above 90% packet delivery, thereafter the packet delivery reliability decreases. At 11 senders, the packet delivery reliability is 83%. The trend shows a clear decrease in reliability under moderate contention for the A-MAC protocol.

6.4 TelosB Average Current Consumption for Controlled Network

This section presents results from experiments which were carried out to measure the current consumption of networked nodes running this work on the TelosB platform. In a multi-hop topology the current consumption of the nodes will vary depending on their position in the tree topology. Sink neighbours will usually have the lowest current consumption, as these nodes are sending packets to an always listening sink. Leaf nodes must only send their own packets to their parent and must not forward packets for neighbouring nodes, they achieve current consumption that is slightly higher than sink neighbours, it is slightly higher because leaf nodes must send their packets to duty-cycled neighbours. Routing nodes must receive and acknowledge incoming packets and forward these to receiver duty-cycled neighbours, this group of nodes reach the highest current consumption of the three network node types.

For this experiment nodes were placed in close proximity, this was done in attempt to remove the overhead caused by link failures seen in real deployments and to measure the cost of successfully delivering packets [6]. The following network functions were measured: Leaf node sending (to duty-cycled neighbour), Sink neighbour sending its own packets, Sink neighbour forwarding packets for one child, and Routing node forwarding for one child node. Nodes were configured to send packets at varying intervals, between 10 seconds and 5 minutes. To create graphs which show typical errors/ variations in the average current consumption, tests were performed for a minimum of six times the data send interval.

Figure 6.7 shows the results of the experiments which were carried out. It shows four different traces, Sink Neighbour sending only its own packets, Sink Neighbour forwarding for one node, Leaf Node sending to duty-cycled parent, and Routing node forwarding for one node to a duty-cycled parent. When the data send interval is 10 seconds, the Sink Neighbour achieves an average current drain of 36 μA, the Routing node achieves the highest of the four at 62 μA. Moving to a data send interval of one packet every 5 minutes, Sink Neighbour and Sink Neighbour forwarding for one node achieve just under 25 μA. The Leaf node and Routing node achieve an average current of approximately 26-27 μA. At this point when packets are being sent every 5 minutes, the largest

[6] Link failures are costly and typically trigger the transmission of beacons

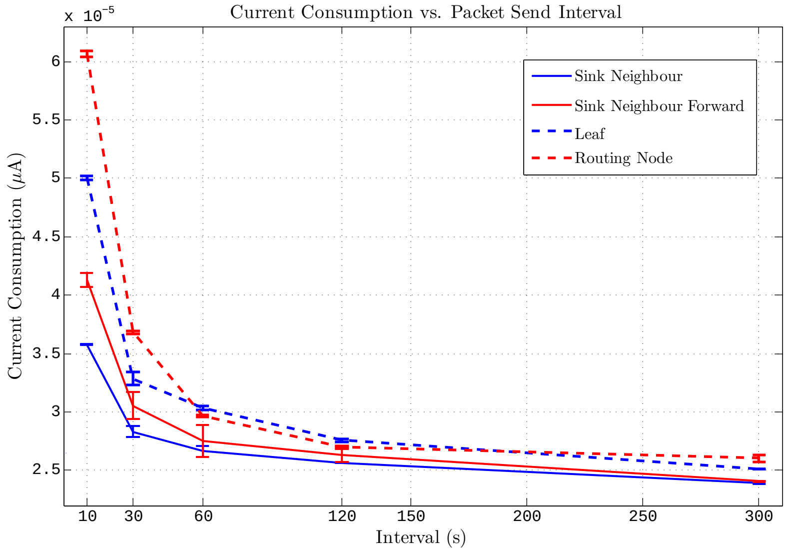


Figure 6.7: **Current Consumption of Networked Nodes vs Send Interval**

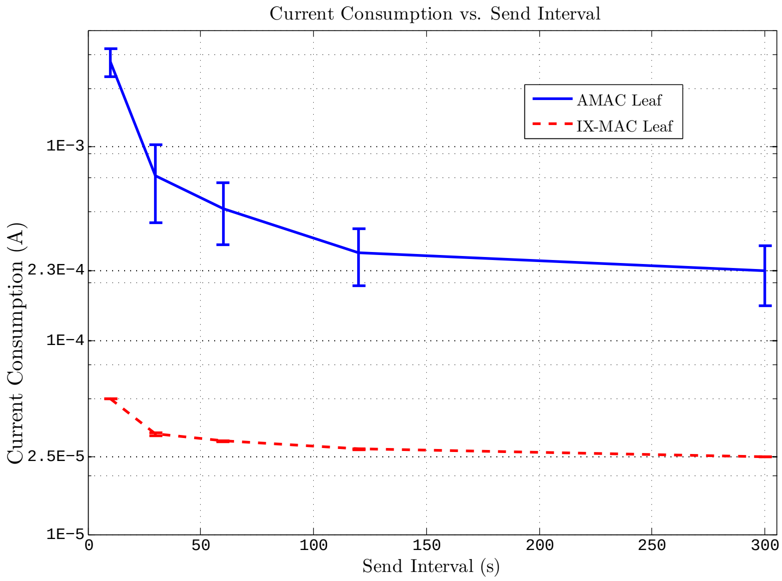


Figure 6.8: **Current Consumption of Networked Nodes A-MAC, IX-MAC**

contributor to the average current drain is the receive check. This is due to the extremely low TX duty cycle offered by *IX-MAC*.

