ENDOCRINE INSPIRED CONTROL OF WIRELESS SENSOR NETWORKS: DEPLOYMENT AND ANALYSIS

Tom Blanchard
Department of Computer Science
Aberystwyth University

A thesis submitted for the degree of
Doctor of Philosophy

May 2016

Abstract

Many domains, such as geographical and biological sciences, can benefit from the ability of wireless sensor networks to provide long term, high temporal and spatial resolution sensing. Such networks must be able to trade off various requirements against each other to extend network lifetime while still providing useful, good quality data. The challenges faced by equipment in the field can very unpredictable and therefore a wireless sensor network should be able to cope with these challenges and return to a balanced state.

Using readily available, low-cost components, this work was inspired by the human endocrine systems ability to maintain homeostasis, or balance, in a large number of parameters simultaneously. This work developed a number of endocrine inspired methods. These were aimed both at improving the power usage of nodes in a wireless sensor network and improving the quality of the data collected. Methods for improving power consumption and data quality were achieved. These methods were successfully deployed, for the purposes of environmental monitoring on a mesh network consisting of 20 nodes, for a period of almost 6 months. Analysis showed that the use of power by individual nodes was improved and that the endocrine inspired methods, aimed at improving data quality, were successful. Node lifetimes were extended, duplicate data reduced and the quality of data improved. The use of low-cost, readily available components was largely successful, and challenges and changes to these components were discussed.

Acknowledgements

I would like to thank Aberystwyth University for providing me with opportunities to teach and for funding this work through the Aberystwyth Postgraduate Research Studentship.

There are a large number of people, without whom, this thesis would not have been completed and have supported me throughout my years as a PhD student.

I would like to thank Professor Jon Timmis and Dr Myra Wilson for their feedback on the thesis. Thanks go to everyone in the Department of Computer Science at Aberystwyth University and especially my supervisors Mark Neal and Fred Labrosse. To everyone in E54a and fellow PhD students in particular Mike Clarke, Peter Scully, Colin Sauze and Marek Ososinski who have provided helpful advice, interesting conversation and opportunities for procrastination. To Ian Izzet for all the help and advice with assorted mechanical and electrical problems.

To all of my friends who supported me and helped me stay relatively sane over the years.

Thanks go to my Mum and Dad who supported and helped me throughout both of my degrees, I couldn't have done it without you. Finally and most importantly, I would like to thank my partner Frances who has supported me throughout and helped me survive the tough times. Thank you for the support, proof reading, motivation and help whenever I needed it.

Contents

1 Introduction **1**

1.1 Hypothesis and Research Question 4

1.2 Aims and Objectives 4

1.3 Chapter Summaries 5

1.4 Key Contributions 6

1.5 Summary 6

2 Literature Review **9**

2.1 Introduction 9

2.2 Hardware and Software 10

2.2.1 Hardware Platforms 10

2.2.1.1 Mica Platforms 10

2.2.1.2 Telos/TMote Platforms 11

2.2.1.3 Intel IMote Platforms 11

2.2.1.4 Arduino 12

2.2.1.5 Other Platforms 12

2.2.1.6 Summary of Hardware Platforms 13

2.2.2 Software 13

2.2.2.1 TinyOS 13

2.2.2.2 Contiki 14

2.2.2.3 LiteOS 14

2.2.2.4 Arduino 15

2.2.2.5 Other Operating Systems 15

2.2.2.6 Summary 15

2.3 Mesh Networks 16

2.4 Real World Deployments 17
  2.4.1 Environment, Habitat and Crop Monitoring 17
    2.4.1.1 Environment Monitoring 17
    2.4.1.2 Glacier Monitoring 19
    2.4.1.3 Volcano Monitoring 20
    2.4.1.4 Other Monitoring Deployments 21
  2.4.2 Localisation and Tracking 22
  2.4.3 Summary of Real World Deployments 23
2.5 Power Control in Wireless Sensor Networks 24
  2.5.1 Duty Cycling 25
    2.5.1.1 Synchronous Duty Cycling 25
    2.5.1.2 Semi-Synchronous Duty Cycling 26
    2.5.1.3 Asynchronous Duty Cycling 27
    2.5.1.4 Summary of MAC Duty Cycling Schemes 29
  2.5.2 Adaptive Power Consumption Schemes 30
  2.5.3 Summary 31
2.6 Routing 32
  2.6.1 Classical Routing Protocols 32
    2.6.1.1 Flat Protocols 32
    2.6.1.2 Query Based Protocols 33
    2.6.1.3 Hierarchical Protocols 34
    2.6.1.4 Location Based Protocols 34
    2.6.1.5 Data Centric Protocols 35
    2.6.1.6 Summary of Classical Routing Protocols 36
  2.6.2 Non-Classical Routing Protocols 36
    2.6.2.1 Ant Colony Inspired Routing 37
    2.6.2.2 Bee Inspired Routing 38
  2.6.3 Summary 39
2.7 Data Quality in Wireless Sensor Networks 39
  2.7.1 Time Synchronisation 40
  2.7.2 Anomaly Detection 42
  2.7.3 Summary 42
2.8 The Endocrine System 43

2.8.1 Hormones 44
2.8.2 The Endocrine System 45
2.8.3 Homeostasis 46
2.8.4 Artificial Endocrine Systems 47
2.8.4.1 Hormones in Robotics 47
2.8.4.2 Summary of Hormones in Robotics 52
2.8.4.3 Hormones in Wireless Sensor Networks 52
2.8.4.4 Summary of Hormones in WSNs 55
2.9 Summary of Literature Review 56

3 Preliminary Designs and Deployments 59
3.1 Greenland Deployment 59
3.1.1 Hardware 60
3.1.2 Methods 61
3.1.3 Results and Lessons 62
3.2 Ystumtuen Digimesh Deployment 65
3.2.1 Hardware Changes 65
3.2.2 Methods 66
3.2.3 Results and Lessons 68
3.2.4 Time Stamping 70
3.2.5 Packet Format 71
3.2.6 Conclusions 71
3.3 Ystumtuen Modified Digimesh Deployments 72
3.3.1 Light Varied Transmission 73
3.3.2 Time Synchronisation and Duty Cycle 74
3.3.2.1 Methodology 77
3.3.2.2 Results 77
3.4 Summary of Preliminary Deployments 78

4 Methodology 81
4.1 Node Design 82
4.1.1 Arduino Platform, Microcontroller and Sensors 82
4.1.2 Radio Transceiver 86
4.1.3 Mechanical Design 87

4.1.4 Base Station 87
4.2 Control System and Duty Cycling 89
4.3 Transmission and Routing 91
4.3.1 Unicast Transmission 91
4.3.2 Broadcast Mechanism 93
4.3.3 Packet Routing 93
4.4 Deployment 95
4.4.1 Network Topology 95
4.4.1.1 5 Node Topology 95
4.4.1.2 20 Node Topology 96
4.4.2 Deployment Procedure 97
4.4.2.1 NetDeployer Software 99
4.5 Calibration 100
4.5.1 Pressure 101
4.5.2 Temperature 102
4.5.3 Other Sensors 102
4.5.4 Results of Calibration 103
4.6 Meteorology 104
4.7 Hormone Methodology 105
4.7.1 Hormones Used 106
4.7.2 Selfish Hormone Mechanism 107
4.7.3 Anger Hormone 111
4.7.4 Light Hormone 114
4.7.4.1 Rapid Light Change Detection 115
4.7.5 Centre Hormone 118
4.7.6 Wind Hormone 120
4.7.7 Hormone Production 122
4.7.8 Hormone Decay 122
4.7.9 Hormone Combination 123
4.7.10 Hormone Transmission 126
4.8 Summary 126

5 **Power Usage** **129**
- 5.1 Uncontrolled variables . . . 130
- 5.2 Experiments . . . 131
  - 5.2.1 Control Experiments . . . 133
    - 5.2.1.1 5 Node Power Consumption . . . 133
  - 5.2.2 Selfish and Anger Hormone Experiments . . . 134
- 5.3 Temperature Calibration for Battery Data . . . 134
- 5.4 Filtering . . . 137
  - 5.4.1 Voltage Change Filtering . . . 138
  - 5.4.2 Valid Voltage Filtering . . . 139
  - 5.4.3 Inter-packet Gap Filtering . . . 139
  - 5.4.4 Filtering Results . . . 140
    - 5.4.4.1 Light Integration Method . . . 140
- 5.5 Naïve Model . . . 141
- 5.6 Results . . . 145
  - 5.6.1 Effects of Anger Hormone . . . 155
- 5.7 Summary . . . 158

6 **Data Quality** **161**
- 6.1 Data Quality . . . 162
- 6.2 Experiments . . . 163
- 6.3 Hypotheses . . . 165
- 6.4 Metrics . . . 166
  - 6.4.1 Packet Counts . . . 166
  - 6.4.2 Standard Deviation against N method . . . 166
  - 6.4.3 Duplicate Data . . . 167
- 6.5 Results . . . 168
  - 6.5.1 Packet Counts . . . 168
    - 6.5.1.1 Packet Count Periodicity . . . 172
    - 6.5.1.2 Light, Centre and Wind Experiments . . . 174
    - 6.5.1.3 Packet Count Conclusion . . . 178
  - 6.5.2 Standard Deviation and Sample Size . . . 179
    - 6.5.2.1 Standard Deviation Conclusion . . . 185

6.5.3 Duplicate Data 186
6.5.3.1 Duplicate Data Conclusion 188
6.5.4 Power Analysis of Data Experiments 190
6.6 Summary 191

7 Conclusions and Future Work 195
7.1 Power Management 195
7.2 Data Quality 196
7.3 The Endocrine System and Wireless Sensor Networks 197
7.4 Sensor Node Platform 197
7.4.1 Mechanical Hardware 199
7.5 Key Contributions 199
7.6 Limitations and Lessons 200
7.7 Future Work 201

Bibliography 203

A Integrated Light vs Voltage Change Plots 225

B Light Detector Event Plots 227

C Periodicity Correlations 229

D Duplicate Data Histograms 231

E Experiment Weather Conditions 233

F Node Address Mapping 235

Chapter 1

Introduction

Like many areas of research the limiting factor in the study of geological, anthropogenic and biotic processes is often the amount and quality of the data available. Many regions and processes of interest share common features in that they are some combination of remote, dangerous and lacking in infrastructure. Examples include volcanoes, glaciers, deserts, floodplains, jungles and oceans. Currently, significant time, effort and money is invested in developing new data gathering techniques and planning expeditions to deploy sensors or perform surveys of areas of interest. Whilst people are generally good at this type of work they get tired, need to rest, can only be in one place at a time, are expensive to move and support and most importantly are not expendable. Additional constraints such as adverse weather, limited time between events, unpredictability of an environment and logistical problems are prevalent and can result in missing, or having limited time to observe, events that are of interest.

A specific example is the glaciological research done by researchers in the Institute of Geography and Earth Sciences at Aberystwyth University. Ideally glaciologists would like to know what a glacier is doing at all times, but are particularly interested in certain events, such as calving[1] or melt-water lake draining. While the events in themselves are very interesting, it is understanding the part they play in a larger process that is often considered to be of more importance. This means studying not only the sudden draining of melt-water lakes through the glacier but their impact on the movement of the glacier, the frequency or size of calving events and the glacial environment. However, bringing people and equipment to glacial environments is expensive and in some cases impossible

---

[1] The release of ice from the edge of glaciers.

depending on the time of year. As a result data can only be collected at certain times of year for short periods of up to a few months. Calving and melt-water lake draining events are unpredictable and therefore acquiring measurements before, during and after the event across the glacier is impossible. Conversations held with glaciologists working in IGES and other universities have revealed that the amount of effort required to survey glaciers is disproportionate to the data return.

Wireless sensor networks are often considered to address these issues. Wireless sensor networks were envisioned to consist of thousands of small low cost, low power and expendable sensor nodes that cover an area and are capable of autonomously collecting and disseminating data amongst themselves. The current reality, however, of wireless sensor network technology does not live up to these ideals. Network sizes in all but a few cases are less than 200 nodes, with most being a few tens of devices. The cost per node is still too high to make the production and deployment of thousands, or tens of thousands, feasible. While microcontroller, radio and sensor technologies have come a long way in the last few decades many sensors are still expensive, bulky and require large amounts of power. The high cost of nodes combined with the low numbers of nodes available in a deployment results in each node being “important” rather than expendable. As the number of nodes in networks is often limited due to cost, there is a trade-off to be made between spatial resolution and area covered. When a large area must be monitored, nodes must be placed at “key” locations with little to no redundancy which further reinforces the importance of each node. Typically each node has only a small amount of energy stored in a battery, although some may utilize an energy scavenging technology, such as solar, wind, thermal or vibration. A sensor node is typically broken down into three main sections; sensing, data processing and communications. To reduce the power consumption of a node, the power requirements of one (or more) of these sections must be reduced. To reach operational lifetimes of multiple years, nodes will typically spend the majority of their time in a low power sleep mode occasionally waking to perform some sensing, processing or transmission task.

The temptation is to engineer a solution to each particular deployment, environment and task in an attempt to guarantee performance. This can be a lengthy and expensive task and since the real world is generally unpredictable, there is no guarantee of success. The alternative is to adapt the performance and behaviour of each node for its current environment and situation whilst, at the same time, maintaining the long term goals of the network. If each node is considered important then each node must try to acquire, process and record data whilst keeping the levels of various resources such as battery power, processing power, communications bandwidth and storage, stable. Ideally it should be able to take into consideration numerous parameters including its own state, the state of other nodes, the state of the network as a whole, the environmental state and any specified short or long term goals.

Using long term environmental monitoring of a glacier as an example. If the network deployment is supposed to last for several years, it is no use a node exhausting its energy supply in a week due to acquiring and transmitting data at a very high frequency. Instead sensing behaviour should be adapted to the available resources and to the current global task. Perhaps certain times of day are of particular interest or each node may be able to detect environmental parameters that may be precursors to interesting events. It may, occasionally, be worth a node running out of power entirely to acquire some data of particularly high worth.

The human endocrine system is very good at adapting the behaviour of complex biological systems. It plays an important role in the regulation of a number of systems such as potassium, glucose levels and blood pressure within set limits. Through the release of various hormones the endocrine system is able to maintain homeostasis. Some of these hormones act over very long periods of time, such as the release of growth hormone during childhood development. Some are faster acting and shorter lasting effects, such as adrenalin which is an important part of the human fight-or-flight response. In the case of adrenalin the effects produced such as increased blood sugar levels and blood flow to the muscles, are extreme and are not sustainable in the long term. The endocrine system is able to regulate all of these systems simultaneously, putting up with short term perturbations while maintaining the long term goal of stability. Artificial endocrine inspired systems have been shown to exhibit similar characteristics which have proven to be useful in a number of fields, allowing multiple behaviours to be considered and switched between.

In this work an endocrine inspired approach is taken to the control of nodes in wireless sensor networks.

1.1 Hypothesis and Research Question

The main hypothesis for the work presented in this thesis is:

> Endocrine inspired control methodologies can be used to adapt power usage and increase the quality of data from a wireless sensor network in response to environmental factors.

This leads to the following research questions:

- Can endocrine inspired control methodologies be used to adapt the power consumption of a wireless sensor network.
- Can endocrine inspired control methodologies be used to extend the lifetime of a wireless sensor network.
- Can endocrine inspired control methodologies be used to increase the quantity of data from a wireless sensor network.
- Can endocrine inspired control methodologies be used to decrease the duplication of data from a wireless sensor network.
- Can endocrine inspired control methodologies be used to respond to user input.

Answering these questions is the focus of this thesis and will provide insights into how well endocrine system concepts work in a real wireless sensor network.

1.2 Aims and Objectives

This project aims to take a biologically inspired approach to sensor network behaviour modification, using hormones generated by endocrine inspired controllers in individual sensors to trade off data acquisition goals against available resources. This biologically inspired approach to control system design has been the subject of research within the Intelligent Robotics Group of the Department of Computer Science at Aberystwyth University for a number of years. In particular the use of neuro-endocrine controllers in robotics has been shown to be effective in simulation and laboratory experiments[156, 134] and more recently has been shown to enable a significant reduction in power consumption when used to modify the behaviour of power limited robots. Hormones have allowed the combining and mixing of multiple behaviours, softly switching between them providing smooth transitions between them [156].

To answer the research question the following aims have been determined:

1. To test the hypothesis that endocrine inspired control methodologies can be used to adapt power usage and increase the quality of data from a wireless sensor network in response to environmental factors.

2. Investigate whether the endocrine system is a relevant and applicable system to draw inspiration from for wireless sensor network research.

3. Investigate how to measure and assess the power consumption and data quality in a wireless sensor network.

1.3 Chapter Summaries

- Chapter 2 reviews existing work in biologically inspired systems and wireless sensor networks.

- Chapter 4 details the hardware and software choices made for the sensor nodes and presents and discusses the deployment methodology.

- Chapter 3 presents the results of the preliminary network deployments that informed the final design of the sensor nodes hardware, software, experimental setup and methodology.

- Chapter 5 presents the results of a series of experiments focussed on improving the wireless sensor network's power usage in an endocrine inspired way.

- Chapter 6 presents the results of three experiments that aimed to improve the quality of data produced by the sensor network.

- Chapter 7 discusses the conclusions, key contributions, lessons learnt and potential directions for future work.

1.4 Key Contributions

The novel contributions that have been made by this work are as follows:

- Multiple endocrine inspired components can be combined while retaining the behaviour of each individual component.(Chapters 5 and 6).
- The ability of a wireless sensor network to adapt power consumption can be improved through the use of endocrine inspired systems when compared to a simple non adaptive control system. (Chapter 5).
- The quality of data returned from a wireless sensor network can be improved through the use of endocrine inspired systems. (Chapter 6).
- The power adaptation and data quality can be improved simultaneously through the use of endocrine inspired systems. (Chapter 4 and 6).
- Deployment of several endocrine inspired control methodologies on a real world sensor network. (Chapters 3, 4, 5 and 6).
- Development of a metric to analyse how well the energy available to a wireless sensor network was used. (Chapter 5).

1.5 Summary

Much current data acquisition is dangerous, expensive and does not acquire the desired type and quantity of data. Sensor networks claim to solve this problem but the reality is that the technology is not yet at the point where networks of tens of thousands of nodes blanketing an area in perpetuity is a reality. This is partly due to cost of the hardware, power consumption required by sensors and data transmission systems, deployment and networking constraints. As a result each sensor is actually important and needs to balance its own behaviours against the state of other sensor node and the global network goal(s). The human endocrine system is able to regulate a large number of complex systems simultaneously even during some extreme perturbations. The behaviour of each system is able to be softly modulated simultaneously without the need to discretely switch between each one. These properties have, to certain extents, been replicated in artificial endocrine systems. We hope to draw inspiration from the endocrine system to test the hypotheses set out in Section 1.1.

In the next chapter the existing literature in the fields of wireless sensor networks and endocrine inspired system is discussed.

Chapter 2

Literature Review

2.1 Introduction

This chapter provides an overview of work carried out in a number of areas considered to be relevant to this thesis. Firstly, an overview of the hardware and software available for wireless sensor networks and the features provided is presented. A number of real world deployments are then discussed as are the numerous important lessons to be learnt from such deployments. Methods of controlling power consumption, in particular duty cycle based methods are presented and shown to largely be successful in their aims. A number of different routing protocols are described and their relevance to the environmental monitoring scenario is discussed. Data quality as it pertains to this work and related work is detailed in Section 2.7. The basic biology of the human endocrine system, hormones and homeostasis is presented followed by a review of endocrine or hormone inspired work in robotics and wireless sensor networks.

While work has been separated into various categories for the purposes of discussion, much of it is interlinked and affects numerous other aspects of wireless sensor networks. For example many methods of improving power consumption also improve data quality and vice versa. Discussions on hardware, software and real deployments are considered especially important as ultimately, the goal of most work in wireless sensor networks is to be used in a real sensor network, in the real world, for a real task.

2.2 Hardware and Software

2.2.1 Hardware Platforms

There are a large number of available hardware platforms for conducting wireless sensor research. One of the reasons for such a large range of devices is that the hardware is not inherently complex. With the price of microcontrollers, electrical components and board fabrication becoming lower, many research institutions have developed their own hardware. The ability to create hardware allows it to be tailored to the research being conducted with some components or features omitted and others added. There are, however, a small selection of wireless sensor node hardware platforms that are more widely used and are, therefore, discussed in more detail.

A large amount of the initial work into wireless sensor hardware platforms came from the Smart Dust project [189, 92, 93] at Berkeley University. This was a DARPA funded project to develop extremely low power and small sensor nodes, the ultimate goal being nodes that were 1 mm^3 in size. The use of light based communications both passive, using Corner-Cube Reflectors and active, using steered laser beams was investigated and found to be significantly more efficient than RF based communications. Ultimately this research paved the way for the Mica and Telos platforms.

2.2.1.1 Mica Platforms

The first of these is the Mica platform [72] which was developed by Berkeley and Intel. The Mica platform featured an 8 bit microcontroller, the Atmega103L or Atmega128, running at 4 MHz. It had a number of I/O lines, UARTS, SPI and a 10 bit ADC. The microcontroller provided 128 Kbytes of flash memory for program storage, 4 Kbyte of RAM and an external 4 Mbit flash chip was used for persistent data storage. An RF Monolithics TR1000 radio transceiver was used for communications. This provided a maximum data rate of 115 kbits/ps at a frequency of 916.5 MHz. The Mica2 platform [28] improved on the Mica platform by reducing power consumption to sleep currents less than μA, offering devices with a range of transceiver frequencies (868 MHz, 915 MHz, 433 MHz, 315 MHz) and provided 512 KBytes of serial flash for data storage. The same microcontroller was used as in the Mica platform but used a Chipcon 1000 radio transceiver. The MicaZ [125] platform used the Chipcon 2400, 2.4 GHz, radio transceiver but was otherwise the same as the Mica2 platform. Mica2Dot [29] came in a smaller, coin sized form factor and was based on the same hardware as the Mica2 platform with slightly reduced input and output capabilities. This family of hardware platforms [71, 70] were one of the main devices for wireless sensor network research for a number of years.

2.2.1.2 Telos/TMote Platforms

Another popular hardware platform used in wireless sensor network research was the Telos (TMote) and TelosB (TMote Sky)[30, 174] platforms also from Berkeley University. These used a Texas Instruments MSP430 microcontroller running at 8 MHz and ChipCon 2420 radio transceiver. The MSP430 enabled even lower power consumption by providing a 21 μA idle mode and the ability to operate at very low voltages, enabling it to extract more energy from connected batteries. The MSP430 has 46 KBytes of program flash memory and 10 KBytes of RAM. This provided less program space but more RAM than the Mica2 platform which was intended to enable more sophisticated applications. The addition of USB allowed the device to be more easily programmed than other nodes as no external programming devices was necessary. The fast start up times and low power consumption made the platform a popular choice.

2.2.1.3 Intel IMote Platforms

The Intel Imote and Imote2[31, 132, 81] (Manufactured by Crossbow Technology) was another commonly used platform as it provided significantly more computational power than the alternatives. The Intel PCA271 XScale processor has multiple low power modes and the ability to dynamically change its operating frequency from 13 MHz to 416 MHz. It offered significantly more RAM, 32 MBytes of SRAM combined with 250 KBytes of SDRAM and 32 MBytes of Flash for storage data. This increased processing power came at the cost of power consumption which was almost $25 - 75$ times higher in sleep mode than the MicaZ or Telos platforms. The higher energy consumption, it was felt, would be balanced in certain applications by the ability to perform significant amounts of processing on the sensor node therefore reducing the amount of data needing to be transmitted. In particular, high frequency signal analysis could be performed on the node rather than needing to transmit all of the data to some base station for processing. This approach is only beneficial in applications and network deployments where the amount of data transmitted can be significantly reduced by on-sensor processing.

2.2.1.4 Arduino

The Arduino platform [3] is an open source project designed to more easily allow users to build devices capable of sensing, manipulating and interacting with their environment. Founded in 2005, it has since become arguably the worlds biggest microcontroller development platform. It aims to have a low barrier to entry and the programming environment and libraries support a huge range of sensors, actuators and electronic devices. A number of hardware platforms are supported, although the Atmega series of microcontrollers are best supported. This results in a range of microcontrollers that are easily developed for; from simple 8 pin devices running at less than 1 MHz to much more capable 32-bit ARM core devices running at 70+ MHz. A large number of Arduino development boards exist that support a plethora of radio transceivers, making it an attractive platform for development.

2.2.1.5 Other Platforms

In the intervening 10 years since the initial development of the Mica, Telos and IMote platforms, a large number of hardware platforms have been developed. IRIS Motes[126], Shimmer (1,2 and 3)[166], BTNodes, LOTUS motes[124] and Sun SPOTs[167] were all popular and well used in research[89]. More recently, platforms such as the WiSMote mini, Waspmotes[106], the Moteino platform, egs motes[98] and panStamp have been developed. Many of these are more powerful, include more features and are cheaper and easier to acquire than previous platforms.

Many of these platforms are very similar, using the same Atmel 128 or Texas Instruments MSP430 microcontrollers. One reason for this is that these microcontrollers are cheap, well known and supported by many wireless sensor network software stacks and operating systems. Much of the improvement over the years has been in the realm of wireless transceivers[89], with improved data rates, lower power consumption and more hardware accelerators being provided. As computing power and power efficiency have improved, nodes have started to be developed with greater computational power. Examples of this are the egs motes[98] and Sun SPOTS[167]. Most of the commonly used wireless sensor network hardware platforms still have only 50 to 100 KBytes of program memory, a few tens of KBytes of RAM and run at frequencies in 8 – 50 MHz range.

2.2.1.6 Summary of Hardware Platforms

There are a large number of existing hardware platforms that are designed to be used in wireless sensor network applications. Some, like the Mica and Telos series, have been used in a large amount of wireless sensor network research and are very well regarded. However, in recent years there has been a trend to designing and building hardware platforms tailored towards a particular research application or project. Perhaps the biggest driving factors behind this are that different applications can have very different requirements, and the cost of designing and building small quantities of hardware is relatively low. This allows a sensor node to be created using a microcontroller that is familiar to those conducting research, the specific radio transceiver desired and any additional sensors, peripherals or actuators.

2.2.2 Software

There are a large number of real time or embedded operating systems used for embedded or low power hardware today. Many are oriented towards specific domains such as the automotive industry or aeronautical applications. However, there are fewer operating systems designed specifically for wireless sensor networks. The following sections detail some of the most popular and most used operating systems in the field of wireless sensor networks[49, 43].

2.2.2.1 TinyOS

Perhaps the first wireless sensor network operating system was TinyOS[54] initially released in 2000. It, like the Mica and Telos hardware platforms was developed by Berkeley University in conjunction with Intel and Crossbow Technology. Whilst originally being written in C and Perl, TinyOs quickly transitioned to nesC[54], a dialect of C that was optimized for wireless sensor systems and the associated constraints (such as limited memory). TinyOS focussed on being “event centric” and a platform for wireless sensor network research. TinyOS is very small, around 400 bytes of code and data memory combined, which is essential for systems with so little memory and nesC does not allow dynamic memory allocation. TinyOS has no multi-threading capabilities, instead it relies on event handlers and run-to-completion tasks. Typically, an event is handled by an event handler which posts any necessary tasks to be scheduled by TinyOS for execution later in a non pre-emptive FIFO manner. This allows TinyOS to support concurrency and nesC detects data races at compile time to help eliminate race-condition bugs. TinyOS supports a large number of hardware platforms. Due to its widespread use, this support is usually provided by the makers of the hardware platform. TinyOS has an associated simulator called TOSSIM[54] which allows programs, written for TinyOS, to be tested on a variety of deployments and large numbers of nodes.

2.2.2.2 Contiki

Contiki[42, 41] is another popular open source operating system designed with wireless sensor networks and embedded systems in mind. It was developed in 2002 by Adam Dunkels with support from several hardware manufacturers such as TI, Atmel and Cisco. Contiki uses “protothreads” a mixture between threads and event driven tasks as a large part of its architecture. This provides the ability to support concurrent operations, such as blocking event handlers, while maintaining a low memory footprint. There are a variety of networking stacks provided by Contiki including IPv4, IPv6 and a large number of lightweight protocols for wireless sensor networks. Dynamic memory allocation is allowed and provides a managed memory allocator to reduce memory bugs. Contiki supports a large range of hardware including the popular Atmel AVR and TI MSP430 microcontrollers used in a lot of hardware platforms. It also provides a file system for flash memory (Coffee), a simulator (Cooja), power awareness, a command shell and dynamic module loading.

2.2.2.3 LiteOS

LiteOS[23] is a UNIX-like operating system for wireless sensor networks. It aims to provide a more familiar environment to users than operating systems such as TinyOS, by providing features such as a UNIX-like system shell, dynamic loading and a hierarchical file system. LiteOS provides a multithreaded kernel which enables it to run programs concurrently as threads. Wireless sensor nodes running LiteOS can be mounted to the root file system of a nearby base station allowing access to their file systems from the base station. This enables remote programming, testing and monitoring of programs. Unlike TinyOS, LiteOS uses threads to maintain execution contexts, allowing threads to be suspended until some operation has finished. It also provides support for internal events, using callback functions, to eliminate the overhead of context switching, creating and destroying threads[23, 22].

2.2.2.4 Arduino

The Arduino [3] platform also includes a large amount of software libraries for making use of the features on a number of supported micro-controllers. There is no provided operating system, mainly due to the limited resources on most Arduino compatible devices, but there is a thriving support and hobbyist community. This makes finding solutions to problem relatively easy and there are a large number of third party open source libraries, resources, tutorials and hardware. This results in a platform that is easy to use, enables quick prototyping and is constrained only by hardware limitations rather than software choices made by others.

2.2.2.5 Other Operating Systems

Other wireless sensor network operating systems exist, such as Mantis [13], ERIKIA and Nano-RK [48]. They all focus on prioritising low kernel RAM usage as it is typically the most constrained resource. They also provide threads, semaphores and mutexs in a similar manner to larger operating systems. Many resources are allocated statically, for example Mantis allocates threads statically so there is a maximum number of possible threads at any one time and Nano-RK allocates task parameters statically.

2.2.2.6 Summary

Typically the wireless sensor network operating systems discussed above, with the exception of ERIKA, are not hard real time operating systems. While many tasks are time sensitive, missing a packet acknowledge window, taking a light reading a few milliseconds late or taking several seconds to carry out a signal processing task, are unlikely to severely impact the sensor node or application. Most of the operating systems presented also aim to conserve power as much as possible. TinyOS sleeps as soon as it has run out of tasks to schedule and threaded operating systems such as LiteOS, Mantis and Contiki will sleep when there are no threads that can be executed. Mantis, for example, allows threads to call the UNIX-like function “sleep()”. As long as all threads call the sleep function then the microcontroller will enter a low power sleep state until a thread needs to run.

There are a number of software platforms or operating systems available for use in wireless sensor networks. Many have a long list of impressive features that would be useful for a wireless sensor network deployment. One commonality is the complexity of the solutions, while significantly less complex than a desktop operating system, is still non trivial. Choosing a platform imposes a number of limitations and constraints, some of which may be unanticipated and as such an option with the most flexibility is desirable.

2.3 Mesh Networks

Mesh networks are networks consisting of a number of nodes that form an ad-hoc multihop network. Each node acts as both a mesh client and a mesh router and forwards data for other nodes that cannot directly communicate with their target [2]. Wireless sensor networks typically fall into this category of network for several reasons. As they do not use a fixed infrastructure for communications they are a type of ad-hoc network. Nodes are typically homogeneous and prone to failure meaning that each node should be able to act as a router for other nodes to provide a decentralised self healing network. Wireless sensor networks are required to cover large areas that necessitate the use of multihop routing. While wireless sensor networks typically transmit data to a specific base station node, that node often does not manage the network. Significant effort in mesh network research is directed at providing network connectivity to areas that lack it [50, 129]. One example of this is the one laptop per child [137] laptops which use a low power 802.11 wireless module to provide low power meshing to nearby devices even while the laptop is in low power modes. These types of mesh networks aim to provide high bandwidth, low latency connections which comes at cost in terms of power consumption. As many of the largest mesh networks exist in cities or built up areas, this does not present a problem. Many mesh networks such as the Freifunk system in Germany use off the shelf equipment, typically 802.11 wireless access points. These run custom firmware and routing protocols such as AODV [144], BATMAN [87] and OSLR [82].

The bandwidth (multi megabit) and latency (sub 10 ms) provided by many high powered mesh networks is far in excess of what is typically needed in a wireless sensor network. Low power mesh networks sacrifice bandwidth and latency in an effort to reduce power consumption. Route discovery and maintenance is expensive in terms of power consumption and as such significant research is conducted into routing protocols. Many of these are discussed in more detail in Section 2.6. Perhaps the most widely used low power mesh networking standard is Zigbee [210] which is widely used in home automation [56].

The aspect of mesh networks that is most appealing to the wireless sensor network domain is the ability to self heal. The resulting network is therefore not dependant on any one node. This is key when there is a high possibility of nodes becoming damaged, running out of power or environmental factors affecting signal quality. It will be important to incorporate these elements into any sensor network.

2.4 Real World Deployments

Much of the research into wireless sensor networks uses simulation to test new developments. Due to the large number of possible combinations of hardware platforms, network topologies, number of nodes and techniques to compare against, real world testing is often neglected. Some work does use a small deployment of sensor nodes in a relatively safe environment, such as an office, for a short period of time to validate performance. Large scale deployments in real world conditions are still relatively rare. Comparing a new technique to existing techniques may require multiple experiments in different configurations. This results in a substantial amount of time required to fully test systems. Another issue with real world deployments is that data is unreliably delivered, there are no periodic ‘snapshots’ of the network state. Therefore, it can be difficult to determine the state of the network and the cause of failures or specific behaviour[158]. Some real world deployments have, however, been carried out. The domain which has been the most widely explored through the use of real world deployments is environmental, habitat and crop monitoring. This is, perhaps, due to the large demand for data in these areas [64, 2].

2.4.1 Environment, Habitat and Crop Monitoring

2.4.1.1 Environment Monitoring

Perhaps the first large environmental monitoring deployment was carried out by Berkeley University in conjunction with the Intel Research Laboratory. An initial deployment of 43 nodes was carried out in 2002 before a larger 150 node deployment in 2003. Each deployment lasted around 4 months and was situated on an offshore Storm Petrel breeding colony in Maine called Great Duck Island[173, 146, 175, 112]. The deployment featured dense “sensor patches” which used low power Mica or Mica2Dot hardware to communicate with patch gateways. These gateways formed a “transit network” that relayed data to central base station equipped with a satellite link. The deployment featured nodes in nesting burrows and above ground. Verification was achieved through the use of cameras equipped with infra-red lights. A duty cycle approach was used to reduce power consumption which worked well, with only 5 nodes running out of power in the first deployment. In the second deployment a large proportion of nodes underground showed reduced lifetimes, an average of 60 days vs 104 days, as did those involved in multihop routing which was thought to be as a result of nodes overhearing traffic from other nodes. These deployments exhibited a high node failure rate and showed that there were unexpected low level behaviours such as the MAC-layer synchronisation of nodes. The Great Duck Island deployments highlighted the importance of reliability in nodes and equipment, while providing a large and useful dataset.

A separate deployment by the same team was carried out in 2004 to monitor a redwood tree using 33 nodes[179]. The deployment lasted for 44 days and collected approximately 49% of the data it set out to. Due to the high sensing rate, every five minutes, this still yielded a significant quantity of good data. The physical deployment of the network was also found to be important. Small variations in sensor placement and alignment had large impacts on the quality of data collected. The need for a network monitoring system was highlighted when the internal data storage became full on many nodes and was not detected during deployment.

These unforeseen issues that come with real world deployments were also found by the Extensible Sensing System project[57]. A number of assumptions were found to be incorrect such as: good RF connectivity, that hardware and software would work in the real world as it did in the lab, that energy consumption was more important than data and that domain scientists would know where sensor nodes should be placed.

EPFL’s SensorScope project aims to provide a wireless sensor network for high altitude climate monitoring[158, 9, 79]. Instead of using the low power listening techniques employed in other deployments the nodes use a synchronised duty cycle approach. Nodes synchronise their clocks to the base station and all wake or sleep at the same time. This removes the need for long preamble transmission and was considered to lower power consumption. A number of deployments have been carried out using the SensorScope platform ranging from 6 to 97 nodes for durations of 4 to 180 days in a number of Alpine locations.

Many deployments featured single hop topologies, although some smaller deployments did use multi hop network topologies. One of the reasons cited for the use of single hop topologies was that such networks were quick and simple to set up which was required due to the unpredictability of the events being monitored.

A very similar deployment to the SensorScope deployments is the PermaSense project[176, 65]. These deployments were located in several locations in the Swiss Alps and used relatively small numbers of nodes, between 10 – 25. Sensor rods were inserted into the rock to gain data on heat transport in frozen rock walls. Nodes had estimated lifetimes of 3 – 5 years and were built to be robust due to the sub zero temperatures, risk of icing and rockfalls. The first deployment provided a real world test bed and was eventually left in a mode where data was only stored locally on each node. One of the critical areas requiring work was considered to be time synchronisation as the crystal oscillators used were subject to high drifts in the negative temperatures.

A recent large scale deployment was the GreenOrbs project[111] which used 330 nodes to monitor the ecology of a forest environment and determine whether wireless sensor networks can scale to much larger numbers of nodes. The deployment is ongoing with plans for even larger deployments. Results showed that energy efficiency and multihop routing are not as big a concern in large sized networks as previously thought.

2.4.1.2 Glacier Monitoring

Another significant real world environmental monitoring sensor network deployment was the GLACSWEB project [121, 123] conducted by Southampton University in Briksdalbreen, Norway. The GLACSWEB project aimed to track the movement of ice in a glacier as the glacier flowed towards its terminating face. Eight nodes were deployed at a depth of 50 to 80 m into the ice of the glacier through holes made using a high pressure hot jet-wash. A base station was mounted on the surface of the glacier and periodically communicated with the glacially embedded nodes and recorded its location once a week using differential GPS. A separate reference station acted as a data relay and was situated in a location with mains power and an internet connection. Each node was expected to have a lifetime of almost 10 years sampling at a rate of once every 4 hours. Due to the inaccessibility of the nodes, they were designed to be robust with data storage capabilities and the ability to self restart. Communication was especially difficult due to the large quantities of ice and water between nodes and the base station[138]. Of the 8 nodes deployed after 3 months only 3 nodes were still communicating with the base station. It was felt that this could be due to node failure, as they were under significant stress, issues with the base station or the nodes being carried out of range by the glacial movement.

The performance of the system was improved in subsequent deployments, with larger data buffers to allow for long periods with no communication[118]. The base station was improved by using a combination of high and low power processors[119] and these improvements were tested with new deployments in Iceland[122, 120]. These deployments lasted significantly longer and provided more data than the initial deployment. This work suggests that it can take multiple deployments and many years to create a working wireless sensor network for a specific real world domain.

Another successful glacial network deployment was conducted in 2013 on Helheim Glacier in Greenland [117, 130]. This deployment used Libelium Waspmotes equipped with two 2.4 GHz Zigbee transceivers and precision GPS equipment to monitor the movement of the glacier. Twenty nodes were deployed for a period of 53 days during which 19 nodes successfully communicated with the base stations. This deployment differed from most others by using 4 base stations. The 20 nodes were organised into 4 networks of 5 nodes and each network was able to communicate with two base stations. This was successful in improving reliability and dealing with the glaciers and nodes movement. Results showed that there were occasions where one base station was able to connect to a node when the second base station wasn't. The deployment was highly successful and gathered a large amount of very high resolution data, even capturing a node being lost during a calving event. The deployment was especially significant as no power saving was attempted and with the use of solar panels the nodes were all able to survive the desired length of time.

2.4.1.3 Volcano Monitoring

Another environment that has benefit from wireless sensor network monitoring is volcano monitoring. In 2005 16 nodes were deployed for 19 days at the active Reventador volcano in Ecuador[193, 192]. Each node sampled seismic data at 100 Hz and used a seismic event detection algorithm[191] to store and later transmit interesting seismic events. The nodes needed to be synchronised so that the seismic data could be accurately timestamped for further processing. When an event was detected by a node a message was sent to the base station. As long as enough nodes reported detecting an event, data collection was initiated. Nodes constantly timestamped and stored data locally and when commanded, would transmit the requested data to the base station. This allowed for high resolution, 100 Hz, event data capture. Once initial problems had been fixed the network worked well with at most 4.5 % packet loss over the 54 hour deployment. One node was seriously damaged by flying debris from the volcano, which highlights some of the unforeseen dangers of real world deployments.

More recently work was carried out with 13 nodes on St Helens Volcano using the OASIS (Optimized Autonomous Space In-situ SensorWeb) system[76]. The OASIS system uses iMote2 platform in conjunction with accurate GPS devices, seismometers, infrasonic sensors and lightning sensors. Nodes were air-dropped to their location and were able to form a mesh network. A TDMA scheme called TreeMAC was used as the GPS device on each node was able to provide very precise global clock. The network was able to detect and record 140 events compared to 160 detected by a significantly more expensive control unit. The deployment did suffer from failures due to lightning strikes, which again highlights the difficulty in real world deployments.

2.4.1.4 Other Monitoring Deployments

Crop monitoring is another environmental sensing domain that has seen real sensor network deployments. This is often driven by the desire to create micro-climate models of crops to enable higher yields[53, 209]. The LOFAR-agro project was the first large-scale experiment in precision agriculture in the Netherlands. One pilot study carried out as part of the project attempted to monitor a potato crop to help protect against *phytophthora*, a fungal disease capable of quickly destroying a whole harvest[104]. The node hardware used was similar to the Mica2 hardware and used TinyOS. 109 nodes were deployed on stakes around a target field. A large number of problems, hardware and software failures resulted in only 2% of the expected quantity of data being collected. Many of the reasons given for the failures, such as difficulty debugging and time constraints, were due to the difficulties encountered when deploying a sensor network. The projects second deployment[62], the following year, was much more successful with over 50% of the estimated data collected and more recently work into assessing the viability of larger deployments in Mauritius is being conducted[95].