A similar experiment was performed to analyze the current consumption of A-MAC over CTP. It was not possible to setup an identical experiment in A-MAC, because A-MAC does not support an always listening network sink, the network sink must perform receive checks at the same rate as networked nodes [7]. Instead the current consumption of a Sink Neighbour was measured

[7]Due to the receiver initiated nature of A-MAC, it does not support always listening

across varying data send intervals (equivalent to Leaf Node in *IX-MAC* testing). The current consumptions are comparable because both nodes are sending packets to a duty-cycled receiver.

Figure 6.8 depicts the results of this experiment, also included in Figure 6.8 for comparison, is the Leaf node trace of *IX-MAC*. A-MAC sending every 10 seconds equates to an average current drain of 2.7 mA, *IX-MAC* manages 50 μA, a 54 times reduction in power consumption. At the last measured interval of 300 seconds, A-MAC achieves 230 μA and *IX-MAC* 25 μA, almost an order of magnitude lower.

6.5 Long term Performance in a Deployed Network

A network of 24 nodes was deployed to compare the performance of A-MAC running CTP and this work. Identical physical layer properties were used, transmit power was set to -10 dBm and the protocols operated on Channel 26 [8]. The final code size of this work was 8 kB and 22 kB for A-MAC with CTP. 12 nodes were deployed running A-MAC and 12 using this work, pairs of nodes were placed at identical locations. All nodes were configured to send packets every 1 minute. Figure 6.9 shows the two network sinks connected via USB to a PC.

The nodes were placed at approximately 1.5 m height and distributed around an office on a single level. Nodes 1-6 were placed in close proximity to the network sink (10-20 m line of sight). The remaining 5 nodes (7-11) were placed in locations where it was known that multi-hop communication was necessitated to reach the network sink. In this experiment, the goal was to measure the reliability and power consumption of a more dynamic network in a busy office environment. In this scenario, link failures were more likely to occur due to people and objects being moved around on a regular basis. The reliability of the deployment was measured by analysing the timely arrival of the node's 60 second interval packets. If a packet was not received within 70 seconds of the last packet, a packet was deemed to have been lost. This is a fair measure

nodes, the receive check interval of the sink must also match that of the networked nodes

[8]Setting both protocols to operate on the same channel helped to create some in-band interference

Figure 6.9: **Identical Node Locations**

because the networking protocol should still be able to deliver a packet within this time-frame even if a link failure occurred.

The power consumption was measured by recording the power consumption of one node from each protocol at identical locations in the network using the precision DC power analyzer. The location chosen was one where the nodes were out of range of the network sink. The time interval over which power consumption was measured was 9 hours (from 08:00 to 17:00) and both deployments operated for 4 days. This period of time was chosen as it is busiest time in the office, a period when link failures are most likely to occur due to moving objects.

The results from the deployments are given in Table 6.1. The power consumption of nodes 11 from A-MAC and this work was measured. Node 11 in A-MAC achieves an average current drain of 473 μA with a standard deviation of 14 μA, Node 11 in this work achieves an average current drain of 47 μA with a standard deviation of 6 μA. The standard deviation was measured by dividing the 9-hour logging period into 6 x 1.5 hour sub bins.

A similar network was deployed using the 868 MHz platform to compare power

Table 6.1: **Deployment Results**

A-MAC	**Reliability**	**Hops**	**IX-MAC**	**Reliability**	**Hops**
1	99.91	1.92	**1**	99.95	1
2	99.58	1	**2**	100	1
3	98.61	1	**3**	99.93	1
4	99.91	1.07	**4**	100	1
5	99.75	1.56	**5**	99.97	1
6	99.5	1	**6**	99.98	1
7	99.75	1.06	**7**	99.97	1.9
8	99.18	1	**8**	99.86	1.95
9	99.41	2	**9**	99.85	2.25
10	99.91	2	**10**	99.76	3.14
11	99.38	3	**11**	99.97	2.33
Average	99.53(σ0.39)	1.51	**Average**	99.93(σ0.07)	1.59

consumption figures. Identical parameters were used for testing, T_W was 2 seconds, the send interval was 60 seconds and 12 nodes were deployed. It was not possible to use identical locations to the 2.4 GHz deployment due to the inherent increased transmission range of 868 MHz transceivers compared to 2.4 GHz.

In Section 5.4 results from a 52 node deployment were given, power consumption results were presented by analysing the radio duty-cycle of deployed nodes. In this experiment the device under test was connected to the DC Power Analyser, powered at 2.8 V, and data was logged for 9 hours as in previous experiments. The device under test was not a sink neighbour, but a node further removed from the network sink, this means its transmissions were directed to duty-cycled neighbours. Over the 9-hour period the 868 MHz node achieved an average current consumption of 11.7 μA with a standard deviation of just 200 nA. It achieved an average reliability of 99.96 %.

6.6 Discussion

Both protocols achieved an average reliability above 99.5 % for the 11 deployed nodes of each protocol. A-MAC achieves an average reliability of 99.53% while this work achieves a higher average of 99.93%. Considering that both protocols operated in parallel on the same channel, these reliability figures are good. The most interesting result is the power consumption at which this work achieves a reliability of greater than 99.9%. This demonstrates its efficient operation and reliable underlying MAC layer.

The observed measured power consumption agrees with the values presented in Figures 6.7 and 6.8. According to Figure 6.7 a routing or leaf node should achieve in the region of 30-35 μA with a send interval of 60 seconds. The deployment measured 47 μA, the slight increase can be explained by the fact that Node 11 was occasionally forwarding for 2 child nodes. These figures include the 10 μA overhead of the TelosB's sleep current.

According to Figure 6.8, a node running A-MAC with a send interval of 60 seconds should achieve a current consumption in the region of 460 μA, the actual deployment measured 473 μA. For the 868 MHz platform Figure 4.10 estimates a current consumption of approximately 20 μA with a send interval of 60 seconds, the deployment measured 11.7 μA, the 8.3 μA discrepancy is explained by the fact that Figure 4.10 used a T_W of 0.5 s, the deployment used 2 s.