This is not an exhaustive review of all wireless sensor network deployments in the monitoring domain, there have been many others [178, 133, 10, 208]. The common theme is that it is hard to deploy these networks in the real world, things break and go wrong and do not work as they should theoretically work. As such, solutions are usually engineered for maximum reliability rather than allowing for true autonomous behaviour in nodes [8]. In many deployments the nodes are not directly networked for example Argo float network [151], ADIOS [90] GPS system or HiTemp project [201]. These systems typically have sensors that communicate using some external infrastructure such as satellite, GSM or WiFi. ADIOS and Argo floats both use satellite modems to communicate and HiTemp makes use of nearby WiFi access points. The Argo float project has deployed, over the last 15 years, almost 4000 drifting buoy sensors. These sensors descend to a depth of 2000 m and while taking temperature and salinity profiles. The project is considered to be very successful, with more than 1 million profiles collected.

Structural monitoring has also seen the use of wireless sensor network deployments. There is often, however, existing power[195] or communications infrastructure that can be used to simplify the sensor network solution. An example of a wireless sensor network deployment for structural monitoring without external infrastructure was the work done to monitor the movements of the Golden Gate Bridge [96].

2.4.2 Localisation and Tracking

Most of the remaining real world deployments fall into the category of tracking or localisation. Generally tracking refers to attaching the sensor node to the target and localisation implies a static network that is able to locate and track a mobile target passing through the network. However, the terms are often used interchangeably in the literature.

ZebraNet[91, 205] was perhaps the first wireless sensor network for tracking wildlife. Previous solutions typically used VHF collars that sent out a “ping” that could be located. More accurate solutions used GPS and satellite communications which, due to their power consumption, limited the amount of data that could be transmitted. The cost of satellite based solutions was also prohibitively expensive for more than a few nodes. ZebraNet trialled a number of deployments, from 2002 to 2007, in the Mpala Research Centre in Kenya. Sensor node hardware evolved from a GPS module with multiple radio transceivers to a fairly standard TI MSP430 based solution with a high power 900 MHz transceiver and GPS module. ZebraNet used a flooding based protocol [110] in which each node (attached to its respective Zebra) woke up once every two hours and transmitted as much location data as possible, in five minutes, to any other nodes in range. Each node was equipped with 4 Mbits of flash storage which was partitioned into blocks for local storage and blocks for storing data from other nodes. Eventually, a node was expected to be in range of a base station and be able to transmit all of its data. Due to the high power radio modules the power consumption of the nodes was too high to be sustained by batteries alone, small solar cells were therefore added to recharge the batteries. The network successfully tracked 7 zebras for two weeks, recording GPS location information every 8 minutes. Proposals have since been made on how to improve the flooding protocols used in ZebraNet [11]. Numerous other wildlife tracking deployments have been carried out such as the RFID-wireless sensor network hybrid in [45] used for badger tracking and environment monitoring.

Much of the work in wireless sensor network tracking or localisation is oriented towards military applications. The ExScal [7] project deployed 1000 nodes with another 200 nodes providing a network backbone to enable tracking and event detection within an area. The network produced 1 millions observations and was tested using low frequency traffic, high frequency traffic and burst traffic. Nodes showed a reliability of at worst 58 % and at best 86 %. The Trio project [44] created a deployment of 557 solar power nodes for the purpose of testing multi-target tracking. It operated for four months and was able to track three people crossing paths in real time. VigilNet [184, 66] used a deployment of 200 nodes to provide realistic data for a 10,000 node simulation with a goal of creating both long term and large scale tracking networks. The Self Healing Minefield [152] was a DARPA project focused on localising and tracking a target in an area. Unlike most other networks, the nodes were equipped with four gas thrusters that allowed them to move a short distance a few times, before running out of propellant. This allowed the network to 'self-heal' by having undamaged nodes physically reposition themselves.

2.4.3 Summary of Real World Deployments

There are a significant number of real world deployments of wireless sensor networks. Most deployments take multiple attempts to get to a useful operational state. In many instances a small network of a few sensors is trialled for a period of time. Lessons from the small deployment are used to inform work into increasing the size and capabilities of a larger deployment. The unpredictability and harshness of the real world often leads to node failures, failures or poor performance of protocols and unreliability. Despite this, there are now several deployments with a relatively large (more than 100) number of nodes active for long periods of time. It is felt that this shows that long term real world deployment of wireless sensor networks is definitely achievable and that working towards a real world deployment is a worthwhile goal.

2.5 Power Control in Wireless Sensor Networks

One of the most important driving factors in wireless sensor research is power consumption. The nodes in real world wireless sensor networks deployments are almost always battery powered and sometimes equipped with energy scavenging technology, such as solar panels. There are a broad range of methods for improving power usage or reducing power consumption. Some of the improvements are due to technological progress in sensor, microcontroller and battery technologies. However, how energy is used by the node usually plays the most significant role. In particular power consumption for real world sensor network deployments is crucial, as accessing the deployment site may be very difficult, time consuming and expensive making battery replacement impractical.

The primary method for saving power is to turn off any components that are not essential at that point in time. Sensors typically only need to be powered when actually taking a measurement, although some may need to be constantly powered. It stands that if the quantity of data being collected does not change then the power consumption of the sensors is fixed. This leaves the microcontroller itself and the chosen communications transceiver, which is usually a radio transceiver. The microcontroller itself can be put into a low power mode for large periods of time due to the limited amount of processing typically carried out on a wireless sensor node. If we assume that the collected data must be transmitted, i.e. it cannot be stored for later retrieval, then the amount of power required to transmit this data is also fixed at the transceiver's power consumption per bit. Power savings therefore must be made by turning the transceiver and microcontroller off as often as possible while maintaining the ability to route data through the network. While there may not be a need for multi-hop routing in some scenarios, the ability to initiate communication with a specific node in a timely manner is often required. It follows, therefore, that one of the areas that has the most significant impact on power control is communications medium access. MAC (Medium Access Control) dictates how a device deals with; sending packets to another device, transmission collisions and accessing the transmission medium.

2.5.1 Duty Cycling

Perhaps the most widely used method of conserving power is duty cycling various components. The transceiver is typically the component that expends the most energy and in most cases requires that the microcontroller also be powered to control it. Care must be taken when powering down the transceiver module, otherwise data from other nodes will be lost. Duty cycling is one way to support both requirements; low power consumption and the ability to reliably transmit and receive data. Many duty cycling schemes are actually designed as new MAC layers as this is where most of the transceiver's power consumption is controlled.

The authors of [25] consider there to be three categories of duty cycling techniques; synchronous, asynchronous and semi-synchronous. As the name implies synchronous duty cycle methods require some form of synchronisation, the performance of which is often the limiting factor in communication. Time synchronisation is discussed in more detail in Section 2.7.1. Asynchronous duty cycling is typically less complex but the time taken for data to reach its destination is large or unpredictable making it unsuitable for some applications. Whilst this categorisation is not perfect it is perhaps the best that is available and as such will be used when reviewing the existing duty cycling techniques.

2.5.1.1 Synchronous Duty Cycling

In synchronous duty cycling schemes, time is divided into slots that are assigned to specific nodes. A node is allowed to transmit only in this time slot. To ensure that nodes keep to their allotted time slots, it is assumed that time synchronisation is performed to correct any clock drift. TRAMA[149] uses the concept of scheduled and random access slots. Scheduled time slots are purely for the node they were assigned to, whereas random access slots operate on a contention based policy. Information on neighbouring nodes and a transmission schedule is propagated to neighbours. An Adaptive Election Algorithm is then used to select which node is assigned to a time slot. This allows other nodes to enter low power states and save power.

RT-Link[153] uses GPS receivers or AM radios to allow nodes to be very tightly synchronised. This has the benefit of allowing collision free operation and known end-to-end delays for data. The signal from the GPS or AM radio is used to mark the start of a finely slotted period of communication. Cycles are split into frames, where each frame is split into smaller slots marked as either scheduled or contention slots. Nodes initially use contention slots to communicate their existence, which allows them to be scheduled a time slot for transmission. When a node ceases to communicate, its time slot is eventually reclaimed.

TRAMA and RT-Link are "rendezvous" schemes, where all nodes turn on at the exact same time, staggered schemes exist. CUPID,[101] PELLMAC[142] and LEEMAC[77] are all schemes in which nodes are are scheduled to wake up at slightly different times to try to prevent the data forwarding interruption problem. This occurs when a node upstream of the transmitting node is unaware of the ongoing transmission and goes to sleep, breaking the transmission route and causing delays.

There are other synchronous duty cycling schemes, however, they operate in similar ways to the methods discussed. These techniques impose significant additional complexity in any solution that uses them and the requirement that there be some form of very accurate synchronisation system for all nodes. Time synchronisation typically requires additional hardware or a large amount of communications overhead to maintain accuracy which results in a higher power consumption and potentially increased cost. The benefits are largely in a low end-to-end delay which for many applications is not of great significance.

2.5.1.2 Semi-Synchronous Duty Cycling

In semi-synchronous duty cycling systems, nodes are grouped into clusters in which nodes are synchronised. The clusters interact with other clusters asynchronously. This simplifies the synchronisation process as only neighbouring nodes need to synchronise with each other and thus removes the multi-hop error associated with most time synchronisation protocols.

S-MAC[200] makes use of very loose synchronisation achieved by simply letting nodes exchange timestamps. Nodes also exchange schedule information, indicating when they will next go to sleep. Upon receiving a schedule broadcast, a node will either set its own schedule to the new schedule, or combine the new schedule with another schedule. Thus "virtual" clusters are formed containing nodes with the same schedules. Some nodes may follow multiple schedules and therefore be in two clusters. Maintaining two schedules is not essential to operation and can be a waste of energy. S-MAC also features long fixed sleep and active periods, 300 ms, which makes for long delays, multiple seconds, in multi-hop networks. While tolerable for some applications, for others it would be a major problem. T-MAC[182] improves the power consumption by allowing the duty cycle to be adaptive, with a node going to sleep as soon as it detects there is no traffic. T-MAC makes the assumption that a wireless sensor network should be able to cope with the highest expected load. However, when the load is lower than this maximum, the amount of time spent consuming power should be minimised. By enabling nodes to modify their duty cycle, this and a number of other improvements such as prioritising nodes with full transmit buffers are achieved. Under homogeneous load T-MAC achieved similar power savings, under variable loads T-MAC outperformed S-MAC by a factor of 5.

In the LEACH[68] protocol nodes elect a "cluster head" which is responsible for scheduling nodes in its cluster and aggregating data. LEACH also randomizes the choice of cluster head, as the role is likely to require a higher energy consumption. By rotating the role of cluster head, the extra energy consumption can be distributed across all nodes. The use of CDMA codes allow nodes from different clusters to be active and transmitting at the same time to minimise collisions. Experiments showed that LEACH extended node lifetime by as much as 8 times when compared to a direct transmission protocol.

Semi-synchronised duty cycle approaches are less complex than fully synchronised approaches. The synchronisation is typically less strict however the complexity of the methods as a whole is still fairly high.

2.5.1.3 Asynchronous Duty Cycling

One of the first widely used asynchronous duty cycling approaches was the B-MAC[145] protocol. In B-MAC a very low duty cycle is used, with the transceiver being awake for just long enough to determine if there is any activity on the channel and if not it returns to a low power sleep mode. If there is channel activity, the transceiver is kept awake to receive the data. To transmit a data packet the transmitting node first transmits preamble data for a period equal to the duty cycling period. This guarantees that the intended target will power up its transceiver, detect the preamble and be ready to receive when the transmitting node finishes the preamble and transmits the packet data. This results in a protocol in which the transceiver could be powered for less than 1% of the time, while maintaining a very high packet delivery rate. X-MAX[20] improves on B-MAC by using a number of small addressed preamble packets rather than one long preamble resulting in a number of benefits. By addressing the preamble packets, a node that detects channel activity can also detect whether it is the intended target and if not return to a low power sleep mode. This helps alleviate the "overhearing problem" caused by B-MAC. The preamble transmission can also be interrupted with an acknowledgement from the intended recipient as soon as it detects a preamble packet. This can save the transmitting node a considerable amount of energy. A variation of X-MAC used in the UPMA (Unified Power Management Architecture for Wireless Sensor Networks) [97] in TinyOS uses the data packets as the preamble packets. Packets are transmitted repeatedly for one duty cycle period or until the receiving node acknowledges the packet. This greatly simplifies the transmission process.

Another asynchronous approach is to let the receiver initiate communications. A receiving node transmits a beacon indicating it is able to receive data. A node wishing to transmit will remain awake and wait for this beacon. PTIP [33] is one example of such a approach but assumes that all nodes are only 1 hop from the base station. In PTIP a node transmits a beacon and the base station responds with any data that is queued for it. If there is no response within a certain amount of time the node returns to sleep.

RI-MAC[69] improves on PTIP by reducing power consumption and allowing operation in multi-hop networks. The transmitting node in RI-MAC waits for a beacon from the node it wishes to transmit to. As soon as it receives the beacon, it transmits the data frame back as a response. This requires pre-loading the packet into the transceiver's transmit buffer to minimise the response time. The receiving node acknowledges the reception of the data packet with another beacon, thus allowing the whole process to repeat. The receiving node only has to stay awake long enough to transmit the beacon and wait a few milliseconds for the response. The amount of time the receiving node will remain awake is adjusted based on the number of channel collisions and is included in the beacon transmission. The time between the transmission of beacons is randomly chosen from the range $L \times 0.5 < beaconInterval < L \times 1.5$, where L is the duty cycle period, to help reduce the number of beacon collisions. PW-MAC [177] uses a pseudo random sequence to generate wakeup times allowing a transmitting node to power up its transceiver just before the receiver transmits its beacon indicating it is ready to receive data.

One approach suggested to conserve power is to only turn on the radio when it is required. To achieve this a second very low power radio, in the nW range, is added to a node. This radio is always on and a received signal on the low power radio causes the normal radio transceiver to be powered up and communications initiated. A receiver with a low enough power consumption to be useful has not been achieved. An alternative is to design a radio module that can be woken by the presence of a strong enough radio signal in a similar manner to RFID tags. Both of these technologies require that nodes be physically close to work. STEM[160] uses two radios at different frequencies to allow nodes to signal to each other that they wish to communicate at the same time as other radio traffic. A preamble sampling approach is used on the signalling radio and the added cost of a second transceiver is considered acceptable given the benefits. Most modern radios allow the modification of the transmission frequency and as such there is no reason why this approach could not be implemented on one radio module.

Finally a random approach to duty cycling can be taken. The RAW[143] protocol is an example of this, making the assumption that the network will be densely populated enough that there is a statistically good chance that other nodes will be awake when a node wishes to transmit. In a dense enough deployment there will be a number of different paths from the transmitting node to the receiving node. The authors considered the use of hop based or geographical distance based metrics when considering where to route packets. Simulations showed that in a sufficiently dense network the RAW protocol consumed 80 % less energy than a solution that did not use power management.

2.5.1.4 Summary of MAC Duty Cycling Schemes

The duty cycle of most of the approaches discussed above is dependent on the number of transmissions a node receives or transmits. The high power consumption of the radio module results in the power consumption increasing as the communications load increases. It is clear that how a sensor node accesses the transmission medium will, in many cases, be the most significant factor in power consumption. This impact is reduced if the sensor node has a high power consumption even with the transceiver in sleep mode. This may be the case in applications where the sensors require constant power consumption, for example a sensor that requires precise thermal control. Ideally a solution that does not need extra hardware and is simple and easy to debug in the field is desired. This lowers the cost of the device, the complexity of control and programming and allows more efficient deployment. These requirements exclude synchronised schemes and many semi-synchronised schemes. Huang et al [75] suggest that “Recently, the responsibility of establishing communication is gradually shifted from the sender side to the receiver side” and that synchronised schemes “are very vulnerable to interference”. Most asynchronous schemes are simple and require no additional hardware. X-MAC and B-MAC have been used in a number of large real world deployments and are therefore considered to be a good starting point. As the deployment scenario has scope for energy harvesting, it is not necessary expend significant effort in finding the lowest energy solution.

2.5.2 Adaptive Power Consumption Schemes

Situations where a node can harvest power in some manner allow for a variable or adaptive power consumption scheme or intelligent use of available energy. Periods of high energy availability can be exploited by recharging batteries or transmitting stored data. When less energy is available, transmission frequency can be reduced, with more data stored, and battery reserves used. This may allow a node with a large energy reserve or input to alleviate the power consumption of other nodes by, for example, performing extra packet routing. A brief overview of some of the techniques used for adapting a node's behaviour to the available energy are presented below. A wider discussion can be found in [171, 190].

There are many forms of energy that are available to be harvested by wireless sensor nodes: solar, wind, RF, hydro, kinetic and thermal. Each source has benefits such as ease of use and drawbacks such as the amount of energy obtained. The deployment locations and desired application are usually the deciding factors in which energy sources are available. Many of these sources are available in environmental monitoring and simply required the appropriate hardware. A discussion on types of energy harvesting is therefore omitted although a good discussion can be found in [171].

Jiang et al [83] present Prometheus, a system that uses a super-capacitor, battery and solar panel to provide power for a sensor node. The super capacitor is used as a primary buffer for the solar panel as it can withstand frequent charging and discharging. If the super capacitor contains enough energy, the battery is charged. The battery is used when there is little or no solar energy. Modification of the node duty cycle, to make the most of the available power, is briefly presented. The work done by [46] follows a similar approach using a solar panel, super capacitor and battery to power a Bluetooth wireless sensor node. A modular approach to energy harvesting, super-capacitor storage and battery charging can be found in [140]. An alternative approach is shown in [74, 94] which uses an energy prediction model to estimate the amount of solar energy that will be received by a sensor node. This then allows for the node's duty cycle to be modified to make use of both the predicted and actual amount of energy generated at that particular moment.

An approach to energy aware task management based on a user defined energy policy is presented in [84]. High level goals, such as 'the network lifetime must be greater than a year', are specified and used with the current nodes energy level to change the rate at which tasks that consumed energy are run. Simulated results showed that nodes using this task management approach were able to last for the specified amount of time whilst modifying their behaviour to suit their energy state.

The IDEALS/RMR[1] [128] power management scheme lets nodes decide the importance of data and only participate in packet routing or data transmission if the data is considered important enough. A rule database is used to hold user defined rules specifying the importance of data. The importance of a particular piece of data is then compared to a metric that considers a node's energy reserves and energy harvesting state. If the priority of the data is higher than the energy metric, it is transmitted. Ibrahim [78] presents the AIRT[2] scheme which instead of not transmitting data as in IDEALS/RMR, alters the transmission power of a particular packet based on the importance of the data and a node's energy level.

The BiSNET middleware platform [14, 15] takes inspiration from the behaviour of bees to perform adaptive duty cycle control and adaptive data transmission for forest fire detection. Each node has agents that respond to changes in some sensor value. Agents are able to produce notional 'energy' for its host node which when runs out causes the node to enter a sleep mode. The length of this sleep mode is dependent on levels of 'pheromone', also produced by the agents. The resulting behaviour is that as a parameter being monitored changes quickly, the sleep length of the node is reduced and power consumption increased in favour of increased sensing rate. When the environment is stable the sleep length increases and power consumption is lowered.

2.5.3 Summary

In an environmental monitoring wireless sensor network deployment there is likely to be energy available to harvest. Primarily solar but also wind, hydroelectric and potentially kinetic. The added cost of an energy harvesting system is unlikely to be excessive and will, in many cases provide a significant amount of energy to the point where many of

---

[1] Information manageD Energy aware ALgorithm for Sensor networks with Rule Managed Reporting

[2] Adaptive Information managed energy aware algorithm for sensor networks with Rule managed reporting and Transmission range adjustments

the above techniques are not necessary and simpler methods may suffice. This is a very attractive prospect as it relaxes the energy constraints on a sensor node for environment monitoring. This would, in turn, allow for higher duty cycles and simplified medium access. The additional complexity of a fully managed power system cannot, however, be ignored. The additional hardware required and associated cost could be considered excessive in some applications. An ideal solution would be simple and require little to no extra cost. Whilst prediction models can be of use some form of opportunistic energy usage is perhaps more desirable as accurate long term predictions are difficult.

2.6 Routing

Routing is primarily concerned with the delivery of data between destinations and hence plays an important role in determining data quality. In wireless sensor networks, due to the limited energy available in most cases, routing protocols must be designed with as low an energy consumption as possible. Routing is considered, in this work, to be as much about power consumption as it is about data quality. There is a large body of literature on the topic of routing techniques in wireless sensor networks and a number of good summaries such as [1, 185, 154, 139]. Pantazis et al. [139] consider there to be the following categories of routing protocols; flat, hierarchical, query based, coherent/non-coherent, negotiation based, location based, agent based, multipath and quality of service. This list varies from review to review, mostly consistent but occasionally categorising protocols differently or adding/removing categories. These can be considered to be ‘classical’ routing methods, as suggested by [211].