An interesting quantitative comparison is a Figure of Merit (FOM) to compare the ratio of the achieved reliability to the power consumption. The key to creating long lifetime and sustainable WSNs is to balance the requirements of low-power consumption and high reliability, the equation below 6.1 attempts to compare the trade-off between energy and reliability of both systems, creating a Figure of Merit which factors in both power consumption and reliability.

$$FigureofMerit = Current(\mu A) \times \sqrt{100 - Reliability} \quad (6.1)$$

The square root operation in Equation 6.1 attempts to accentuate higher achieved reliability values. The lower the Figure of Merit the better the performance. Plugging the measured numbers for Node 11 into Equation 6.1, A-MAC achieves a score of 372 while this work achieves 8.

The average hop count for both protocols is very similar at 1.51 for A-MAC and 1.59 for this work. The nodes which were placed far from the network sink (7-11) from this work exhibit more dynamic operation, this is reflected in the average number of hops achieved. The non integer numbers suggest a decent proportion of the transmissions traversed different numbers of hops before reaching the network sink. The measured values for A-MAC showed less variation. This indicates that this work was more dynamic in seeking out more stable links, and this is reflected in the increase in reliability.

The 868 MHz implementation features nodes with individually calibrated 32.768 kHz oscillators. This improves the efficiency of the neighbour learning scheme. The software implementation on the 868 MHz hardware is also slightly more refined (more time was spent optimising MAC timings etc). It outperforms both 2.4 GHz protocols in terms of average current consumption. The average current consumption of the 868 MHz hardware outperforms the same protocol operating on the TelosB platform by a factor of 4 (47/11.7), it outperforms A-MAC by a factor of 40 (473/11.7). If the additional sleep current of the TelosB platform in factored in, this improvement factor is reduced to 3.33 and 39.7.

6.7 Conclusion

This chapter described the implementation and evaluation of a low-power, high reliability WSN protocol stack on the TelosB hardware. It was developed as part of this thesis. It was shown to outperform the current state of the art in a like for like comparison in terms of both power consumption and reliability. Most importantly it provided an excellent trade-off between power consumption and reliability compared to the state of the art. It was also shown to be a much more compact solution in terms of memory footprint, it carried out the same task more efficiently using 8 kB compared to 22 kB.

Chapter 7

Summary & Conclusion

This thesis took a complete application-driven co-design approach, starting with the hardware and extending into the protocol-space. This involved the design, analysis and evaluation of a novel MAC protocol (called *IX-MAC*) and networking protocol (called LARP) for next generation WSN applications. A clean slate approach was adopted to help the system meet the required levels of performance specified in Section 1.5. Here it was specified that the multi-hop WSN solution should achieve an average reliability of 99.9% at a power consumption level that can sustain 10 years of operation from a 2,000 mAh capacity battery.

An ultra low-power and low-cost WSN hardware platform was designed and manufactured as part of this work, it is called the Firemote. Its hardware components were selected to minimise power consumption, maximise communication reliability and to help meet the required system performance characteristics.

A number of MAC protocols were studied and analysed in Section 3.2. It was evident that to create an ultra low-power MAC protocol, it would be necessary to reduce the energy per transmitted packet using some element of time synchronisation. However, fully synchronous protocols such as TDMA or flooding techniques require additional overhead traffic to maintain time synchronisation, therefore it was decided not to use a fully synchronous approach. Instead a hybrid lightweight approach to synchronous operation was designed. The resultant *IX-MAC* protocol uses an improved loose time synchronisation scheme to reduce the energy required to transmit unicast data packets to receiver duty-cycled neighbours. It features a number of other power saving and reliability enhancing features designed to help meet the desired system performance requirements.

IX-MAC differs from other schedule learning systems reported in the literature, in that it does not require exchange of scheduling information. Its optimised RTS/ CTS wakeup preamble can also be adjusted precisely and efficiently to account for oscillator drift. The collision detection and avoidance algorithm that was developed provides excellent reliability under contention and outperforms the state of the art in terms of packet delivery reliability in dense networks. The novel semi-synchronous operation enables neighbour latency measurement at the MAC layer, these values are propagated through the network stack to the routing protocol. The various power saving features implemented in *IX-MAC* result in large improvements over the state of the art in terms of power consumption, particularly when sensor readings must be reported frequently.

A lightweight cross-layer routing protocol was developed to enable multi-hop networking capabilities. Its goal was to seamlessly sit on top of the underlying MAC layer and to provide reliable end-to-end packet delivery for many to one traffic patterns, as well as latency aware route selection. It achieves beacon free operation by nodes efficiently exchanging routing information in payload acknowledge frames. This technique allows nodes to attain up-to-date knowledge of their neighbourhood with a single broadcast communication and removes the need for periodic beacon transmissions. It is the first routing protocol to achieve latency aware route selection, this is achieved by exchange of information between MAC and routing layers. It contains multiple reliability enhancing features such as loop-back avoidance and it stores multiple potential routing options to improve performance in lossy networks.

In Chapters 5 and 6 results from deployments are given. It was shown that the routing protocol provides the desired overall reliability of 99.9%. Importantly, as shown in Sections 5.4 and 6.5, the routing protocol exhibits lightweight operation and does not impact profoundly on the power consumption.