2.6.1 Classical Routing Protocols

A selection of classical routing protocols are presented below, it is by no means exhaustive and more detailed analyses can be found in the aforementioned review papers.

2.6.1.1 Flat Protocols

Nodes in flat protocols are considered to be homogeneous, all nodes are equally important and behave in the same manner. Pantazis et al [139] considers flat protocols to be divided into proactive and reactive protocols. Wireless Routing Protocol [131] is a proactive protocol which builds routing tables for all nodes in the network. TORA (Temporarily Ordered Routing Algorithm) [141] is considered reactive as routes are discovered when needed. TORA uses a 'height' gradient to determine where to route data and always routes data 'down'. This height value is recomputed locally and propagated as necessary if a link fails. AODV (Ad-Hoc On-Demand Distance Vector)[144] routing is common reactive routing protocol. Requests for a route are broadcast from a source and rebroadcast by nodes that receive the request until a node with a route to the desired destination, or the desired destination node, receives the request. A reply message is sent back to the source along the reverse path that the request took. As the reply message makes its way to the source of the routing request, routes are set up in intermediate nodes. DSR (Dynamic Source Routing) [88] is another popular reactive routing protocol that functions in a similar manner to AODV. The main difference is that it uses source routing, the whole route being known by the transmitting node, rather than relying on intermediate nodes having appropriate routing table entries.

2.6.1.2 Query Based Protocols

Query based routing protocols propagate queries throughout the network. When a node can fulfil a query it does so, transmitting data back to the base station. COUGAR [199] routing treats the wireless sensor network as a kind of distributed database, each node holding part of the data. Queries for particular types of data are generated and propagated through the network. If new queries can be merged with existing queries then this is done to prevent redundant sensing and power consumption. Query plans are then generated and sent to any nodes that will be needed to fulfil the query. This allows data aggregation and processing to be performed 'in network' and reduces the amount of energy expensive data transmissions that need to occur. One example is detecting when the average temperature in a region exceeds a certain threshold. The query plan picks one node as the data aggregator and others as data generators. The aggregating node is then responsible for calculating the average and detecting if it exceeds some threshold. If it does then this fact is relayed to the base station [199].

2.6.1.3 Hierarchical Protocols

Hierarchical routing protocols introduce the concept of layered or grouped communication. The focus is often in how to form layers, organise layers and the behaviour of nodes in layers. Hierarchical protocols are often more scalable than others and benefit from not having to maintain large routing tables. LEACH (Low-Energy Adaptive Clustering Hierarchy) and LEACH-C [68, 61] was previously discussed in terms of its duty cycling nature. It is also, however, a routing protocol. Nodes form clusters with elected, or in the case of LEACH-C specified, cluster heads which act as data aggregators and provide routing to and from its cluster. HEED (Hybrid Energy-Efficient Distributed Clustering) [202] extends LEACH and offers a number of improvements designed to reduce power consumption. HEED uses adaptive transmission power in inter-cluster transmissions and considers the amount of energy available to each node and the intra-cluster cost of communications when electing cluster heads. PEGASIS (Power-Efficient Gathering in Sensor Information Systems)[109] can be considered an extension of sorts to LEACH and makes the assumption that nodes are aware of the locations of the other nodes in the network. A chain of nodes is then formed using either a greedy algorithm starting from some node or precomputed and one node is randomly chosen to transmit data to the base station. Data is then transmitted from the end of the chain towards the 'leader' node and is aggregated at each intermediate node. This technique assumes all nodes can talk to base station, are aware of the global state of the network and that aggregating data is appropriate to the application. The TEEN (Threshold sensitive Energy Efficient sensor Network) [113] protocol was described for use in situations where continuous data is not needed. A 'hard' threshold is distributed via cluster leaders which is used by nodes to determine whether they should transmit data. If a sensed value exceeds the threshold then the node transmits this data to its cluster leader. If subsequent sensed values exceed the hard threshold by more than a 'soft' threshold, then they are also transmitted. The Adaptive Period TEEN [114] protocol added more functionality such as the ability to query the network for historical, current and persistent data.

2.6.1.4 Location Based Protocols

Location based protocols use a node's location as the method of addressing. As a result nodes that use location based routing protocols must have location information available to them. GAF (Geographic Adaptive Fidelity) [197] uses a grid based system to attempt to find nodes that are considered equivalent in terms of packet forwarding. Each node in a grid location can communicate with each node in an adjacent grid cell. The end goal is to have only one node in each 'grid' location on at any time so as to conserve energy. GEAR (Geographic and Energy-Aware Routing) [203] also assumes that each node knows it's location using either GPS or some other localisation method. Heuristics based on the amount of energy available to each node and it's location are used to route packets towards a target region containing the target node. Once the data reaches the target region it is disseminated using a method called recursive geographic forwarding, where the packet is forwarded to subdivisions of the target region. Nodes using the MECN (Minimum Energy Communication Network) [150, 105] protocol use a local search to build a minimum power topology. Each node determines its neighbouring nodes and creates enclosures that are considered to contain all nodes that can be reached with minimum energy. Any nodes that lie outside of this enclosure, referred to as the 'relay region' are more efficiently transmitted to via a node inside of the enclosure. By combining this knowledge with that of surrounding nodes a strongly connected graph can be formed.

2.6.1.5 Data Centric Protocols

Nodes using DD (Directed Diffusion) [80] routing generate data in attribute-value pairs. Interest for a particular piece or type of data is diffused from the node that wishes to have the data throughout the network. This forms an interest gradient that leads towards the node interested in the data. Multiple gradients can co-exist for different pieces of data leading to different nodes. Nodes determine if they have received any interest and if so generate data to fulfil the interest and transmit down the interest slope via multiple paths. At each intermediate node in a route the data can be dropped if the intermediate node has not received any interest for that data. Certain routes are then reinforced for better performance. The SPIN (Sensor Protocols for Information via Negotiation) [67] protocols make a number of improvements to classic flooding protocols. While flooding is able to distribute data around a network, it typically does not do so in the most energy efficient manner. SPIN uses negotiation between nodes, meta-data associated with data and energy awareness to improve efficiency. A three stage handshake protocol is used whereby a node advertises that it has new data to it's neighbouring nodes. Any nodes that do not already have this data and would like to receive it send a request back to the advertising node which replies with the data. The nodes that have just received the new data then repeat the process with their own neighbouring nodes and thus the data is disseminated throughout the network. SPIN-2 allows a node to reduce participation or opt out of participating in the protocol as its energy level approaches some predefined threshold.

2.6.1.6 Summary of Classical Routing Protocols

There are a wide range of routing protocols available with some more suited to an environment monitoring scenario than others. Many protocols have set up phases, require route maintenance or hold cluster head elections. An initial set up period is acceptable as network deployment is likely to be a time consuming process anyway. Protocols that need to transmit large amounts of control messages may not be suitable for a environmental monitoring scenario. The flow of data is likely to be asymmetric, mostly flowing to base station and the topology of the network, aside from failures, is likely to remain fixed. As a result some classes of routing protocol, such as hierarchical, are more suitable than others. A proactive routing scheme would most likely be the best solution, however the storage requirements may rule it out on the types of available sensor node hardware.

2.6.2 Non-Classical Routing Protocols

Zungeru et al [211] provide a review and set of benchmarks for a large number of routing protocols used in wireless sensor networks including many of the classical protocols described in Section 2.6. The authors consider there to be a small number of broad categories of routing protocols and in particular focus on classical routing and swarm intelligence routing. These are further subdivided into data-centric, location-based, hierarchical and QoS-aware protocols. The majority of swarm intelligence based routing work uses ants, and ant colony optimization [38], as their inspiration although some consider bee colonies instead [55].

Swarm intelligence routing inspired methods are decentralised, removing the need for prior knowledge of the network. Swarms are typically composed of simple autonomous individuals but the resulting system as a whole exhibits adaptivity and resilience. The foraging behaviour of ants and bees in particular has been used as the inspiration for a large number of routing protocols as well as the flocking behaviour of birds (Particle Swarm

Optimization). In particular, the foraging behaviour of bees and ants does actually solve a routing problem. The problem of routing individuals along paths to sources of food and back to a central point (nest or hive). It is considered to be a strong analogy for routing protocols. The individuals from the these analogies are considered to be packets moving around a network. By communicating with each other at nodes, information on paths and routes can be shared.

2.6.2.1 Ant Colony Inspired Routing

Ant based routing protocols have found use in IP networks (AntNet) [34] but more recently have been applied to wireless sensor networks where power constraints are more limited. For most of the classical categories of routing protocol, as discussed in Section 2.6, there is work on one or more ant colony routing protocols. A good overview of swarm intelligence and ant based routing work can be found in [154]. Typically ‘ants’ are sent from a source node into the network looking for a particular destination node. At each node the ant decides which node to travel to next, based on a probability distribution over available neighbouring nodes. Once the destination is found, the ant backtracks to the source laying down pheromone as a marker for other ants.

The Ant Routing[206] protocols provide three protocols based on AntNet. The first, Sensor-Driven Cost-Aware Ant Routing, equips the ants with ‘sensors’ to enable them to choose lower cost paths. In addition to discovering its neighbours a node will associate a cost with transmission to each one. This cost affects the probability of a route being chosen by an ant. The second protocol, Flooded Forward Ant Routing, floods ants into the network using broadcast transmissions. This attempts to solve the problem of a lone ant randomly traversing the network in search for a destination, a potentially lengthy process in a large network. Finally Flooded Piggybacked Ant Routing is designed for dynamic networks and combined the data and the ant into one packet. This results in a higher energy consumption, due to the increased packet size, but is effective in data delivery. EEABR (Energy-Efficient Ant-Based Routing) [188] uses a ‘colony of artificial ants’ which travel through the network and look for paths between the source and destination node. At each node an ant will choose which node to visit next based on a probability determined by the node’s energy level and the amount of pheromone present on the link between the two nodes. When an ant reaches the destination node, it backtracks along the route it took to the source node whilst depositing pheromone at each node. By repeating this process the most energy efficient routes are found. Bio4Sel (Biologically-Inspired Optimization for Sensor Networks)[32] uses similar methods to Ant Routing and EEABR. It consists of a bootstrap phase in which nodes determine their neighbours and their neighbours energy levels and consumption. An initial route discovery phase is then performed with ants sent out to discover the shortest routes from each node a base station. Neighbouring nodes with longer routes to the same base station have the pheromone associated with them decreased, in order to decrease the likelihood that they will be used. Lastly a data transmission phase is started wherein data is sent towards the base stations by probabilistically picking nodes based on their energy level and pheromone level. As packets travel towards the base station pheromones are deposited along the route they choose, reinforcing these routes. The pheromone associated with a neighbouring node with a lower energy level dissipates faster than one with a high energy level, thus balancing the energy consumption of the whole network. The energy level of neighbouring nodes is recalculated when sending or receiving to or from that node and is occasionally updated with the actual level.

2.6.2.2 Bee Inspired Routing

Bee inspired protocols are largely very similar to ant inspired protocols. An example of this is the BeeSensor [155] protocol. BeeSensor uses four types of 'bee' agents: packers, scouts, foragers and swarms. The packer is responsible for taking data packets and finding an appropriate route for them. Scout bees, like many ants in ant based protocols are divided into forward and backward types. Forward bees travel from a source in search of a destination using a broadcast transmission. Backward scouts travel back from the destination to the source to report on the route quality. Foragers carry the data from source to destination along a route generated by the scouts. Unlike ant based protocols however, the route is fixed with intermediate nodes making no routing decisions other than to use their routing table to forward the forager (data). As a forager represents a route from source to destination, it is possible for a node to run out of foragers and therefore loses its path to the destination. Foragers attempt to return from the destination once they have delivered the data, if they fail to do so, the route at the source node eventually runs our of foragers and is abandoned. This allows bad routes, due to failure, to cease to be used.

2.6.3 Summary

There are other swarm intelligence based routing techniques, such as PSO [103], which use similar analogies to the above. These protocols are typically good at extending network lifetime by distributing the energy consumption load as they change. The analogy does occasionally cloud the actual workings of a particular technique but largely fits the problem well. These techniques have also been applied in other areas of wireless sensor research, for example in clustering [161].

In an environment monitoring application these techniques are likely to be of interest. They consider energy consumption and are able to distribute the energy consumption load over multiple nodes to help extend the networks lifetime. The protocols are not overly complicated but may still be more complex than is desirable when it comes to implementation on real hardware. The diffusion of the pheromones in particular is an interesting concept not dissimilar to the way that hormones work, see Section 2.8.1.

2.7 Data Quality in Wireless Sensor Networks

The majority of real world sensor network deployments, many of which are discussed in Section 2.4, use a carefully engineered control system. This is often due to the fact that the sensor network is gathering data that is valuable and therefore care is taken to ensure that the data collection will be successful by creating fixed engineered solutions. As a result much of the work on intelligent or adaptive solutions is never deployed for significant periods of time in a real environment. In this work we do not consider data quality from an information theory stand point, such as Shannon entropy [162] or Kolmogorov complexity [99]. Whilst this is a valid approach to take, we consider that it may be difficult to test in a real deployment and that there are other important aspects to data quality that are worth considering. If we consider that each measurement taken by a sensor is worth transmitting to the base station then the percentage of successfully delivered packets will impact the quality of the data; fewer packets delivered results in a lower quality of data. Typically areas that focus on this problem are routing and clustering. Clustering attempts to group nodes together under the command of a cluster head. This cluster head node assumes responsibility for managing nodes in its group, forwarding data and thus ensuring data quality. Routing is concerned with how to deliver data through a series of intermediate nodes to a target location.

Data quality is also application specific, with different applications having different demands and thus weighting aspects of data quality differently. In the context of long term environment monitoring, factors such as end-to-end delays or time stamp inaccuracies are often not the important aspect. Environments and thus environmental parameters typically change slowly and small delays in data arrival are tolerable and in some cases unavoidable due to the location of the target environment.

2.7.1 Time Synchronisation

As discussed in Section 2.5.1.1 accurate time synchronisation is a key issue in a number of duty cycle MAC protocols, especially those that are based on TDMA. As a result it has a bearing on node power consumption and also network lifetime. Time synchronisation is also crucial aspect in data quality for a number of sensing applications. Target tracking or localisation techniques may use acoustic based localisation which requires tightly synchronised time between nodes. If nodes were synchronised to within 100 μs, they could theoretically calculate the distance to a sound to within a few cm. Where the timing or propagation characteristics of an event are important time synchronisation, again, is required. Applications in which this may be important could be the monitoring of volcanoes, glaciers, buildings, earthquakes and many more. In these applications, the accuracy of time synchronisation is of the utmost importance. In the case of monitoring specific events it may be beneficial to have accurate timestamps for the time at which each node detected the event although depending on the application the necessary accuracy may vary.

There are a number of time synchronisation techniques that are widely used. One of the first was TPSN (Time-Sync Protocol for Sensor Networks) [52] in which every node in the network is assigned a level indicating their distance from the time reference source through the generation of a spanning tree. Pairwise synchronisation is performed using the aforementioned hierarchical structure, starting with the nodes with the lowest level.

Pairwise synchronisation is achieved by an unsynchronised node, node A, transmitting a synchronisation packet to an already synchronised node, node B, and receiving a timestamped packet back. By time stamping the packet when; it leaves node A, is received by node B, leaves node B and is received by node A the clock drift between nodes A and B can be determined. The propagation delay, less than a microsecond over distances of a few hundred meters, is negligible compared to all other possible inaccuracies. Initial results showed that TPSN based synchronisation between a pair of nodes was able to achieve an average error of 16.9 μs.

RBS (Reference Broadcast System) [47] is another time synchronisation protocol and was developed around the same time as TPSN. In RBS, one node transmits a reference broadcast packet which is then received by multiple nodes. These nodes time stamp the packet as it arrives, using their local clock, and exchange this information with their neighbours. This comparison allows nodes to calculate their time offset to other nodes and by repeating the procedure, clock skew can be estimated and accounted for. Modifications to allow RBS to work in multi-hop networks where there are multiple nodes that provide reference broadcasts and provides the ability to synchronise to an external time source, such as a GPS module.

Perhaps the most widely used method is FTSP (Flooding Time Synchronisation Protocol) [115]. In FTSP, a node's current estimate of the global time is broadcast to all nodes in range. Upon receiving this packet, the receiver will obtain their current local time. By carefully profiling the time taken to transmit and receive the packet average, errors of 1.4 μs were achieved. FTSP also works in multi-hop scenarios through the use of what the authors call "reference points" which consist of a global time and local time that refer to the true time.

More recently, GTSP (Gradient Time Synchronisation Protocol) [168] was proposed which suggested that it is more important for nodes to be closely synchronised to their neighbours than to some global time. Experiments showed that GTSP was performed better than FTSP in terms of synchronisation between physically close nodes.

All of these techniques can achieve synchronisation of, usually, a few tens of μs in a real world multi-hop network. However, these methods are not low power, they require the transmission of a large number of packets and by using high frequency clocks, often the 8 MHz system clock, nodes cannot enter the most effective power saving modes. While most node hardware platforms offer some RTC (Real Time Clock) functionality this is almost always driven by a 32 kHz crystal which can offer, at best, 30 μs resolution. Therefore the time synchronisation performed using these clocks is less accurate.

VHT[159] provides a solution to this problem by using both high frequency, 8 MHz, and low frequency, 32 kHz, clocks and performing synchronisation between the two. This allows a device to enter low power modes, during which the low frequency oscillator is left running, and use the high frequency oscillator to provide a higher resolution timer when the node is awake.

2.7.2 Anomaly Detection

Due to the high energy cost of transmitting data to a central location, identifying anomalous or redundant data and behaviour is important. There are a number of possible attacks that can be performed on wireless sensor networks affecting the quality of the data returned. Even if the sensor network is not directly under attack in long deployments there are likely to be equipment failures due to hardware degradation or software bugs. These nodes may behave erratically and in unexpected ways that are detrimental to the performance of neighbouring nodes and the network as a whole. A survey on some of the techniques used in anomaly detection can be found in [148]. Lim et al [108] make use of an immune system inspired method, called RDA (Receptor Density Algorithm), to classify types of radio interference experience by a node and react accordingly. By reacting differently, depending on the type of interference it is experiencing, the node is able to reduce the amount of energy expended by unnecessarily performing route discovery whilst increasing the number of packets that were successfully received. The work of [204] also uses the immune system, this time focussing on Danger Theory and the innate immune system for intrusion detection. They use a wireless sensor network, running directed diffusion routing, as a test bed for the intrusion detection system. While a positive result was achieved, some of the assumptions made in the simulation, such as all-to-all communications, may not hold up in a real deployment. Wallenta et al [187] also use a Danger Theory inspired algorithm based on the Immune System algorithm called DCA (Dentritic Cell Algorithm). Interest cache poisoning in the directed diffusion routing algorithm is the target of the anomaly detection. A number of danger and safe signals are generated by the modified directed diffusion algorithm which are then classified using the DCA. The system was both simulated and tested on a small deployment of 10 nodes and was able to detect instances of cache poisoning attacks.

2.7.3 Summary

When a large number of sensors are deployed managing the data and avoiding duplicate, unnecessary or incorrect (anomalous) data is important and thus impacts data quality.

Data quality can be improved in a number of ways:

- By ensuring that the data collected by a node is good data.
- By ensuring that data is delivered to its destination reliably.
- By ensuring that nodes and the network survives for its intended lifetime.
- By adapting the behaviour of the node and/or network to the application domain.
- By preventing the collection and transmission of bad (incorrect or unnecessary) data.

Achieving this in a decentralised manner is important, especially in network deployments in harsh conditions that can cause node failure, unpredictable or undesired behaviour in nodes. Often the concept of data quality is closely related to the type of data being collected, the particular environment the network is deployed in and what the data will be used for. Therefore, techniques that aim to improve data quality in wireless sensor networks as a whole should be able to adapt to different deployments, situations and use cases. The real world is also an ever changing and often unpredictable place. As a result, even in a single specific deployment, techniques that aim to improve data quality must be able to adapt. The original concept of wireless sensor networks considers networks of 1000s of devices and the ability for a method to scale is important as we are gradually moving towards larger and larger deployments. There are parallels between wireless sensor networks and many natural and biological systems. Ant and bee colonies, bird swarms, the immune system, the human brain, natural selection and others have provided the inspiration for a large quantity of work [102]. Much of this work is aimed at improving the quality of data and reducing power consumption in some way. These biological and natural systems are often seen express a number of desirable behaviours such as resilience, self-healing, autonomy and adaptivity.

2.8 The Endocrine System

The endocrine system is considered to fit the domain of wireless sensor networks well, in particular its distributed nature and ability to adapt while maintaining control of a number of vital parameters. In this section, the basic functioning of the human endocrine system is presented and briefly discussed. Work that has developed and made use of artificial endocrine algorithms or techniques, mostly in the robotics domain, is described. The relationship between wireless sensor networks and the endocrine system is outlined with a discussion on existing endocrine or hormone inspired work in the field of wireless sensor networks.