Using the custom built 868 MHz hardware, the overall system achieves the required performance characteristics of achieving 99.9 % reliability on a current consumption budget of $\approx 10\ \mu$A. The 2.4 GHz implementation on the TelosB platform achieves the required reliability performance but the current consumption is above the level required to provide 10 years of operation. The 20 mA receive mode current of the TelosB hardware would require a longer receive check interval than 2 s to be used to reach the desired current consumption level, the sleep mode current would also need to be reduced. The TelosB test network in Chapter 6 also used a send interval of 1 minute, whereas in reality a 5-10 minute send interval would suffice.

The combination of the hardware platform designed in Chapter 2 and the software described in Chapters 3 and 5 balances the requirement of high reliability and ultra low-power operation. This thesis describes a set of techniques which can easily be adapted to other WSN applications where long lifetime from battery powered sources is required.

7.1 Contribution to Knowledge

- This work contributes to the existing knowledge in WSN technology by describing the operation of a lightweight semi-synchronous system.
- This work showed that semi-synchronous operation can be achieved without nodes sharing a common notion of time or without the need for nodes to exchange scheduling information.
- The extensive experimental evaluation proves it outperforms the state of the art in terms of power consumption and reliability.
- Using the neighbour schedule learning technique described, this work successfully decouples the energy required to transmit data from the receive check interval.
- It showed that layer 2 receive check mechanisms can perform well in terms of power consumption when a careful design is implemented.
- It successfully combines a number of power saving and reliability enhancing features into one communication protocol.
- This work achieves latency aware route selection and describes precisely how it can be achieved.
- The routing protocol describes how exchange of routing information can be achieved without the transmission of beacons.
- This thesis proves that an ultra reliable multi-hop WSN can provide a battery life which exceeds 10 years.

7.2 Future Work

There are a number of areas where this work could be expanded and improved, these are listed below:

- **Security**: Before this WSN solution could be used in an industrial application, a layer of security would need to be added. With many of the radio transceiver IC's capable of supporting hardware security encryption, this would be a feasible task. It may have a negative impact on the energy efficiency of the protocol stack and this would make for interesting future work.
- **Regulatory Certification**: This process would need to be carried out before this system could be integrated into a commercial home security application. This body of work would involve ensuring the system meets the specifications described in any of the relevant standards.
- **Application**: This work was designed specifically for periodic wireless monitoring applications. An interesting topic would be to find specific applications where the system that was designed could find up-take.
- **Bringing WSN Technology a Step Further**: With the results achieved in this work, there are few reasons why this technology cannot be used to make WSN usage more widespread. This would involve engaging with and convincing people beyond academia that this technology can be used for a multitude of sensing applications.

Bibliography

[1] Crossbow telosb. http://www.xbow.com/.

[2] Fire detection and fire alarm systems. control and indicating equipment, 1997.

[3] M.L. Abd-El-Barr, M.A.M. Youssef, and M.M. Al-Otaibi. Wireless sensor networks - part i: topology and design issues. In *Electrical and Computer Engineering, 2005. Canadian Conference on*, pages 1165–1168, May 2005.

[4] Marcus Autenrieth and Hannes Frey. PaderMAC: A low-power, low-latency MAC layer with opportunistic forwarding support for wireless sensor networks. In *Proceedings of the 10th international conference on Ad-hoc, mobile, and wireless networks*, pages 117–130, Paderborn, Germany, 2011. Springer-Verlag.

[5] Andre Barroso, Utz Roedig, and Cormac Sreenan. μ-mac: an energy-efficient medium access control for wireless sensor networks. In *Wireless Sensor Networks, 2005. Proceeedings of the Second European Workshop on*, pages 70–80. IEEE, 2005.

[6] A. Bildea, O. Alphand, F. Rousseau, and A. Duda. Link quality metrics in large scale indoor wireless sensor networks. In *Personal Indoor and Mobile Radio Communications (PIMRC), 2013 IEEE 24th International Symposium on*, pages 1888–1892, Sept 2013.

[7] D. Boyle, M. Magno, B. O'Flynn, D. Brunelli, E. Popovici, and L. Benini. Towards persistent structural health monitoring through sustainable wireless sensor networks. In *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2011 Seventh International Conference on*, pages 323–328, Dec 2011.

[8] Michael Buettner, Gary V. Yee, Eric Anderson, and Richard Han. X-MAC:

A short preamble MAC protocol for duty-cycled wireless sensor networks. SenSys '06, 2006.

[9] Nicolas Burri, Pascal von Rickenbach, and Roger Wattenhofer. Dozer: ultra-low power data gathering in sensor networks. In *Proceedings of the 6th international conference on Information processing in sensor networks*. ACM, 2007.

[10] K. Casier, L. Tytgat, S. Verbrugge, M. Pickavet, and I. Moerman. Building the business case for wireless sensors in a factory setting. In *FITCE Congress (FITCE), 2011 50th*, pages 1–6, Aug 2011.

[11] Feng Chen, R. German, and F. Dressler. Towards ieee 802.15.4e: A study of performance aspects. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on*, pages 68–73, March 2010.

[12] Ugo Maria Colesanti, Silvia Santini, and Andrea Vitaletti. Dissense: An adaptive ultralow-power communication protocol for wireless sensor networks. In *Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011 International Conference on*, pages 1–10. IEEE, 2011.

[13] Luiz H. A. Correia, Daniel F. Macedo, Aldri L. dos Santos, Antonio A. F. Loureiro, and José Marcos S. Nogueira. "Transmission power control techniques for wireless sensor networks". *Comput. Netw.*, 51(17):4765–4779, December 2007.

[14] Geoff Coulson, Barry Porter, Ioannis Chatzigiannakis, Christos Koninis, Stefan Fischer, Dennis Pfisterer, Daniel Bimschas, Torsten Braun, Philipp Hurni, Markus Anwander, Gerald Wagenknecht, Sándor P. Fekete, Alexander Kröller, and Tobias Baumgartner. Flexible experimentation in wireless sensor networks. *Commun. ACM*, 55(1):82–90, January 2012.

[15] Douglas SJ De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A high-throughput path metric for multi-hop wireless routing. *Wireless Networks*, 11(4):419–434, 2005.

[16] Manjunath Doddavenkatappa and Mun Choon Chan. Aal an experience of building indriya. *National University of Singapore*, 2009.

[17] Adam Dunkels. The contikimac radio duty cycling protocol. Technical report, ISSN, 2011.

[18] Simon Duquennoy, Olaf Landsiedel, and Thiemo Voigt. Let the tree bloom: Scalable opportunistic routing with orpl. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, SenSys '13, pages 2:1–2:14, New York, NY, USA, 2013. ACM.

[19] Prabal Dutta, Stephen Dawson-Haggerty, Yin Chen, Chieh-Jan Mike Liang, and Andreas Terzis. Design and Evaluation of a Versatile and Efficient Receiver-initiated link layer for Low-Power Wireless. SenSys '10, 2010.

[20] Amre El-Hoiydi and Jean-Dominique Decotignie. Low power downlink mac protocols for infrastructure wireless sensor networks. *Mob. Netw. Appl.*, 10(5), October 2005.

[21] Sinem Coleri Ergen and Pravin Varaiya. Pedamacs: Power efficient and delay aware medium access protocol for sensor networks. *IEEE Transactions on Mobile Computing*, 5(7):920–930, July 2006.

[22] Emre Ertin, Anish Arora, Rajiv Ramnath, Mikhail Nesterenko, Vinayak Naik, Sandip Bapat, Vinod Kulathumani, Mukundan Sridharan, Hongwei Zhang, and Hui Cao. Kansei: A testbed for sensing at scale. In *IN PROCEEDINGS OF THE 4TH SYMPOSIUM ON INFORMATION PROCESSING IN SENSOR NETWORKS (IPSN/SPOTS TRACK*, pages 399–406. ACM Press, 2006.

[23] Soledad Escolar, Stefano Chessa, and Jesús Carretero. Energy management in solar cells powered wireless sensor networks for quality of service optimization. *Personal Ubiquitous Comput.*, 18(2):449–464, February 2014.

[24] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh. Efficient network flooding and time synchronization with glossy. In *Information Processing in Sensor Networks (IPSN), 2011 10th International Conference on*, pages 73–84, 2011.

[25] Federico Ferrari, Marco Zimmerling, Luca Mottola, and Lothar Thiele. Low-power wireless bus. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, SenSys '12, pages 1–14, New York, NY, USA, 2012. ACM.

[26] Deepak Ganesan, Bhaskar Krishnamachari, Alec Woo, David Culler, Deborah Estrin, and Stephen Wicker. Complex behavior at scale: An experimental study of low-power wireless sensor networks. Technical report, Citeseer, 2002.

[27] Omprakash Gnawali, Rodrigo Fonseca, Kyle Jamieson, Maria Kazandjieva, David Moss, and Philip Levis. Ctp: An efficient, robust, and reliable collection tree protocol for wireless sensor networks. *ACM Trans. Sen. Netw.*, 10(1):16:1–16:49, December 2013.

[28] Omprakash Gnawali, Rodrigo Fonseca, Kyle Jamieson, David Moss, and Philip Levis. Collection Tree Protocol. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys'09)*, November 2009.

[29] Carles Gomez, Antoni Boix, and Josep Paradells. Impact of lqi-based routing metrics on the performance of a one-to-one routing protocol for ieee 802.15.4 multihop networks. *EURASIP Journal on Wireless Communications and Networking*, 2010(1):205407, 2010.

[30] Yu Gu, Liang He, Ting Zhu, and Tian He. Achieving energy-synchronized communication in energy-harvesting wireless sensor networks. *ACM Trans. Embed. Comput. Syst.*, 13(2s):68:1–68:26, January 2014.

[31] Andrew Hagedorn, David Starobinski, and Ari Trachtenberg. Rateless deluge: Over-the-air programming of wireless sensor networks using random linear codes. In *Proceedings of the 7th International Conference on Information Processing in Sensor Networks*, IPSN '08, pages 457–466, Washington, DC, USA, 2008. IEEE Computer Society.

[32] Vlado Handziski, A Köpke, Andreas Willig, and Adam Wolisz. Twist: A scalable and reconfigurable wireless sensor network testbed for indoor deployments. *Technical University Berlin, TKN Technical Report TKN-05-008*, 2005.

[33] Mark Hempstead, Michael J Lyons, David Brooks, and Gu-Yeon Wei. Survey of hardware systems for wireless sensor networks. *Journal of Low Power Electronics*, 4(1):11–20, 2008.

[34] http://www.tinynode.com/?q=product/tinynode584/tn 584-868. *Tinynode 584 User Guide.*

[35] J.W. Hui and D.E. Culler. Extending ip to low-power, wireless personal area networks. *Internet Computing, IEEE*, 12(4):37–45, July 2008.

[36] J.W. Hui and D.E. Culler. Ipv6 in low-power wireless networks. *Proceedings of the IEEE*, 98(11):1865–1878, Nov 2010.

[37] Philipp Hurni, Torsten Braun, and Markus Anwander. Evaluation of WiseMAC and Extensions on Wireless Sensor Nodes. *Telecommunication Systems*, 43(1-2):49–58, 2010.

[38] Jaein Jeong, D. Culler, and Jae-Hyuk Oh. "Empirical Analysis of Transmission Power Control Algorithms for Wireless Sensor Networks". In *Networked Sensing Systems, 2007. INSS '07. Fourth International Conference on*, pages 27–34, 2007.