2.8.1 Hormones

The concept of 'internal secretions' originated from the French physiologist Claude Bernard's work on the physiology of the liver in the 1850s in which he discovered that the liver was able to produce glucose and not simply store it. It was previously believed that glucose in the blood was purely a result of eating food. It was one of the first hints at the bodies ability to self regulate. In 1895, Edward Shäfer showed that diabetes could be induced in dogs by the removal of their pancreas and that a pancreatic graft could prevent diabetes [157]. He concluded that the pancreas must be secreting something into the blood that prevented excessive glucose production. Shäfer also worked on the extraction of epinephrine from the adrenal glands of various animals and showed that an injection of epinephrine in a dog produced a large rise in blood pressure and heart rate [136]. Ernest Starling introduced the word "hormone" in 1905 in a lecture on the chemical functions of the body [169]. The word hormone comes from the Greek *hormone*, meaning to 'set in motion'. Of this lecture [27] says "This important lecture expanded Shäfer's ideas on internal secretions and developed the concept that hormones could be produced in certain regions of body to circulate in the bloodstream as chemical messengers to other regions, where they act at specific target sites to regulate the metabolic needs of the whole organism". This high level description still largely holds true today.

There are primarily three types of hormones: steroid-based, peptide-based and amino acid-based. These groups of hormones are structured differently and as a result the way they are produced, transported and affect cells varies. Peptide based hormones are water-soluble and so dissolve well in blood plasma, whereas steroid based hormones need to bind to carrier proteins for transport. Steroid hormones can, however, enter cells through their membranes to interact with receptors inside the cell. Peptide based hormones cannot penetrate membranes and must bind to receptors on the cell. The mechanism for this cellular chemical response to a hormone binding was discovered by Earl Sutherland in the 1960s when he showed that epinephrine used a 'second messenger' called cyclic adenosine monophosphate to affect a receptive cell [172]. Different types of hormone are also stored differently, peptide based hormones are produced and then stored in the cell that secretes them whereas steroid hormone producing cells store cholesterol which is then used to create the hormone.

2.8.2 The Endocrine System

The body contains two main regulatory systems, the endocrine system and the nervous system. The nervous system is fast acting while the endocrine system reacts more slowly and its effects generally last for longer periods of time. The nervous system sends signals directly to a cell and uses neurotransmitters to bridge the gap between nerve cells. In the endocrine system, hormones enter the circulatory system and are distributed throughout the body. As a hormone circulates through the bloodstream it can bind to receptors on or in cells that are receptive to that type of hormone. This allows multiple hormones to coexist in the bloodstream without interfering with each other. Once bound, the hormones trigger a chemical response in the target cell. This response is often an increase or decrease in the production of some substance, for example the hormone glucagon stimulates the production of glucose in the liver.

The endocrine system is the system of hormones and hormone producing glands as well as mechanisms of their production, transport and interactions within the body. The classical view of the endocrine system is that there are a number of discrete hormone producing glands such as the: hypothalamus, pituitary, thyroid, parathyroid, adrenals, pancreas, ovaries or testes, kidneys and the gut. There is, however, evidence of hormone producing cells being scattered throughout the body for example in the skin and whole gastrointestinal tract. These are considered to be part of the ‘diffuse endocrine system’ that produce hormones but ‘do not form a discrete endocrine gland’ [73]. In addition to this while classically hormones travel through the bloodstream to distant cells some hormones act locally, affecting different cells in the tissues which produce the hormone which is termed *paracrine*. Other hormones affect the same type, including the same cell, that produces the hormone. This is called *autocrine* action. Hormones can have a mixture of actions [73].

Hormones are considered to have a certain biological half life due to being metabolised or excreted. Some hormones, such as most peptide based hormones, have a short half-life of a few minutes. Others, such as cortisol, have a much longer half life of 90 minutes [73]. It is also possible to have inactive reserves of hormone in the blood stream; steroid based hormones that are bound to a carrier protein are considered to be biologically inactive until they unbind.

There are a number of patterns of hormone secretion. As the endocrine system is key to homeostasis in a number of systems in the body, episodic secretion is quite common. This is often to try to achieve a set point in the level of a system. For example, the regulation of blood glucose or blood potassium level. When these deviate from their set level, hormones are released to bring their concentration back to the 'right' level by stimulating or suppressing various systems. Some hormones are released frequently in bursts and their release is not due to any particular stimulus. Other hormones are released in some form of diurnal cycle, with levels increasing or decreasing at specific times of day. Control of when these hormones are released is usually by the body's 'internal clock' in the hypothalamus. Lastly some hormones, such as thyroxine, are kept at a set level [73, 27].

2.8.3 Homeostasis

The concept of homeostasis was first articulated by Claude Bernard [12] in the 19th century. Bernard described the necessity of maintaining a stable environment in the body. He considered that different parts of the body were responsible for the controlling some variable to maintain its stability. In the 1930s Walter Cannon expanded upon this idea and suggested the word homeostasis, meaning tending towards stable equilibrium, to describe this behaviour [21]. Homeostasis can be described as the bodies ability to keep a system at a stable level, despite perturbations, conducive to the optimal functioning of cells.

There are a large number of parameters that are kept in homeostasis in the body such as blood sugar, salt levels, body temperature and blood pressure. The endocrine system plays an important part in the homeostasis of these parameters. While individual hormones can affect many systems in the body, they rarely work alone. Many hormone based regulatory systems work in 'cascades' where one hormone stimulates the release of another hormone which in turn effects a change in some system [63]. This allows a form of amplification to take place where a very small amount of one hormone stimulates the release of a much larger amount of some other hormone. To prevent these systems from running out of control, the endocrine system employs negative feedback. The simplest form of this is that the final product in a hormone cascade inhibits the release of hormones further up in the cascade, although intermediate hormones in the cascade can also have the same inhibiting effect. Depending on the mechanism the inhibitory effect can be relatively quick, minutes, or slow, hours or days.

Lastly, many bodily parameters that are important and must be kept close to a set point are controlled by multiple hormones in an antagonistic manner. In the simple case of two hormones, one acts to increase the level whereas the other acts to decrease it. An example of this is insulin which suppresses the production of glucose in the liver and glucagon which promotes glucose production. There are also cases where hormones act together to produce a stronger effect [73].

2.8.4 Artificial Endocrine Systems

In the 1950s Ross Ashby built a machine called the homeostat, comprising of four units containing pivoting magnets the angle of which he considered to represent a variable that needed to be maintained. Each of the four units output a signal that was proportional to the deviation of the magnet from the centre position. Each unit had three inputs which were connected to the output of three other units. Each input to a unit was connected through a commutator and a potentiometer to a coil that could affect the position of the magnet. The potentiometers and commutators could be set to random positions by each unit if the magnets position exceeded more than 45 degrees. The system exhibited what Ashby called 'ultrastability' in keeping the magnets in the central position. When the positions of the magnets was perturbed the system attempted to bring them back to a central point, changing the input potentiometers and commutator settings if necessary. This is considered to be the first example of an artificial homeostatic system.

More recently hormones have been used for task allocation. Brinkschulte et al [16, 17] showed that in a grid processor a hormone inspired task allocation system worked well whilst being decentralised and possessing the ability to self heal and reassign tasks.

2.8.4.1 Hormones in Robotics

The field which has made the most use of the the concepts of homeostasis and the endocrine system is robotics where it has been used in a number of different ways. As some of the issues facing robots are similar, such as task selection and power control, to those faced in wireless sensor networks a summary of this work is provided. In the early 1990s Arkin [5, 4] noted that robots must be able to adapt "its actions and plans, as opposed to continuing in a manner that will lead to its ultimate demise". He considered that in the human body most of this self monitoring and subtle changes in behaviour were not conscious decisions but performed by the endocrine system. Robots, according to Arkin, should be able to dynamically re-plan, taking into account their internal state and the state of the environment. Arkin modified his AuRA robot control architecture by adding hormone receptors to the motor schemas produced by the path planning module. When the robot's fuel was running low, hormones were released that caused the robot to take a shorter but riskier path to its destination consuming less fuel in the process.

Brooks [19] used the work of [100] to extend his subsumption control architecture to use hormones for the purpose of integrating multiple behaviours in a single robot. Any process running on the robot could excite a condition, such as panic or drowsiness. A condition could be excited by multiple processes and the excitation level decayed linearly over time. 'Releasers' summed a set of conditions and are comparable to a hormone value. They were given names such as adrenaline or sleepine. Each behaviour had an associated activation threshold, the crossing of which promotes or inhibits the behaviour. An activation level can be affected by a function of releasers thus allowing 'hormones' to modify the running of behaviours. The SOZZY vacuum robot described by [198] uses a similar technique as Brooks. It also uses the subsumption architecture and behaviours to control it. In addition four 'emotions' are used: joy, desperation, sadness and fatigue. Different emotions resulted in different behaviour for example 'joy' resulted in the robot consuming dust and exploring and 'fatigue' caused the robot to seek out its charging station as quickly as possible. Each of these emotions had an associated hormone which was produced in response to some stimuli. For example, sadness hormone was produced in response to loosing the signal from a beacon station. Joy hormone increased when vacuuming dust and decreased if they battery was low or the robot was running out of time. The hormone values were compared and the emotion associated with highest hormone was carried out. Each hormone had some stabilizing function that attempted to pull the hormone back to its original level to prevent the hormone from becoming saturated.

The homeostatic zone of viability as described by Ashby was the inspiration for the work of Avila-Garcia et al [6]. Robots are considered to have vital parameters which must stay within a certain range for the robot to remain 'alive' for example battery level. A number of motivations are used that define a tendency to behave in a certain way in response to some internal or external factors, for example hunger. Behaviours were then defined that resulted in the increase or decrease in the level of either energy or temperature, the two vital parameters. This allowed the robot to attempt to rectify the cause of the initial motivation. It was also noted that remaining well within the vital zone was more optimal than oscillating around its boundaries. A hormone was introduced based on the robots current 'risk of death' and the proximity to another robot (a competitor). The hormone affected the level at which the robot stopped consuming a resource. As a result, as the risk of death increased the robot took fewer and fewer risks, attempting to stay as close to the centre of the viability zone as possible.

The use of hormone as a component of an emotion system for robots was further explored by Gadanho et al [51]. Four emotions were used: happiness, sadness, fear and anger. The robot was also given 'feelings' such as hunger, pain or warmth. Intensity of each emotion is dictated by a group of feelings. A threshold based system was used to decide which emotion is active, if no emotion was above the threshold the robot was 'neutral'. Each feeling had an associated hormone that influenced its value therefore in turn affecting the emotional state of the robot. Each emotion produced hormones to change the feelings so as to give rise to that emotion. This allowed emotions to change the robots perception of itself and its environment. Hormone values were quick to rise but slow to decay, allowing emotions to persist.

Hormone-based control for reconfigurable robots was presented by Shen et al[164, 163] to allow dynamic reconfiguration of the robot and to allow distributed control of movement. Each robot had several attachment points for other robots, allowing them to connect and form different configurations. Communication between robots was performed through these physical connections and as a result the robots formed a network. Individual robots were capable of transmitting hormones to any neighbours, where neighbours were directly connected robots. First, a topology discovery process was implemented using hormones transmitted between neighbours. Control of locomotion in various configurations, such as a snake or a wheel, using hormones was then explored. Each hormone message had no address and how it was used was dependant on the receiving robot. As a result, the system had a similar physiology to a biological hormone system. Hormones took time to circulate through the 'body' and could be consumed in the process impacting the behaviour of the system. Robots discovered their roles based on their position in the collection of robots.

As the hormones were not addressed and topology was constantly evaluated the system was considered to be fault tolerant although it was possible for a non-functioning robot to cause failure of the whole system. The authors later introduced a Digital Hormone Model (DHM)[165] for distributed control of swarming behaviours. In the simulations presented, each member of the swarm secreted activator and/or inhibitor hormones, which were diffused over the area around it, thus affecting the behaviour of other members of the swarm. A number of scenarios such as bypassing barriers, self-healing and search and seizing a target were simulated with good results. More recently, [194] modifies DHM to allow for tasks in the environment that require a certain number of robots. A task hormone was added to allow robots to signal to nearby robots that they had found a task.

Neal and Timmis [134, 135, 183] introduced the concept of an artificial neuro-endocrine system, a combination of neural networks and an artificial endocrine system. The use of multiple biologically inspired techniques was considered important for the replication of homeostasis. In this system weights in a neural network are modified by a endocrine gland connected to the same inputs as the neural network and this allows the inhibition or excitation of the weights in the neural network. This was tested on a robot equipped with a number of sonar sensors. A neural network was trained to cause the robot to wander around an area whilst avoiding objects. The hormone level increased in response proximity to objects and decayed slowly over time. The addition of the hormone caused the robot to behave differently 'running away' from obstacles and 'panicking' when surrounded by obstacles. A similar system was shown by [127] where a robot was equipped with two neural networks; one to seek white surfaces and another to seek black surfaces. Two hormones, a black hormone and white hormone, affected their respective neural network in the manner previously described. Initially the hormone levels were controlled by a simple sinusoidal function producing a cyclic behaviour in the robot. Changing hormone production to be a function of the camera input resulted in a lack of momentum in the system and the robot stabilising in an intermediate state. Lastly, a pooling mechanism was used whereby each hormone produced was added to a 'pool' of hormone that was only released once it had exceeded a certain threshold. This induced a large behavioural change.

In addition Sauze [156] showed the viability of using a neuro-endocrine control system for long term operation in sailing robots. Hormone inspired promotion and suppression of the neural network controlling the sailing robot allowed the power consumption to be reduced while retaining sailing performance. The effect of the addition of a hormone linked to the available sun light was tested in simulation and indicated that the robot would be capable of indefinite operation. This suggests that hormone inspired power control is effective and may be applicable to wireless sensor networks as they are also resource constrained and must adapt to the environment.

This pooling of hormones was also used by [186] to select between a number of tasks on a small number of heterogeneous robots. Each task had an associated hormone pool that must be filled before the hormone was released into the system. Hormones in the system decayed and the task with the highest hormone level was carried out. Robots were able to communicate which task they were currently running to other robots. This allowed hormone levels in the hormone pools to be increased or decreased, depending not only how well they had previously performed a task but on whether another robot was currently performing the task.

In response to the inability of hard coded controllers to adapt to unforeseen and dynamic environments, Stradner et al [170] consider the need for an easily evolvable representation of a controller. The system proposed is the AHHS, Artificial Homeostatic Hormone System, controller for mobile robots. In AHHS, sensors excrete hormones in response to certain environmental conditions, these hormones diffuse throughout the "body" of the robot. Parameters controlling the behaviour of the hormones are evolved to improve performance. Different hormones interact with each other, multiplying or decreasing the level of other hormones and activate actuators resulting in a change in behaviour. As the behaviour of the robot influences the sensors, a feedback loop is formed which is regulated by AHHS to keep hormones at a homeostatic set point. The controller was implemented in simulation and on epuck robots [170]. Experiments showed that the AHHS controller was able to control real robots with limited computational resources. Further work [60, 58] showed that AHHS would be of use in multi-modular robotics [18]. In such systems individual robots can combine to form a 'super' robot that itself can reconfigure its body shape. The AHHS provides a decentralised, sub symbolic, control system that uses hormones produced by sensors to control actuation. These hormones diffuse through the connected sub-robots that form the body of the super robot. Analysis of AHHS showed that it was superior to state-of-the-art approaches in the domains examined. The evolved controllers showed an ability to generalize to other initializations and scale with the number of modules [59].

2.8.4.2 Summary of Hormones in Robotics

From the work in hormone or endocrine inspired systems in robots, it is clear that the common theme is the promotion or suppression of a system(s) or parameter(s). Hormones are frequently used when a relatively long lasting effect on some system is desired and the concept of inertia is beneficial. The fact that hormones are not ‘addressed’ is also considered useful, any system can simply express an affinity for a particular hormone with no regard to who produced it or how it was produced. In much of the discussed work discussed hormones are secreted ‘blindly’ with no knowledge of whether they are received or who has received them. In many cases this is of no importance but in [164, 163] it is entirely possible for hormones to become lost or not make their way through the whole system.

The pooling systems described in [127] and [186] offer a hormone inspired mechanism for quick and dramatic changes in behaviour. In the context of a wireless sensor network, pooling could also reduce the amount of ‘hormone data’ that would need to be transmitted among nodes. The work done on combining endocrine or hormone systems with other biologically inspired systems shows much promise, however, using the hardware available for a wireless sensor node implementing these techniques may be difficult or impossible. The low computational power, memory and existing hardware complexity favours simpler techniques.

Work done by Sauze [156] suggests that hormones and the endocrine system may be useful systems to emulate for long term power control. And, very importantly, shows this to be true in real robots running for extended periods of time.

2.8.4.3 Hormones in Wireless Sensor Networks

There is very little work on hormone or endocrine inspired systems in the field of wireless sensor networks [196]. This is despite the need for wireless sensor network systems to adapt to the environment and maintain some level of homeostasis in a number of systems. An example case is that while in some systems energy consumption must be extremely tightly managed due to a finite amount of available power, systems capable of harvesting energy should attempt to maintain homeostasis in their power levels. In the endocrine system, individual cells behave in a manner of their own choosing, they can express receptors for hormones if they have need of a particular hormone signal and can interpret the signal as they like sometimes performing an action or sometimes secreting another hormone or substance in response. This allows for individual cells that are concerned with individual goals to contribute to a large goal: to keep the body functioning and maintain homeostasis.

Autonomic Computing refers to the ability of distributed systems to self-manage and adapt to change while hiding the complexity from users. Early work by [116] in autonomic computing for wireless sensor networks suggested that some form of agent based system could be beneficial in this area. It was, however, also suggested that the computing power available in the wireless sensor network domain was unlikely to be sufficient to run a fully deliberative autonomic agent. The survey [147] of autonomic computing in sensor networks suggests that most approaches took policy-based or context-based reasoning for monitoring processes and executing 'action plans'. Also noted, is the widespread use of simulation based evaluation over real experiments. The trade-off is considered to be between precision, scalability and performance. DISON [24] is a light weight management layer that is independent from user applications and network protocols and is context aware. DISON provides a good example of a wireless sensor network system that embodies the autonomic computing ideals. A hierarchical management structure is used with a manager node being responsible for the management of a relatively small group of nodes. The framework used allows the use of policies which rely on some collection of context (information about the state of the node) to select management tasks to be run. Experiments using DISON on a real network of sensors showed that the packet delivery ratio was improved, the number of duplicate packets reduced and power consumption lowered although power consumption still too high for long term deployment. In an environmental monitoring scenario it is unlikely that the network will be required to support multiple applications and management facilities could be incorporated into the application.

Work on self-organization inspired by the cellular/hormonal cascades in the body was also been presented by Dressler [39, 40] for Sensor Actor Networks (SANETS). SANETS are similar to wireless sensor networks but include 'actors' or nodes capable of some level of actuation. The un-addressed data-centric nature of signalling cascades and specific reaction to receiving certain data in the body is the inspiration behind Dresslers work. A rule based network is created in which data is transmitted, in type-content pairs, in a similar manner to directed diffusion [80]. When data is received a simple set of rules are used to decide what to do in response. This decentralised approach was tested in simulation where sensor nodes were able to sense the temperature and actor nodes could check if those temperatures exceeded a threshold. Sensor nodes used a gossiping protocol transmitting sensor readings to neighbours based on some probability. The result showed that the system performed better than a centralised base-station approach in which the base station checked the temperature readings against the threshold and transmitted alarms to the actor nodes. The focus of this work was network-centric control of actuator nodes. In an environmental monitoring scenario data must be transmitted to some location for storage. This simple and decentralised approach where individual nodes behaviour is simple is, however, appealing.

Trumler [181, 180] considers the use of hormones for task load optimization in a network. Each of a node's resources, such as memory or CPU time, has a corresponding hormone that indicates how much of the resource is consumed. These hormones are 'piggybacked' on messages sent between nodes as part of services communication. A node receiving data will extract the hormones and aggregate them into a 'load' value and carry out the same process with its own hormones. These two aggregates are compared and if the receiving node has a higher load it will attempt to move a service to a sending node to better distribute resource consumption. A number of strategies for deciding when to transfer services were evaluated in simulation and found to work well at distributing resource consumption among nodes. The particular type of network considered in the work was not strictly defined. It is unlikely that a wireless sensor network would be able to support this type of system due to the large communications overhead.

A clustering protocol that uses hormones is presented in [37]. At the start of a cluster election phase nodes choose a random time, within the election phase, to wait. If the node receives cluster hormone it becomes a member node and transmits member hormone to its neighbours. Nodes receiving member hormone increase their internal member hormone level. Once the random amount of time has elapsed, a node will compare its member hormone with any neighbours that are still awake and if it has the largest, will become a cluster head and transmit cluster hormone. Once the cluster election phase has finished nodes will start their sensing tasks until it is time for a new cluster election phase. This technique was also tested using simulation and was shown to improve network lifetime however a thorough analysis is missing.