[39] N. Joehl, C. Dehollaini, P. Favre, P. Deval, and M. Declerq. A low-power 1-ghz super-regenerative transceiver with time-shared pll control. *Solid-State Circuits, IEEE Journal of*, 36(7):1025–1031, 2001.

[40] M. Johnson, M. Healy, P. van de Ven, M.J. Hayes, J. Nelson, T. Newe, and E. Lewis. A comparative review of wireless sensor network mote technologies. In *Sensors, 2009 IEEE*, pages 1439–1442, Oct 2009.

[41] R.K. Jurgen. Coming from detroit: Networks on wheels: Multiplexed wiring in new models promises cost-effective control and accurate diagnostics, as well as fewer wires and less redundancy. *Spectrum, IEEE*, 23(6):53–59, June 1986.

[42] Aman Kansal, Jason Hsu, Sadaf Zahedi, and Mani B. Srivastava. Power management in energy harvesting sensor networks. *ACM Trans. Embed. Comput. Syst.*, 6(4), September 2007.

[43] Lakshman Krishnamurthy, Robert Adler, Phil Buonadonna, Jasmeet Chhabra, Mick Flanigan, Nandakishore Kushalnagar, Lama Nachman, and Mark Yarvis. Design and deployment of industrial sensor networks: experiences from a semiconductor plant and the north sea. In *Proceedings of the 3rd international conference on Embedded networked sensor systems*, SenSys '05, pages 64–75, New York, NY, USA, 2005. ACM.

[44] Olaf Landsiedel, Euhanna Ghadimi, Simon Duquennoy, and Mikael Johansson. Low power, low delay: opportunistic routing meets duty cycling. In *Proceedings of the 11th international conference on Information Processing in Sensor Networks*, pages 185–196, Beijing, China, 2012. ACM.

[45] Koen Langendoen, Aline Baggio, and Otto Visser. Murphy loves potatoes: Experiences from a pilot sensor network deployment in precision agriculture. In *Proceedings of the 20th International Conference on Parallel*

and Distributed Processing, IPDPS'06, pages 174–174, Washington, DC, USA, 2006. IEEE Computer Society.

[46] Koen Langendoen and Andreas Meier. Analyzing MAC protocols for low data-rate applications. *ACM Trans. Sen. Netw.*, 7(2):1–40, 2010.

[47] Tomas Lennvall, Stefan Svensson, and Fredrik Hekland. A comparison of wirelesshart and zigbee for industrial applications. In *IEEE International Workshop on Factory Communication Systems*, volume 2008, pages 85–88, 2008.

[48] Philip Levis, Neil Patel, David Culler, and Scott Shenker. Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks. In *In Proceedings of the First USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI*, pages 15–28, 2004.

[49] Renjun Li, Chu Liu, and Feng Luo. A design for automotive can bus monitoring system. In *Vehicle Power and Propulsion Conference, 2008. VPPC '08. IEEE*, pages 1–5, Sept 2008.

[50] Yuan Li, Wei Ye, and John Heidemann. Energy and latency control in low duty cycle mac protocols. In *Proceedings of IEEE Wireless Communication and Networking Conference*, March 2005.

[51] Shan Lin, Jingbin Zhang, Gang Zhou, Lin Gu, John A. Stankovic, and Tian He. Atpc: adaptive transmission power control for wireless sensor networks. In *Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 223–236, Boulder, Colorado, USA, 2006. ACM.

[52] Sha Liu, Kai-Wei Fan, and Prasun Sinha. Cmac: An energy-efficient mac layer protocol using convergent packet forwarding for wireless sensor networks. *ACM Trans. Sen. Netw.*, 5(4):29:1–29:34, November 2009.

[53] Xinxin Liu, Jianlin Guo, G. Bhatti, P. Orlik, and K. Parsons. Load balanced routing for low power and lossy networks. In *Wireless Communications and Networking Conference (WCNC), 2013 IEEE*, pages 2238–2243, 2013.

[54] Kay-Soon Low, W.N.N. Win, and Meng-Joo Er. Wireless sensor networks for industrial environments. In *Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent*

Agents, Web Technologies and Internet Commerce, International Conference on, volume 2, pages 271–276, Nov 2005.

[55] A. Meier, T. Rein, J. Beutel, and L. Thiele. Coping with unreliable channels: Efficient link estimation for low-power wireless sensor networks. In *Networked Sensing Systems, 2008. INSS 2008. 5th International Conference on*, pages 19–26, June 2008.

[56] Jhong Sam Lee & Leonard E. Miller. *CDMA Systems Engineering Handbook*. Artech House Publishers, 1998.

[57] D. Moss and P. Levis. Exploiting physical and link layer boundaries in low-power networking. Technical report, Stanford, 2008.

[58] Eoin O'Connell. Doctoral colloquium: Clean slate system for minimum-power maximum-reliability low-rate multi-hop wireless sensor networks. *RTS*, 100:1–000, 2014.

[59] Eoin O'Connell, Victor Cionca, and Brendan O'Flynn. Software techniques for maximizing network functionality in duty cycle restricted ism bands. *Sensors & Transducers Journal*, 160-161:472–481, 2013.

[60] Eoin O'Connell, Victor Cionca, and Brendan O'Flynn. Techniques for Increasing Network Functionality while Remaining within Legal Maximum TX Duty Cycle Requirements. In *SENSORCOMM 2013, The Seventh International Conference on Sensor Technologies and Applications*, pages 1–6, 2013.