The use of hormones as part of a tracking strategy for wireless sensor networks is presented in [207] in order to attempt to reduce the number of nodes awake and consuming power. Nodes were capable of tracking a target within a certain range and had a 'base hormone' level. There were periodic sensing and transmission phases. A node did not attempt to track targets in the sensing phase if the base hormone level was 0, but if it was greater than 0 it did. When trying to find targets to track if none were found a node sent sleep hormone to all of its neighbouring nodes. Finding a target to track resulted in the transmission of wake hormone. A node receiving wake hormone increased its base hormone level by 1, whereas the sleep hormone decreases the base hormone level by 1 recreating an antagonistic hormone system. As a result nodes that detect a target will cause nearby nodes to also attempt to track the target. This causes nodes along the path of the object to consume energy, while the other nodes do not and do not interfere. Simulation showed that this was an effective approach although many implementation details are unclear. This work is expanded upon in [85, 86] by probabilistically sending hormones when a node cannot find a target. Sleep hormone is referred to as hypnotise hormone but otherwise both hormones act and are combined in the same manner as in [207]. When a node that is awake cannot find a target it has a chance to release both hypnotize or awake hormone, this enables a good distribution of awake nodes at all times to detect a possible target. Simulation showed that it performed better than a random waking schedule, a fixed duty cycle and the previous hormone inspired method. The applicability of this technique for discovering requests for service in an Internet of Things context is shown in [36]. The technique is the same that presented in previous work however instead of attempting to detect targets to track nodes can receive requests for services from an IoT device. By specifying the probability that hormones would be transmitted, the number of nodes awake in any round could be controlled. No attempts have been made to try such techniques on a real network of sensors. The behaviour described in this work is very interesting as it is a decentralised approach to control of nodes in a network. The ability for the system to find some stable level in terms of active nodes is of interest.

2.8.4.4 Summary of Hormones in WSNs

The primary feature of the endocrine system and hormones that seems to be used by the existing work, is the un-addressed nature of hormones. In a situation where transmission is costly and resources are limited it is understandable that this is desirable. Information, data and hormones can be simply be sent from a node and what happens to that information is up to the receiver. Given that all data sent over a radio modem is effectively a broadcast message, it makes sense to exploit this and make the content of messages be potentially useful to all recipients.

It is also clear that there is a lack of real world testing of the techniques discussed. While this is understandable due to the amount of time and cost that such activities require, at some point these techniques must be tested in a real scenario. While simulators can provide a good test bed and development environment they often hide many of the difficulties, problems and quirks of the real world which have to be dealt with or fail to model reality with sufficient accuracy [107]. A simulated target tracking system only becomes useful if it actually works in a real environment and building upon simulated work is dangerous as the validity of it has not be verified. There are also a vast range of simulators used, from well the known (such as NS-2 or TOSSIM) to home made simulators. It is often the case that not enough information is provided to replicate a simulated experiment. Kulkarni et al. [102] also find this to be true stating that many techniques remain in development or non-finalized states and never make it out to the real world.

Adaptation to the environment is an important function of the endocrine system and should be an important feature in a wireless sensor network. In particular when the amount of energy a nodes receives or expends is related to its environment adaptation is important. In addition to adaptation there may be desirable goals in terms of behaviour that should be integrated into the adaptation. Perhaps a certain fixed sensing rate is desirable whilst energy consumption is adapted to the current environment or perhaps some event is considered important enough to warrant a change in behaviour. It is these situations that the author feels are well suited to an endocrine inspired control system as the combination of hormones and behaviours is a crucial part of the endocrine system.

2.9 Summary of Literature Review

The first section of this literature review covered available hardware and software for use in wireless sensor networks. It was apparent that there are a number of existing sensor node hardware platforms. Some are more widely used than others, for example the Mica and Telos platforms. There appears to be a trend towards the design and manufacture of hardware for specific application domains. This was partly due to application requirements, but also the relative simplicity of the systems being designed. As a result many hardware platforms have poor support due low numbers of users. The benefits of using a particular existing platform may be overshadowed by the time investment required, poor support, high cost and low availability. The situation for software is a little more promising with a variety of low power software. The complexity of the best and most feature rich systems may however prevent suitable access and control a node.

Of the real world deployments reviewed, most used small, 10–20, numbers of nodes and short deployments of a few weeks or months. It was clear that simpler systems were preferred for gathering data considered valuable. Deployments testing more complex systems were the shortest or were not carried out at all in favour of simulation. Equipment failure, unforeseen issues and lower than expected performance was not uncommon in the real world deployments. The small number of big deployments, such as GreenOrbs, suggests that such an operation is expensive and time consuming but, given the results, achievable. Deployments carried out as part of this work will be financially restricted. Lowering the node cost should be a priority so that the number of nodes used can be increased.

Of the many methods of controlling power consumption, duty cycling was the most common. In particular, duty cycling the radio transceiver is carried in almost every wireless sensor network deployment. There are tradeoffs to be made in latency, throughput and power consumption. The correct balance is almost always application specific. In terms of successful implementation and reliability simpler methods, such as XMAC or BMAC, should be favoured. In an environment monitoring deployment, there is almost always energy available to be harvested. Solar is the easiest and most abundant with the only real limiting factor being the cost. Many deployments make use of solar harvesting to provide indefinite length operation. A simple and cheap solar harvesting solution would provide a way to let nodes recharge their batteries and act opportunistically when power was more abundant.

When it came to reviewing routing protocols it became clear that there are a huge number and variety of ways to route data. The variety is explained by the lack of consensus on what is the most important metric in routing performance. Among the key metrics latency, throughput, power consumption, data aggregation and reduction, adaptability, schedulability and QoS. There is also a large body of work in bio-inspired routing protocols, often modelled after the behaviour of ants, bees or other swarming organisms. To keep experimental conditions as similar as possible between experiments, a simple and controllable solution should be used.

Time synchronisation techniques are available and have been tested that can synchronise clocks to sub millisecond precision. There is an associated overhead in software complexity, power consumption and network traffic in maintaining such high levels of synchronisation. For the type of environmental sensing that is the focus of this work such synchronisation is unlikely to be necessary. In addition implementation of such systems makes requirements on the hardware platform, such as the need for accurate high frequency very low power clocks.

The endocrine system is able to regulate many systems at same time in the human body. While the specifics of how the endocrine system and individual hormones work is complex, the high level concepts are simple. It is these concepts that are considered to be most useful. Hormones are produced and released in response to some stimuli. These hormones travel to cells through the body in the process binding to cells that have receptors for them. The bound hormones produce some effect on the target cell, often stimulating or suppressing a system as a result. As a result of being metabolised or binding the level of hormones decays over time. The review of endocrine inspired work in robotics showed that these ideas show promise for integrating, combining and softly switching between multiple behaviours. Little work was found in the field of wireless sensor networks that used endocrine inspired techniques. Of the work found , non had been deployed to a real world sensor network.

One of the key lessons learnt from carrying out this literature review is that it will be vital to build, test and deploy a sensor network multiple times in the real world. This will allow unforeseen problems to be encountered and allow the hardware and software design to be refined. It is crucial that this is carried out as the amount of time required to run real world deployment based experiments will be high and thus the network platform must be tested and reliable. The review of mesh networking and existing hardware solutions indicated that Zigbee transceivers and Atmel 128 or TI MSP430 series microcontrollers are a technologies worth investigating.

Chapter 3

Preliminary Designs and Deployments

This chapter presents the results of a series of network deployments that were run whilst developing the hardware and software for the sensor nodes. They can be viewed as the design iterations that led to the final design detailed in Chapter 4. This approach was carried out in response to the literature reviewed in Section 2.4, which indicated that many real world deployments inadvertently went through this process. This resulted in unsuccessful or only partially functional initial deployments. In an effort to avoid this, some initial iterative design and testing work was considered beneficial. Each deployment was intended as a proof of concept to test whether the current design was suitable. It required several iterations of the design before a platform suitable for experimentation was achieved. The following sections describe each of these iterations and discuss why each iteration was necessary and what was learnt. The first deployment was in Greenland for a duration of around 5 days. It provided the first indications of the potential problems that would be encountered in a real world wireless sensor network deployment. The following deployments were carried out in the United Kingdom and provided an opportunity to refine the hardware, software, deployment procedures and associated tools.

3.1 Greenland Deployment

In June of 2012 an opportunity arose that enabled the temporary installation of a small wireless sensor network in a glacial fjord in Greenland. This provided an opportunity to test the chosen sensor equipment, discover deployment issues and gather some test data from the meteorology sensors. Five nodes and one base station were deployed over an area of approximately 20 m^2. Due to lack of equipment in this remote location, nodes were placed under, or fastened to, rocks to prevent them from being blown away. This, coupled with the high metallic content in the surrounding rock, severely impacted the effective range of the nodes. The result was a range of around 5 – 10 m instead of the few hundred meters possible in other environments.

Nodes were equipped with XBee transceivers and used the Zigbee protocol. Five nodes were set up as 'End-Devices', so that in the future the radio transceivers could be put into sleep modes to conserve power. The base station was configured as a 'Coordinator' and remained powered and awake at all times so as to not miss packets. The result was a star network topology with the base station in the centre. The Zigbee protocol supports a mesh network configuration where router and coordinator nodes can relay data for the network, allowing nodes to communicate with each other and improve redundancy. However, the set up used resulted in no meshing capabilities as only coordinators and routers are capable of meshing. The decision to configure nodes as end devices was due to Zigbee's requirement that router and coordinator nodes be awake, and therefore consuming power, at all times.

The weather during the time of deployment was dry with clear skies. Coupled with the time of year, this resulted in significant temperature changes between night and day.

3.1.1 Hardware

The nodes were comprised of the Sparkfun USB Weather board and XBee Zigbee transceiver. The base station node consisted of an Arduino Uno microcontroller, a Wireless SD Shield and an XBee module.

The XBee modules were 2.4 GHz Pro Series 2B, 50 mW units and used frequency matched antennas. The modules are capable, theoretically, of communicating over distances up to 1,600 m (line of sight) at 250 kbps.

Each node, except the base station, was powered with a 3.75 V, 2.2 Ah Lithium Ion battery. The base station was powered by two 6 V, 7 Ah Lead Acid batteries.

The electronics and battery were placed in small plastic box with a clear window cut in the lid to enable light to reach the light sensor, see Figure 3.1.

3.1.2 Methods

The five nodes were placed around a large plateau in the fjord leading to Lille glacier in Western Greenland[1]. The initial plan was to distribute the node over an area of approximately 100 – 200 m^2. Unfortunately due to the difficulties encountered while trying to ensure each node could communicate with the base station, only an area of 15 – 20 m^2 was covered. The nodes were deployed for 4 days from the 27/6/2012 to the 1/7/2012.

Figure 3.1: A node in the Greenland deployment. The airflow holes and clear window for the light sensor can be seen.

The nodes themselves behaved very simply; taking measurements once a minute and attempting to transmit them to the base station. Other than the default three Zigbee retries, there were no attempts to queue or retransmit failed messages. The nodes did not utilise a low power sleep mode to conserve energy as an estimate of node lifetime without such energy saving methods was desired. As the base station received data, it time stamped it using an attached RTC (Real Time Clock). Each node used an on-board timer to schedule measurements. High timer accuracy and time stamping data at the base station lead to a maximum time stamp error of approximately 1 second.

---

[1] Latitude 70.4652429 Longitude -50.6834671

3.1.3 Results and Lessons

There were several important lessons learnt during and as a result of the Greenland deployment. The first to be encountered was the poor signal quality. The datasheet for the Series 2B XBee modules states ranges of up to 1600 m, or 1 mile are possible. With nodes located low down to the rock surface, but antennas free of obstruction, the achievable range was only 10 – 20 m. Placing nodes higher up resulted in a marginally better signal but there was no signal penetration through rocky obstructions. It is possible that the rock contained Telluric iron, a very rare form of iron. One of the few, and by far the biggest deposit is in Disko Bay in Greenland, close to where the network was deployed. Regardless of the cause, it became obvious that a node's mounting, especially the position and height is crucial in achieving a good range. It is also clear that sometimes the environment has a large impact deciding where nodes can be placed.

The second problem to be encountered was the power usage. The sensor nodes themselves started with full batteries, 4.2 V for a 3.7 V Lithium battery. After 4 days of operation, all of the batteries were below 3.7 V. This is clearly not acceptable for a system aiming to sustain long term operation. The base station unit fared better, with the battery voltage only dropping by 0.1 V, due to the considerable amount of energy stored in the two 6 V 7 Ah lead acid batteries. The reason for choosing these larger batteries for the base station was due to the fact that it would spend more time receiving data and had to also power an SD card to store data.

The Zigbee protocol includes three types of node; coordinators, routers and end devices. Networks can only contain one coordinator but any number of routers and end devices. The power consumption of the nodes, which were configured as end devices, suggested that they would likely need to sleep to conserve power. However, Zigbee meshing is not designed to accommodate routers or coordinators that sleep. Any node that wishes to route packets, that is a router or coordinator node, must stay awake so that it does not miss the packets. In a larger scale deployment most, if not all nodes, would need to have the capability to route data. Given these requirements and the high power consumption of the XBee transceiver modules, it was clear that alternatives should be investigated. It was felt that the XBee modules themselves were still a suitable hardware platform, but that the Zigbee protocol was better suited to deployments where power was not such a concern, such as home automation.

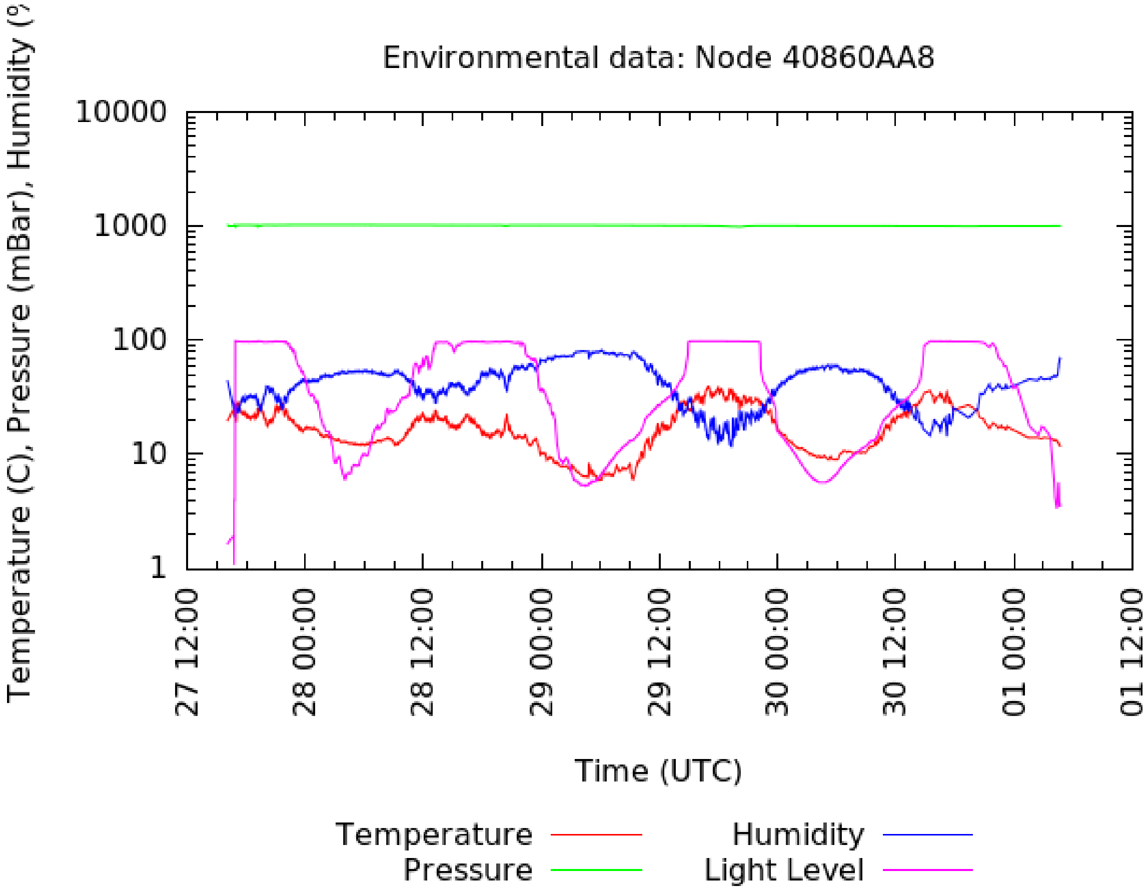

Figure 3.2: Environmental data, plotted on a log scale, from node 40860AA8 for the 4 day duration of the Greenland deployment.

These initial problems were contrasted with the quality of the data recorded. Figure 3.2 illustrates diurnal patterns in temperature, humidity and light level. The pressure remained fairly constant over the four day deployment period. An inverse correlation between temperature and relative humidity can be seen. This as due to warmer air having a higher moisture capacity. If the amount of moisture in the air is fixed, then as the temperature increases the relative humidity drops as the air is capable of holding more moisture. Light sensor data also showed that there was 24 hr daylight. The sensor was almost fully saturated during the day but the light value rose and dropped sharply at the start and end of the day. This was partly due to the geography of the area, high mountains that blocked out the sun when it was low in the sky, and the sensor case, which shielded the light sensor from the sides. Temperature fluctuated from around 6 – 39 °C. The air

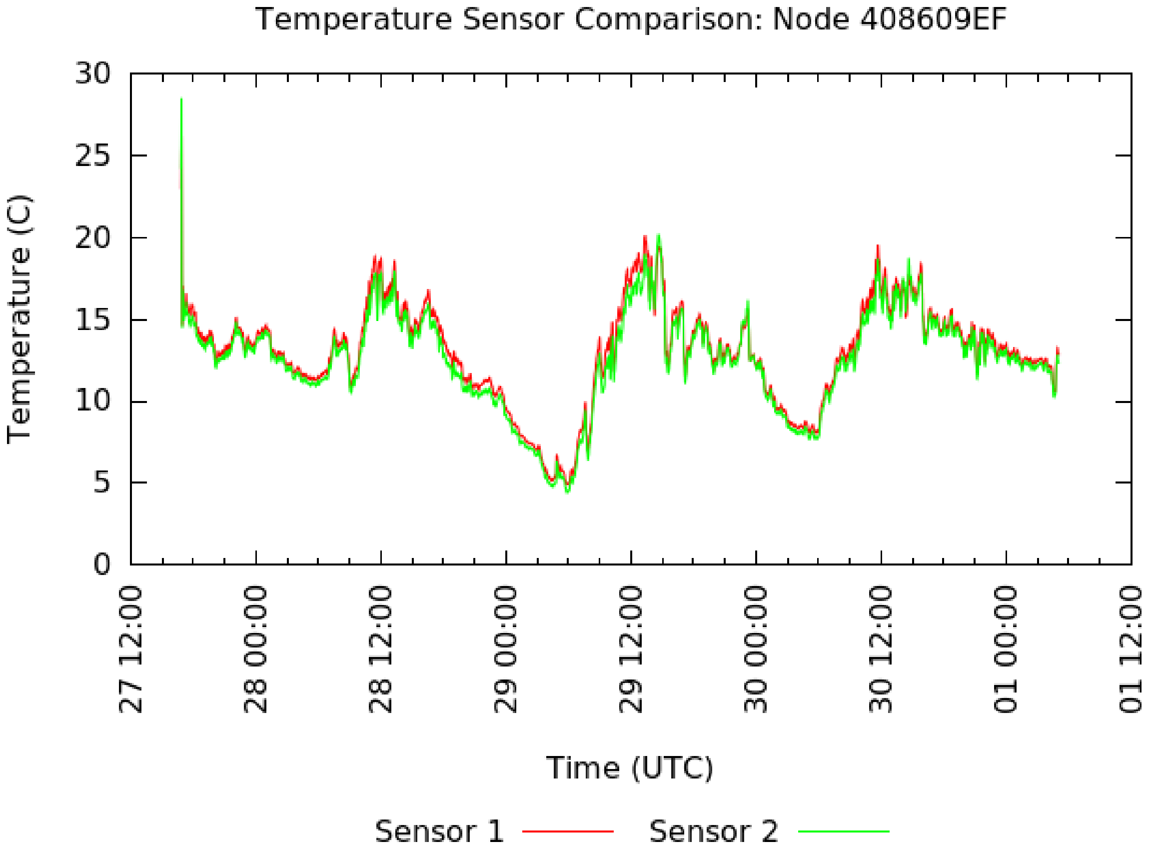

Figure 3.3: Temperature as recorded by both temperature sensor on node 408609EF over the 4 day Greenland deployment.

temperature was relatively low and the close proximity to ice and the sea acted as a cooling influence. Being placed near the ground, in black boxes with poor ventilation likely caused the nodes to heat up in the strong 24 hr sunlight. Figure 3.3 shows the output from the two temperature sensors on the USB Weather Boards. There was an almost constant small offset of around 0.5 – 1 °C between the values returned by the sensors over the course of the experiment. This was likely due to a small calibration error. Neither sensor appeared to be noticeably better or worse than the other.