[61] Eoin O'Connell and Brendan O'Flynn. Demo Abstract: Cross Layer Design for Low Power, Low Delay, High Reliability Radio Duty-Cycled Multi-hop WSNs. In Koen Langendoen, Wen Hu, Federico Ferrari, Marco Zimmerling, and Luca Mottola, editors, *Real-World Wireless Sensor Networks*, volume 281 of *Lecture Notes in Electrical Engineering*, pages 73–77. Springer International Publishing, 2014.

[62] Eoin O'Connell, Brendan O'Flynn, and David Boyle. Clocks, Latency and Energy Efficiency in Duty Cycled, Multi-Hop Wireless Sensor Networks. In *Proceedings of IWASI 2013 - 5th IEEE International Workshop on Advances in Sensors and Interfaces*, 2013.

[63] Eoin O'Connell, Brendan O'Flynn, and David Boyle. Energy & Reliability Optimal MAC for WSNs. DRCN 2014. IEEE, 2014.

[64] Jeongyeup Paek, K. Chintalapudi, R. Govindan, J. Caffrey, and S. Masri. A wireless sensor network for structural health monitoring: Performance and experience. In *Embedded Networked Sensors, 2005. EmNetS-II. The Second IEEE Workshop on*, pages 1–10, 2005.

[65] C. Perkins, E. Belding-Royer, and S. Das. Ad Hoc On-Demand Distance Vector (AODV) Routing, 2003.

[66] J. Polastre, R. Szewczyk, and D. Culler. Telos: Enabling Ultra-Low Power Wireless Research. In *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, pages 364–369, April 2005.

[67] Joseph Polastre, Jason Hill, and David Culler. Versatile low power media access for wireless sensor networks. SenSys '04.

[68] Daniele Puccinelli, Silvia Giordano, Marco Zuniga, and Pedro José Marrón. Broadcast-free collection protocol. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, SenSys '12, pages 29–42, New York, NY, USA, 2012. ACM.

[69] Daniele Puccinelli and Martin Haenggi. Arbutus: Network-layer load balancing for wireless sensor networks. In *Wireless Communications and Networking Conference, 2008. WCNC 2008. IEEE*, pages 2063–2068. IEEE, 2008.

[70] D. Raymond, R. Marchany, M. Brownfield, and S. Midkiff. Effects of Denial of Sleep Attacks on Wireless Sensor Network MAC Protocols". In *Information Assurance Workshop, 2006 IEEE*, 2006.

[71] Michele Rondinone, Junaid Ansari, Janne Riihijärvi, and Petri Mähönen. Designing a reliable and stable link quality metric for wireless sensor networks. In *Proceedings of the Workshop on Real-world Wireless Sensor Networks*, REALWSN '08, pages 6–10, New York, NY, USA, 2008. ACM.

[72] Thomas Schmid, Roy Shea, Zainul Charbiwala, Jonathan Friedman, Mani B. Srivastava, and Young H. Cho. On the interaction of clocks, power, and synchronization in duty-cycled embedded sensor nodes. *ACM Trans. Sen. Netw.*, 7(3):24:1–24:19, October 2010.

[73] Mario Strasser, Andreas Meier, Koen Langendoen, and Philipp Blum. Dwarf: Delay-aware robust forwarding for energy-constrained wireless sensor networks. In James Aspnes, Christian Scheideler, Anish Arora, and

Samuel Madden, editors, *Distributed Computing in Sensor Systems*, volume 4549 of *Lecture Notes in Computer Science*, pages 64–81. Springer Berlin Heidelberg, 2007.

[74] Petcharat Suriyachai, James Brown, and Utz Roedig. Time-critical data delivery in wireless sensor networks. In *Proceedings of the 6th IEEE International Conference on Distributed Computing in Sensor Systems*, DCOSS'10, pages 216–229, Berlin, Heidelberg, 2010. Springer-Verlag.

[75] Javier Vales-Alonso, Esteban Egea-López, Alejandro Martínez-Sala, Pablo Pavón-Mariño, M. Victoria Bueno-Delgado, and Joan García-Haro. Performance evaluation of MAC transmission power control in wireless sensor networks. *Comput. Netw.*, 51(6):1483–1498, April 2007.

[76] R. van Langevelde, M. Van Elzakker, D. van Goor, H. Termeer, J. Moss, and A. J. Davie. An ultra-low-power 868/915 mhz rf transceiver for wireless sensor network applications. In *Radio Frequency Integrated Circuits Symposium, 2009. RFIC 2009. IEEE*, pages 113–116, 2009.

[77] R.J.M. Vullers, R.V. Schaijk, H.J. Visser, J. Penders, and C.V. Hoof. Energy harvesting for autonomous wireless sensor networks. *Solid-State Circuits Magazine, IEEE*, 2(2):29–38, Spring 2010.

[78] Qiang Wang, Yaoyao Zhu, and Liang Cheng. Reprogramming wireless sensor networks: challenges and approaches. *Network, IEEE*, 20(3):48–55, May 2006.

[79] Geoff Werner-Allen, Konrad Lorincz, Jeff Johnson, Jonathan Lees, and Matt Welsh. Fidelity and yield in a volcano monitoring sensor network. In *Proceedings of the 7th Symposium on Operating Systems Design and Implementation*, OSDI '06, pages 381–396, Berkeley, CA, USA, 2006. USENIX Association.

[80] Geoffrey Werner-Allen, Patrick Swieskowski, and Matt Welsh. Motelab: A wireless sensor network testbed. In *Proceedings of the 4th international symposium on Information processing in sensor networks*, page 68. IEEE Press, 2005.

[81] Harald Sundmaeker Patrick Guillemin Peter Friess Sylvie Woelfflе, editor. *Vision and Challenges for Realising the Internet of Things.* 2010.

[82] Alec Woo, Terence Tong, and David Culler. Taming the underlying challenges of reliable multihop routing in sensor networks. In *Proceedings*

of the 1st International Conference on Embedded Networked Sensor Systems, SenSys '03, pages 14–27, New York, NY, USA, 2003. ACM.

[83] Kenan Xu, Quanhong Wang, H. Hassanein, and G. Takahara. Optimal wireless sensor networks (wsns) deployment: minimum cost with lifetime constraint. In *Wireless And Mobile Computing, Networking And Communications, 2005. (WiMob'2005), IEEE International Conference on*, volume 3, pages 454–461 Vol. 3, Aug 2005.

[84] Samuel C. Yang. *CDMA RF System Engineering*. Artech House Publishers, 1998.

[85] Wei 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*, volume 3, pages 1567 – 1576 vol.3, 2002.

[86] Wei Ye, Fabio Silva, and John Heidemann. Ultra-low duty cycle MAC with scheduled channel polling. In *Proceedings of the 4th international conference on Embedded networked sensor systems*, USA, 2006. ACM.

[87] I. R. Young, M. A. Dalton, P. J. McMahon, and L. A. Verhagen. Design of an integrated shallow water wave experiment. *Oceanic Engineering, IEEE Journal of*, 22(1):184–188, Jan 1997.

[88] Wen-Jia Yu, Gao-Shang Wang, and An-Jian Wang. A study of copper futures market abroad and reasonable copper price range. *Diqiu Xuebao(Acta Geoscientica Sinica)*, 31(5):699–704, 2010.

[89] Marco Zúñiga Zamalloa and Bhaskar Krishnamachari. An analysis of unreliability and asymmetry in low-power wireless links. *ACM Trans. Sen. Netw.*, 3(2), June 2007.

Appendix A

Schematics

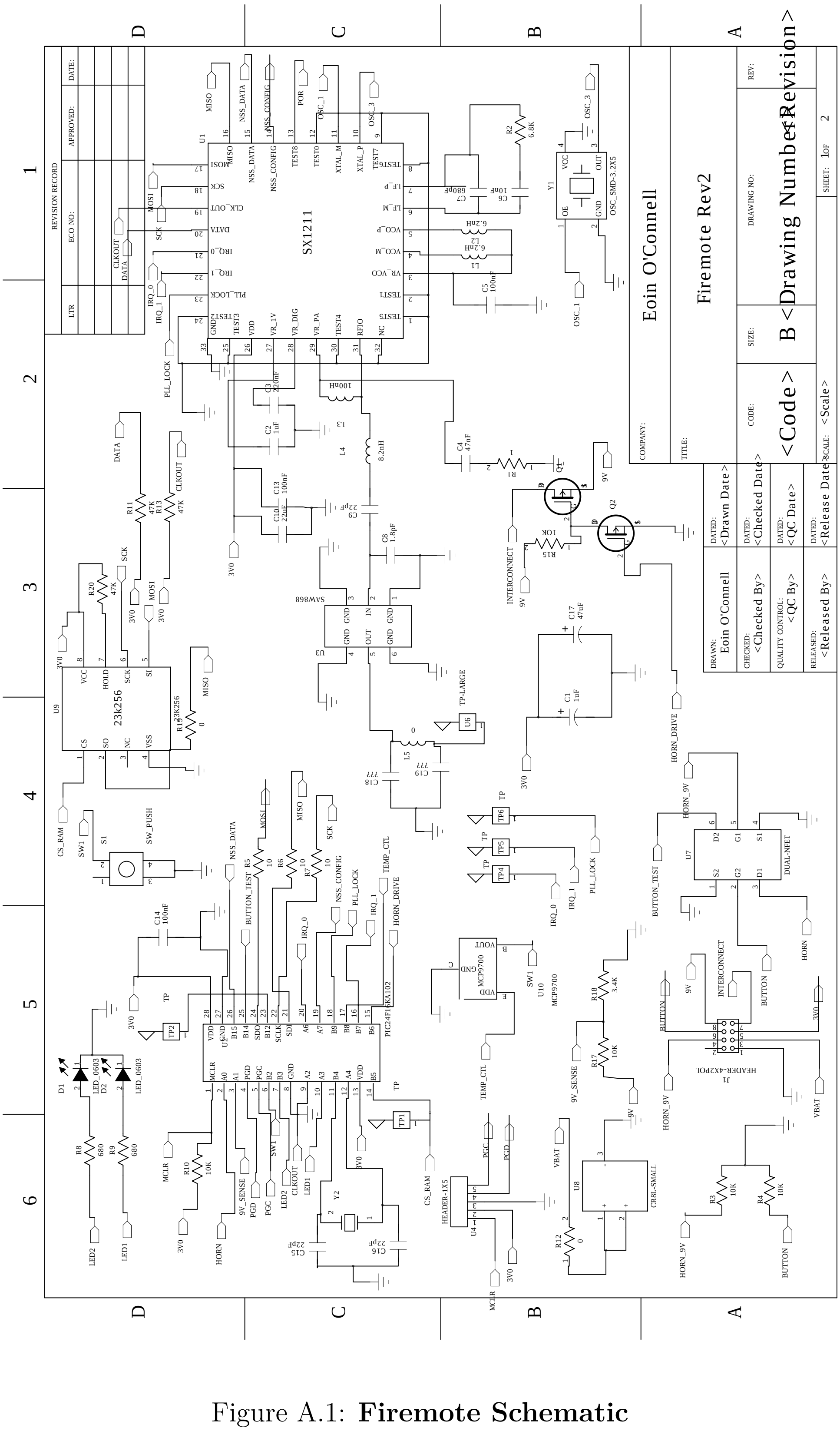


Figure A.1: **Firemote Schematic**