The following is a summary of the lessons learnt from the Greenland deployment:

1. Don't rely on claimed communications range, it can be impacted by a number of factors including the height, position, mounting/enclosure and environment. The environment can dictate where equipment can be located.

2. The sensor enclosure design is important as it affects the data. Potential problems include obscuring sensors and heating due to exposure to sunlight.
3. The data collected by the low cost meteorology sensors chosen was representative of the environmental conditions. Where two sensors measured the same parameter they produced the same output subject to a small constant offset.
4. The Zigbee protocol is not designed to have sleeping router nodes, however the XBee modules used support alternative protocols.
5. The power consumption of the sensor and microcontroller board chosen was too high is left continually powered.

To create a usable sensor node platform the power consumption, enclosure and choice of protocol needed to be addressed but the sensing equipment was considered acceptable for low cost climate monitoring.

3.2 Ystumtuen Digimesh Deployment

The next deployment used five sensor nodes and a base station node which were deployed near the village of Ystumtuen in Mid Wales for a period of just over two weeks. This enabled the testing of deployment techniques, node hardware and Digimesh firmware. A number of changes were made in response to the lessons learnt from the Greenland deployment:

1. New enclosure and mounting designs were used to better secure the sensor nodes. Modifications were made to reduce the impact of the enclosure on the sensor data.
2. The Zigbee protocol was replaced with the Digimesh running on a variant of the XBee modules previously used.
3. The 3.7 V 2.2 Ah Lithium Polymer batteries were replaced with a 6 V 7 Ah Lead Acid battery and 2.5 W solar panel to increase the available power.

3.2.1 Hardware Changes

The original microcontroller board was kept and the software used was very similar. The following changes were made; a higher capacity battery, the addition of a solar panel, a different XBee module and a new sensor case.

The first issue to be addressed from that of the previous deployment was that of power and, as a by product, node construction. The 3.7 V 2.2 Ah Lithium Polymer batteries used in Greenland were replaced with bigger 6 V 7 Ah lead acid batteries. This type of battery chemistry is more robust to being over charged or under-charged and retains more of its capacity in cold weather. In conjunction with the new battery a 2.5 W solar panel was added to the design. It was believed that, even with the suboptimal power consumption of the hardware used, these modifications would be enable the nodes to remain powered for several weeks. As a result, the construction of the node was changed, with the electronics mounted in a box suspended inside another white box. Ventilation holes in both boxes were cut to provide more accurate temperature, humidity and pressure measurements by allowing the free passage of air. The case was also painted white to reduce solar heating of the node by allowing air to flow through the case and by reflecting sunlight. The new case necessitated the use of an LED ‘light pipe’ to channel light to the light sensor. Finally, new 5 dBi antennae were mounted on the top of the box to increase range. The new enclosure is shown in detail in Figure 4.3 in the next chapter.

The final change was to move from a Series 2 50 mW XBee module to a Series 1 60 mW XBee module. The Series 1 modules are capable of using the Digimesh meshing firmware provided by Digi[2]. Digimesh and Zigbee differ in a few key aspects, the most important ones in the context of this work are that router nodes can sleep and time synchronised sleep modes are provided. While not tested in this deployment the synchronised sleeping and ability for routing nodes to sleep was considered to be worth pursuing if the increased battery capacity and solar panel were insufficient.

3.2.2 Methods

The network was organised in a star topology around the base station. The use of Digimesh meant that nodes were able to route data through other nodes to the base station, however, as the nodes were sufficiently close to the base station, this was unlikely. Five nodes were placed around the base station at varying distances. One node, 409a92fb, was very close to the base station, only 5 meters away. Node 409a9307 was placed on the top of a nearby hill with a clear line of sight to the base station. Node 409a9301 was placed at the bottom of the same hill with a relatively clear line of sight to the base station. 409aa1d7 placed

---

[2] The manufacturers of the XBee transceivers.

around 200 m from the base station on another nearby hill and 409aa1d2 was placed on the other side of that same hill. The placement of node 409aa1d2 was an attempt to make communicating with the base station directly very difficult and to force it to route data through node 409aa1d7. Figure 3.4 shows the location of the five nodes in relation to the base station.

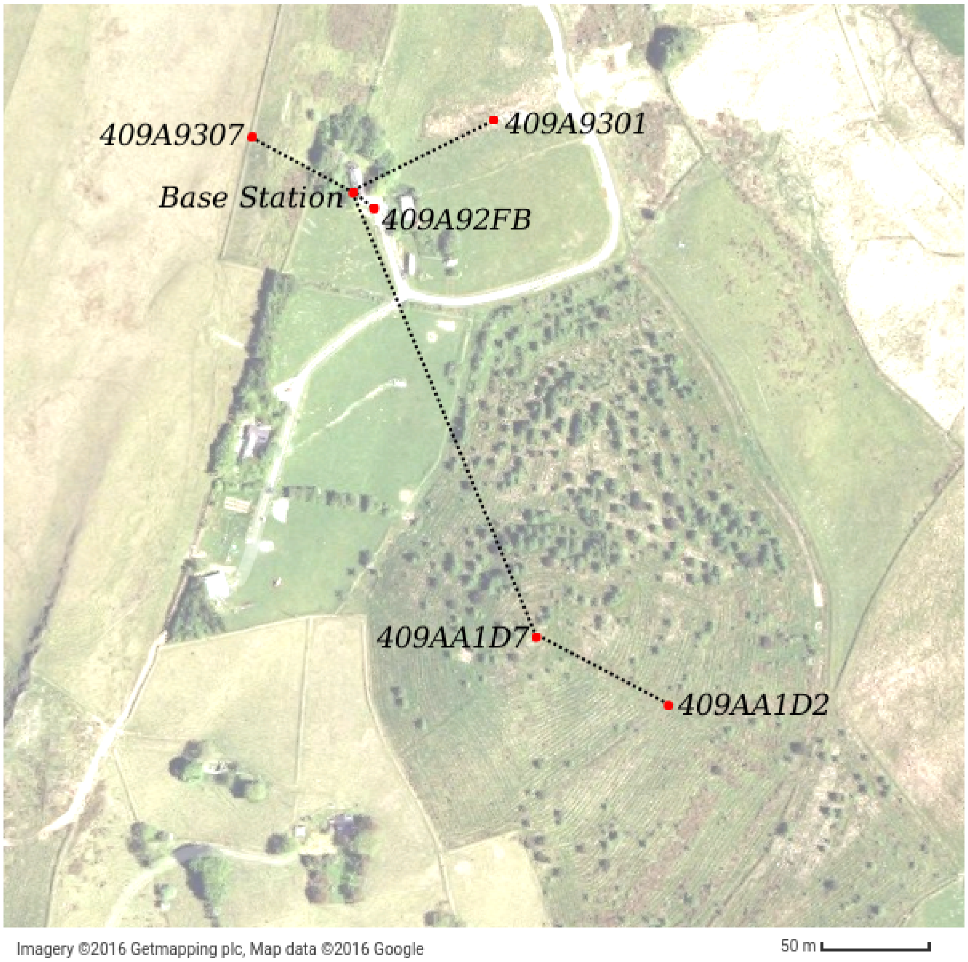

Figure 3.4: Map showing the node locations, base station location and desired links between nodes for the second preliminary deployment. As Digimesh is able to perform route discovery and use alternate routes there was no guarantee that the routes above were exclusively used.

Each node was set to transmit data at a rate of once every 60 seconds and were left until the batteries were deemed completely flat; less than 5 V. The default Digimesh settings, for example number of MAC ACK retries were utilised and no sleep modes were used.

Table 3.1: Number of packets received by the base station from each node over the duration of the whole experiment and the first three days only.

Node	Number of Packets
409a92fb	21259
409a9301	17360
409a9307	19987
409aa1d2	7539
409aa1d7	11797
First 3 Days only	
409a92fb	4239
409a9301	4082
409a9307	4037
409aa1d2	4074
409aa1d7	4222

3.2.3 Results and Lessons

The experiment was started on the 7th of June 2013 and stopped on the 24th of June 2013, which should have resulted in approximately 23985[3] packets from each node.

Table 3.1 shows the number of packets received by the base station from each node. The highest number of packets were received from node 409a92fb, which given its location (within 5 m of the base station), was not unexpected. Nodes 409a9307 and 409a9301 both had good packet counts, 80% to 95% of the number of packets returned by node 409a92fb. Both of these nodes were within 30 m of the base station with a relatively clear line of site. Nodes 409aa1d2 and 409aa1d7 had much worse packet counts and were situated further from the base station than the other nodes.

Figure 3.5 shows the battery voltage of each node over the duration of the deployment. By the end of the experiment most of the batteries were either almost depleted or already depleted. One node had completely stopped transmitting data and three others only

---

[3] The number of minutes between 7/6/13 19:52:00 and 24/6/13 11:37:00

powered up during the day due to the power from the solar panel. All five nodes were operational in the first three days with battery voltages above 6 V. To estimate how well the Digimesh firmware and XBee Series 1 modules performed, the number of packets received during the first three days was calculated and the results are shown in Table 3.1. In three days each node should have sent approximately $60 \times 24 \times 3 = 4320$ packets. All of the nodes were within 6.5 % of this value with 409a92fb and 409aa1d7 coming closest to this theoretical value. The differences were most likely attributable to a small frequency error in ceramic resonator used as the microcontroller clock. The fact that the number of packets from each node was close to the expected number and that there was not much variation between nodes suggested that the Series 1 XBee modules and Digimesh firmware performed well.

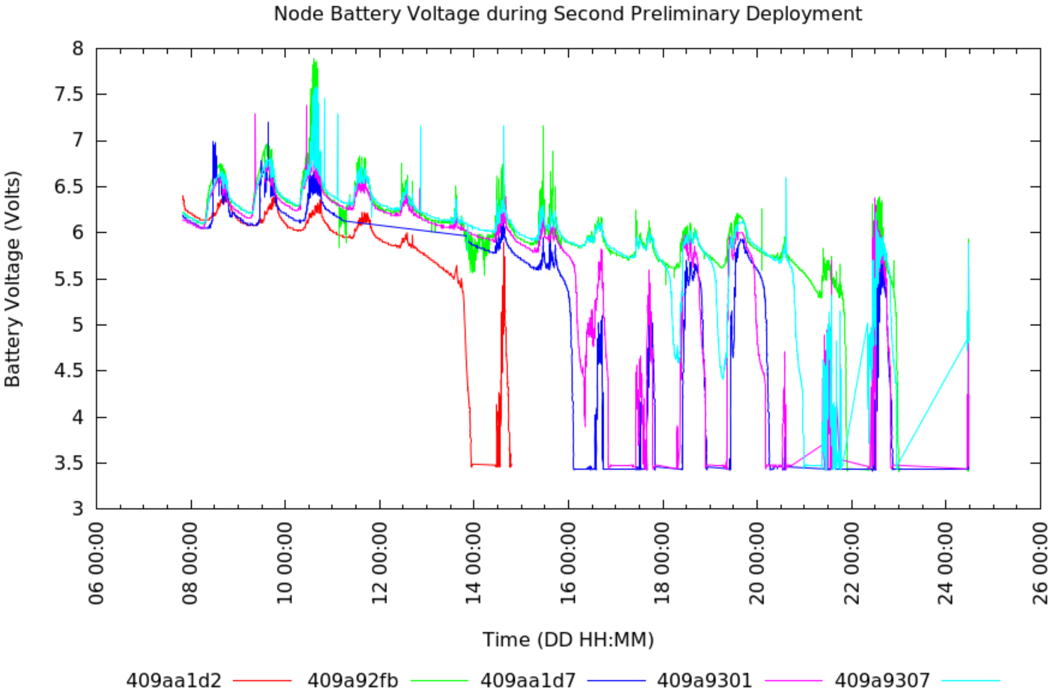

Figure 3.5: Each node's battery voltage over the course of the deployment.

Signal quality between nodes was better than expected which resulted in an almost fully connected network and 409aa1d2 being able to communicate directly with the base station. There may have been some degradation in signal quality due to interference, meteorological factors or other changes in the environment. However, lost packets are believed to be due to lack of power, brownouts[4] or Digimesh failing to route packets successfully.

3.2.4 Time Stamping

Another issue noted, was that of time stamping data as it arrived at the base station node rather than on the node at the time of sampling and the potential for inaccuracy. If a packet were to be retransmitted multiple times due to some form of transmission failure, the time stamp would be wrong. This could be problematic and therefore the worst case scenario was calculated in order to determine if this was acceptable. The worst case when attempting to transmit is when an existing route fails and forces the route discovery process to run. The XBee data sheet[35] states that the maximum time for data to successfully be transmitted to a base station can be calculated, using the values in Table 3.2, as follows:

$$\mathrm{B_{time}} = NN \times NH \times (MT + 1) \times 18\mathrm{ms} \tag{3.1}$$

$$\mathrm{UR_{time}} = 2 \times NH \times MR \times \mathrm{UT} \tag{3.2}$$

$$\begin{aligned} \mathrm{M_{time}} &= \mathrm{B_{time}} + (NH \times \mathrm{UT}) \\ &+ (2 \times \mathrm{UR_{time}}) \end{aligned} \tag{3.3}$$

$$\begin{aligned} \mathrm{B_{time}} &= 3 \times 7 \times (3 + 1) \times 18ms = 1512 \\ \mathrm{UR_{time}} &= 2 \times 7 \times 1 \times 5 = 70 \\ \mathrm{M_{time}} &= 1512 + (7 \times 5) + (2 \times 70) = 1687\mathrm{ms} \end{aligned}$$

where B_{time} is the broadcast transmission time, UR_{time} is the known unicast route time and M_{time} is the maximum amount of time. Using the default values in Table 3.2, the largest delay possible between transmission of data and receipt of data was 1687 ms. This is the very worst case and would require retransmission over a 7 hop mesh link. We consider this to be an acceptable error, given the type of data that was being measured. The fastest changing value was the light level which, due to cloud cover, could change in less than a second. None of the other parameters change quickly; temperature, humidity and air pressure all take minutes to change. If this worst case time delay were to be considered problematic in the future, the solution would be to equip each node with a Real Time Clock module. In addition to an RTC, some form of time-syncing protocol would

---

[4] A drop in voltage below the minimum necessary for operation.

Table 3.2: The default settings for the Digimesh firmware.

Parameter	Function	Value
MT	Broadcast retries	3
NN	Network Delay Slots	3
NH	Network Hops	7
MR	Mesh Retries	1
RR	MAC Retries	0
UT		5ms

need to be implemented to enable each node to maintain the accuracy of its RTC module with respect to a global time source.

3.2.5 Packet Format

The data transmitted by each node was in the form of ASCII characters. The amount of data transmitted could, therefore, be reduced by using the bytes of the sensor values in some known format and transmitting those instead. This would, for example, reduce the pressure measurement '1203.21' which is 7 characters, or 7 bytes, in size to the 4 bytes used to represent a double in AVR C. By employing some form of compression, this could be further reduced. Transmission is very expensive in terms of energy, however this may be overshadowed on the platform used by the high constant power consumption of a node remaining awake and in receive mode. The amount of work involved in implementing compression on this platform makes it undesirable for a saving of around 10 bytes per transmission. If transmission energy cost were to become a constraining factor this could be revisited.

3.2.6 Conclusions

Figure 3.5 shows each node's battery voltage plotted over the duration of the experiment. There was a clear diurnal pattern where the solar panel was charging the battery and therefore pulling the voltage higher. This would suggest that the solar panel was working and that it was able to charge the battery. Despite this, it only took two weeks to completely flatten every node's battery, with some nodes lasting less than a week. The experiment was conducted in mid summer in good, clear weather providing optimal conditions for the solar panels. The amount of available light led to expectations of power stability and therefore nodes were powered on at all times with no attempt at power saving. It was clear, as a result of this deployment, that significant power savings would still need to be made. Putting the microcontroller and/or XBee into some form of low power sleep mode or modulating the transmission rate were obvious candidates for investigation.

Digimesh, with the default settings, performed remarkably well given the circumstances. By the end of the experiment many nodes were only operating during the day and running out of power at night resulting in some nodes continuously losing power and rebooting during the transitions between night and day. However, since all of the nodes could communicate with the base station directly it was unclear if the routing capabilities of Digimesh were well tested. However, the use of Digimesh meant that the route that data took to get to the base station was unpredictable which potentially made keeping the experimental conditions as similar as possible between experiments harder.

The new cases worked well keeping the electronics sheltered from direct sunlight. The light data collected was similar to that collected in the Greenland deployment which indicated that the LED light pipes were successful in channelling light to the light sensor. The location for deployment provided a diverse range of environments to monitor and was suitably exposed and remote to consider deployments there 'real world' deployments. The biggest problem was still power consumption which, even with a larger battery and solar panel, was too high.

3.3 Ystumtuen Modified Digimesh Deployments

Previous deployments showed that power consumption was the limiting factor in preventing a suitably long network lifetime. Two modifications of the Digimesh based architecture were tested in an attempt to achieve a relatively lower power node architecture. The first modification was to modulate the frequency of data transmissions by the available light level. When this modification failed to extend the network lifetime, a second approach was tested. The second modification was to duty cycle the radio module to conserve power and use a time synchronisation method to ensure that all nodes were awake at the same time for the Digimesh routing protocol.

Nodes were deployed in almost the same locations as in the previous experiment. The main difference is that the node furthest from the base station, 1D2 was moved to within 10 m of the base station. This was an attempt at preventing Digimesh from trying to route messages. The experiment was run until the node batteries were considered flat. The experiment was not successful with Digimesh seemingly unable to operate in the duty cycled manner used.

3.3.1 Light Varied Transmission

In an attempt to lower power consumption the transmission rate was modulated by the available light level. The baseline transmission rate was lowered in order to be able to exploit periods of high energy availability by increasing the transmission rate. If the baseline power consumption was too high then this technique would be unlikely to significantly improve the overall power consumption of the nodes. If this was the case, then steps to lower the baseline power consumption before continuing would need to be taken.

The nodes were deployed between the 26th of June 2013 and the 3rd of July 2013, a period of around 6.75 days. The node hardware used was the same as the previous experiment; USB Weather Board, custom case, 6 V 7 Ah lead acid battery and a 2.5 W solar panel.

The rate of transmission, and therefore sensing, ranged from every 10 seconds to every 10 minutes. Values from the light level sensor, 0 to 100, were mapped linearly to a sensing rate; a high light level to a high sensing rate and a low light level to a low sensing rate. Three nodes, 409a92fb, 409a9301 and 409a9307, were operational for the duration of the experiment and delivered, for the most part, a similar number of packets. Whilst some of the variation in packet counts was due to the occasional dropped packet, most variation appears to be as a result of differing light levels due to different locations. Nodes 409a91d7 and 409a91d2 were in locations where the sky was partially obscured due to tree or plant coverage. As a result, these nodes received less light and therefore less power. The effect of this can be seen in Figure 3.6, in early hours of the 3rd of June, when their battery voltages drop significantly to around 3.5 V. Remarkably these nodes continued to function at this very low voltage for several hours. By the end of the same day, the remaining nodes all had a battery voltage of less than 6 V and were exhibiting a downwards trend.

The results of this experiment suggested that further power savings would need to be

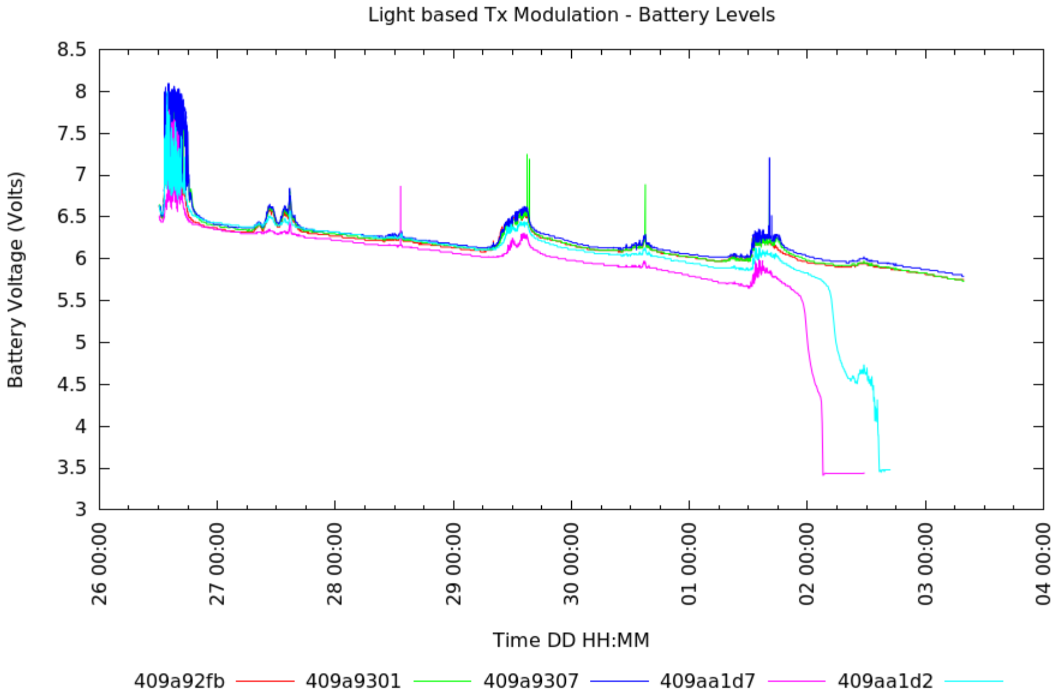

Figure 3.6: Battery data from the light varied sensing rate deployment. Nodes 409aa1d2 an 409aa1d7 ran out of power after 6 days, with the other three nodes exhibiting a downwards trend.

made. The experiment was started just after the longest day of the year, the 21st June 2013, in good weather and therefore experienced an almost maximal solar energy input. Despite this, after only a week, every single node had either depleted or almost depleted its battery. The power consumption of the sensor node was 0.43 W and the solar panel should be able to provide 2.5 W. The average output of the panel was clearly significantly less than 0.43 W and therefore the power consumption needed to be reduced by a large proportion of this.

3.3.2 Time Synchronisation and Duty Cycle

The most effective methods of reducing power consumption are to reduce the power consumed by modifying the electrical (or mechanical/software) design of the system or to reduce the time the system spends consuming power. With a microcontroller, this can be achieved by entering low power “sleep” or “power-down” modes. Most modern microcontrollers support this functionality, whereby various features of the microcontroller are shut down to reduce power consumption. The Atmel 328p microcontroller used in the sensor nodes has multiple low power modes with each resulting in a different level of power saving. Only a little power can be saved by turning off the Analogue to Digital circuitry. A lot can be saved by stopping all of the oscillators and effectively "turning off" until awakened by some predefined source or trigger. These low power modes have a penalty in reduced, or total loss, of functionality. This is a technique used by numerous other sensor node platforms as discussed in Section 2.5.1.

Whilst power savings could be made by utilising low power modes in the microcontroller, the XBee transceiver consumed considerably more power than the microcontroller. The XBee transceiver consumed between five and six times as much power as the microcontroller, around 0.33 – 0.36 W compared to the microcontrollers 0.06 W. Reducing the power required by the XBee module therefore has a more significant impact on the overall power consumption. Like the microcontroller, the XBee module also has a sleep mode. In this mode power consumption is reduced to, theoretically, only a few micro amps, but the XBee cannot perform any tasks. There is also a small period of time after waking the XBee in which it cannot send or receive data.

The sleep mode obviously cannot be used constantly as the XBee would be unable to transmit or receive data. To manage this, a sleep duty cycle is often used. Over a defined period of time, some percentage of that time is spent 'awake' and the rest spent 'sleeping' to save power. The duty cycle used for this deployment was 20%; the XBee spending 80% of its time in a sleep mode to conserve power. The calculations for determining this duty cycle are as follows. The solar panel used is a 2.5 W panel at 6 V open circuit. This should result in a possible charging current of 416 mAh. The actual current was measured, on a sunny day without clouds, to be 270 mA at 6 V. Due to the lack of sunlight at night, environmental factors such as obstructions and accumulation of dirt a conservative estimate of 10 % was used to estimate the average power output of the solar panel. As a result it was assumed that the solar panel would be able to supply, on average, 27 mA (at 6 V) to the battery. The sensor node system used 12 mA whilst the transceiver was sleeping and 72 mA while it was awake. The current draw of a node in the 20% duty approach would be 72 mA for 20% of the time and 12 mA for 80% of the time, for a total of 24 mA. Considering there will be some inefficiencies in charging the battery this is very close to the theoretical 27 mA produced by the solar panel. As a result it was expected that the system would not run out of power.

In order for the Digimesh firmware used on the XBees to be able to route data, all nodes needed to be 'awake' at the same time. The reason for this is due to the fact that in order to route messages around the network, the receiving node must be awake at the same time. If node A needed to transmit data to node B, then node B would need to be "awake". Theoretically if each node had a perfect clock, as long as all nodes were turned on at the same time, they would remain synchronised. However the oscillator in every node experiences substantial drift, sometimes as much as 200 to 300 ms per minute. As a result some form of synchronisation was required.

The Digimesh firmware for the XBee module provides facilities for synchronised sleeping of nodes. In this mode, the XBee module itself handles synchronisation and can inform a microcontroller that it should now be sleeping. The sleep period, wake length and sleep length can all be changed on an individual and the changes automatically propagate throughout the network. Unfortunately the system is very much a black box and, more importantly, does not allow individual nodes to have differing sleep settings whilst remaining synchronised. This would mean that individual nodes' power consumption could not be modified while remaining synchronised with the rest of the network.

As a result the Digimesh time synchronisation was not used and a simple pairwise synchronisation scheme was built on top of the Digimesh firmware at the application layer. It used the Arduino *millis()* function to provide the current time. The time synchronisation worked as follows:

- Node A needs to synchronise its time to that of Node B.
- Node A stores its current time ($t0$) and transmits a packet to Node B.
- Node B receives this packet, stores its current time ($t1$) in the packet and sends it back to Node A.
- Node A receives the response and stores the current time ($t2$) again. It then calculates the offset using the three time stamps in the following manner: $\frac{t2+t0}{2} - t1$ and applies it to its clock.

The method worked by assuming that the time taken to transmit the packet one way was the same as it was to transmit a packet back. While this was not always the case, for the level of synchronisation required it was considered a fair assumption. An example of the synchronisation calculation was as follows. Node A sent a packet to Node B at time 100 ($t0$), Node B received it at time 117 ($t1$). Node B then transmitted a packet containing 117 to Node A which received it at time 120 ($t2$). We assumed that the time taken to send a packet from Node A to Node B was the same as the time taken to send a packet from Node B to Node A. All the times were local to each node as each node had its own clock source to keep track of time. The midpoint of the times $t0$ and $t2$ was $\frac{t2+t0}{2} = \frac{100+120}{2} = 110$ and if the nodes were synchronised this should have been equal to $t1$. If they were not synchronised then the offset between the two nodes could be calculated as follows: $\frac{t2+t0}{2} - t1 = 110 - 117 = -7$. This value was used by Node A to correct the its time. As long as this process was performed regularly then the clocks remained synchronised.

3.3.2.1 Methodology

Only the XBee's sleep mode was used, leaving the possibility of enabling the microcontroller's sleep modes at a later date. Each node would wake the XBee module only when necessary for transmitting or receiving data. The length of time spent sleeping or awake could later be modified to further improve or relax power saving efforts. Synchronisation was performed once per minute between each node and the base station. The XBee transceiver module was put into a 20% on, 80% sleep duty cycle. To achieve the 20% time on, the XBees were turned on for 2 seconds in every 20 seconds and a further 6 seconds once per minute for time synchronisation. This totalled 12 seconds spent on in every 60 seconds. Each 2 second period was considered to provide enough time for nodes to transmit or route data back to the base station. The nodes were placed in the same locations as those shown in Figure 3.4.

3.3.2.2 Results

An initial deployment with multi-hop paths to the base station was attempted but was unsuccessful. While the time synchronisation worked fairly well, synchronising nodes to within ±20 ms it did occasionally fail. In particular nodes that were more than 1 hop from the base station did not synchronise well. This was usually due to a node needing to retransmit a time synchronisation packet, the CSMA/CA backoffs due to the wireless channel not being free or Digimesh routing failures. This resulted in the time offset that was calculated being incorrect or unable to be performed at all. While this can, and has, been solved in many existing time synchronisation solutions the functionality exposed by the Digimesh API makes doing so very hard. In addition to this, the Digimesh protocol did not function reliably when tasked to transmit data in the 2 second ‘on’ periods of the duty cycle.

3.4 Summary of Preliminary Deployments

The source of the Digimesh problems was not precisely determined and it was decided that using a simpler firmware, which provided fewer ‘high level’ features but offered more control, was the right decision. There were a number of key factors considered when making this decision.

- Many of the operations performed by the Digimesh protocol were difficult to monitor and thus difficult to control.
- The protocol is designed to accommodate sleeping nodes and even has a synchronised sleep mode. However in synchronised sleep the sleep cycle of individual nodes was not controllable. Efforts to overcome this using time synchronisation to schedule network-wide wake times were unsuccessful.
- The multihop routing system employed by Digimesh prevented access to packets by the microcontroller at intermediate hops in a route.

While Digimesh offers a number of useful features, they were not able to be leveraged in a useful manner for this work.

The efficacy of the duty cycle method of reducing power consumption was not directly tested due to the failure of the time synchronisation and Digimesh deployment. However, by examining the current consumption of a node it was determined that the power savings were considerable and would most likely be sufficient for longer term operation.

- The current draw of a node without the duty cycle approach was 72 mA or roughly 0.43 W at 6 V.
- The current draw of a node in the 20% duty approach was 72 mA for 20% of the time and 12 mA for 80% of the time. This totals 24 mA or 0.144 W at 6 V.
- The lead acid battery was rated at 6 V and 7 Ah which was 42 Wh of energy.

The number of days a node using the duty cycle approach should, theoretically, be able to survive for before running out of power was calculated as follows. The number of hours a node would last was equal to the energetic capacity of the battery (42 Wh) divided by the energy consumed by the node in an hour (0.1416 W). This resulted in approximately 292 hours, just over 12 days, of operation with no solar input. Comparatively, the non duty cycle approach would have yielded around 4 days using the same battery. The calculations in Section 3.3.2 showed that the solar panel would be sufficient for indefinite operation.

The preliminary deployments were very useful in determining which approaches to use with the hardware available. They yielded the design for and an opportunity to test the sensor node case and mounting system. The meteorological sensors chosen were also tested and produced data that was relatively accurate and reliable, see Section 4.5 for more detail. The preliminary deployments enabled the testing of the wireless transceiver, in particular the achievable range, which later informed the placement of nodes for experiments. The process provided key insights into the problem which were used when designing the final hardware, software and network layout used for experimentation.

Chapter 4

Methodology

This chapter presents a discussion on how the human endocrine system was used as inspiration for the control systems in this work. The notional hormones used during experiments are presented and the mechanisms through which they were produced, used, decayed and combined are detailed. The mechanical, electrical and software designs of the wireless sensor nodes used throughout the course of the research are discussed. Design decisions are explained, justified and the implications of these decisions on the research aims are detailed. The transmission and routing systems created to allow nodes to communicate are presented along with the limitations imposed by these systems. Deployment methodology is explained including the network topology and deployment procedures. Lastly an example post calibration of the meteorology data obtained is examined to determine its effectiveness in this case.

The work carried out in Chapter 3 helped guide the final electrical and mechanical design of the node described in Section 4.1 and the design of the duty cycling system described in 4.2. The scenario of a wireless sensor network for environment monitoring, discussed in the introductory chapter, is used to guide the design of the sensor nodes, hormone systems and experiments. The number of nodes to be deployed is small, just 20, similar to many real world deployments discussed in Section 2.4.1.1. Each node is important as with the low number of nodes, each node failure results in an appreciable loss in data and negatively impacts the network's ability to route information. The power consumption is also important as many real world deployments are in remote locations making replacing batteries or nodes difficult or impossible and desired deployment times are either very long or indefinite.

4.1 Node Design

This section details the choice of microcontroller, sensors and transceiver used in the sensor nodes. Justifications of each choice are provided along with a discussion on the alternatives and ramifications of the hardware chosen. The mechanical design of the nodes and node mounting system are also presented. Much of the work carried out in Chapter 3 informed the decisions made in this chapter.

4.1.1 Arduino Platform, Microcontroller and Sensors

An ATmel ATmega microcontroller platform was chosen as the basis of the sensor node. A large factor in this decision was the compatibility with the Arduino platform. Arduino is a combination of IDE, toolchain, libraries and compatible microcontrollers. Originally, only a small number of 8-bit ATmega microcontrollers were supported but compatibility has been greatly expanded in the last few years and now includes 32-bit ARM microcontrollers which provide significantly more computational power. The Arduino platform has gained a large following which has a number of benefits:

- Support - There is a very active community containing substantial expertise. Troubleshooting and bug-fixing is greatly expedited as a result. Resources and documentation are more readily available as well as a large number of tutorials and guides.
- Hardware - A huge number of devices are designed for, or have been made compatible with, Arduino devices. This results in a great variety of technologies being readily available, with good support and expertise to go with them.
- Software - Libraries have been written for almost all commonly used Arduino sensors and devices. Many devices having a selection to choose from.

All of these factors increased the chance of rapid development and testing phases of the sensor node, as there was little existing expertise in the area of wireless sensor networks in the department.

Another initial design choice was to use the USB Weather Board from Sparkfun Electronics, referred to from now on as the ‘Weather Board’, as the platform for the sensor nodes. This board, shown in Figure 4.1, provides a microprocessor platform complete with sensors and the requisite power and programming circuitry with pin breakouts for transceiver hardware and external sensors. Again, this removed the need to design and create a platform which, as it was not the focus of the research, saved significant time.

There are a large number of commercially available wireless sensor network platforms for example the Berkely Mica family (MICA, MICA2, MICAz, MICA2Dot), IRIS, IMote, the Telos family and many more and are discussed in more detail in Section 2.2. These devices are designed specifically for wireless sensor network research. This means they are very low power, well tested and have been used for research purposes already. They have a number of features such as precision timers, real-time clocks, on board storage that are useful in wireless sensor network development. Many routing protocols, time synchronisation protocols and other mesh networking techniques are available and have been well tested and utilised in real world deployments. However, these devices were hard to obtain and expensive. Much of the existing work using these devices was conducted in isolation not as part of a whole sensing system. It was felt that using the Arduino environment and Weather Board would produce a functioning software and hardware platform in less time, whilst being more flexible, easier to work with and more robust to real world deployment conditions.

The Weather Board, with a 2.4 GHz transceiver module and all of the sensors, the battery, solar panel and enclosure came to around \$150 (£93) which was comparable or cheaper to the alternatives. The equipment was quick and simple to purchase arriving in a matter of days. Finding a price for many of the existing platforms was difficult and delivery times were unknown.

As most deployments of Wireless Sensor Networks are battery powered, power consumption is very important. Low sleep and wake current consumption are very important. The Weather Board's current consumption whilst awake is within the range found in the alternatives (10 mA – 25 mA). Sleep current consumption, however, is significantly higher. The alternatives to the Weather Board typically have sleep currents of <20 μA, compared to the Weather Board which uses around 5 – 6 mA. A node based on the Weather Board platform will have a much shorter life, due to power consumption. This, however, was not considered to be a bad thing in this instance. The higher power consumption put much more pressure on the techniques used in the wireless sensor network to manage power consumption and adapt to the available power. It also allowed for real-world experiments over relatively short time-scales that could examine the performance when available power was exhausted or very scarce. It was felt that a high power consumption would actually be beneficial for the research being conducted.

Many of the available node platforms were developed specifically for use in wireless sensor network research. The benefits of this are that a lot of research has been conducted on these platforms. This should allow better comparison of results and techniques to existing work, but given the number of available platforms it is likely that to draw accurate real world comparisons, work would need to be reimplemented on one or more platforms.

The software environment used in many of the Wireless Sensor Network platforms is typically some form of RTOS (Real Time Operating System). Commonly used RTOSs are TinyOS, Contiki and FreeRTOS. These provide task scheduling, asynchronous events, timing and time keeping, and a whole framework for development. These are just some of the features that are useful for wireless sensor network development. Having reliable task scheduling and an event system that is well tested removes the need to reimplement them, thus allowing time to be spent on developing experiment logic. There are, of course, downsides of such operating systems. The most obvious downside is that such platforms force things to be done a certain way to fit in with the way the OS designer felt things should be done. For example, the concept of tasks is very useful, however, the allocation of maybe 100-150 bytes of RAM per task can be problematic in a system with only a few KB of total RAM. Program memory can also be an issue, many low end microcontrollers having only 32 KB. Whilst it would be possible to use RTOS with the Weather Board, at the time the research was conducted, they would have required porting to the relevant platform. It was felt that the low barrier to entry and plentiful examples provided by the Arduino environment would result in a shorter time to the first usable prototype.

The flexibility, speed and ease of development were the deciding factors. It was felt that the discussed benefits were not enough to compensate for the time overhead in acquiring and learning how to make use of existing wireless sensor network platforms.

The Weather Board chosen used an Atmel Atmega 328P microcontroller running at 8MHz. The Atmega 328P has 32KB of Flash memory, 2KB of RAM and 1KB of EEPROM. Other useful onboard resources included several timers, ADC, SPI, I2C, UART and power management. The Arduino environment either provides libraries for these features or third party libraries are available. Included on the board were a BMP085 pressure and temperature sensor, SHT15 humidity and temperature sensor, TEMT6000 light level sensor and battery level sensor. There were also headers provided for a radio transceiver compatible with an XBee pin layout and connectors for a wind speed and direction sensor and a rain sensor. The sensors have good accuracy and precision, as detailed below, but were not well calibrated in the deployments carried out.

Table 4.1: Accuracy and precision of the sensors used on the sensor nodes.

Sensor	Accuracy	Resolution
BMP085 Pressure	+/- 1hpa (absolute) +/-0.2 hpa (relative)	0.01 hpa
BMP085 Temperature	+/- 1 C (absolute)	0.1 C
SHT15 Humidity	+/- 2% RH	0.05% RH
SHT15 Temperature	+/- 0.3 C	0.01 C

To aid programming and debugging a USB mini connector with an FTDI USB to logic level serial IC was provided. The boards could be powered from either the USB port or a battery, selectable by a switch. The microcontroller's UART was able to be routed to either a radio transceiver or the FTDI USB controller.

The board itself was 67 mm by 52 mm, making it small enough to be practical for large scale deployment. Although it was sold as a weather monitoring platform, the board was not waterproof or water resistant. This necessitated conformal coating of the boards.

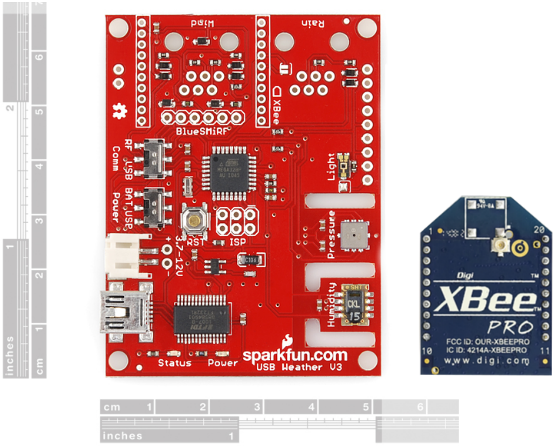

Figure 4.1: The Sparkfun USB Weather Board and Series 1 XBee transceiver module used for the sensor nodes. The Weather Board contains light, temperature, pressure, humidity and battery level sensors.

4.1.2 Radio Transceiver

The Weather Board was designed to take some form of radio transceiver module. There are many available including WiFi, Bluetooth, 433MHz 802.15.4 and XBee modules. The module chosen was a Series 1 60 mW XBee module running Digi's 802.15.4 firmware and is shown in Figure 4.1. The alternatives to the chosen XBee module and the reasons why it was chosen are discussed below.

WiFi is very high power, in the region of 1 W (300 mA at 3.3 V) although the modules are a reasonable price, around £20. It is easy to use WiFi in a simple star network but meshing requires ad hoc modes and it quickly becomes complex. The range is fairly short as the focus is on a higher bandwidth. With the introduction of Bluetooth LE (Low Energy), Bluetooth's power requirements have dropped significantly. Modules are quite cheap and the LE modules have an effective range of around 20 to 50 m. There are a selection of available 433/866/915 MHz modules available. These are predominantly low power, cheap and have a good range. These benefits are counteracted by the lack, at the time of development, of software libraries and support to aid development. Routing and meshing must be developed to work on top of these devices. XBee modules are a compromise in terms of power, cost, range and software support and options. Power consumption, around 200 mW, is higher than most modules other than WiFi. The per device cost is relatively expensive at approximately £25. Ranges of several hundred meters are achievable although claims of one to two miles are made in the documentation. The modules are able to run different, pre-written firmwares allowing several communication stacks to be used. There is firmware available for Zigbee, Digimesh[1] and a simple 802.15.4 MAC layer.

Of the available options XBee emerged as the most flexible and simplest to develop within the time available. The cost was acceptable for the number of nodes required and the range would allow coverage of a relatively large area without the need for an excessive number of nodes. The power consumption of an XBee module is in the region of 200 mW in receive mode, which is high. However it was felt that, with a small solar panel, node lifetime would be multiple months. The pressure that the high power consumption placed on the power saving methods was seen as a positive rather than a problem, as the effects of any power control should be more noticeable.

---

[1] Digi's proprietary meshing technology

4.1.3 Mechanical Design

To protect the electronics and sensors from the elements and facilitate repeated deployment of the sensor network, the design of the node enclosure and mounting system was important. The electronics needed to be protected, but the sensors could not be sealed inside a box as they wouldn't provide accurate data. To achieve this, the following design was arrived at.

The microcontroller board, sensors and XBee transceiver were mounted inside a small plastic box. Holes were cut into the ends of this box to allow air to circulate. This box was then mounted, on stand-offs, in another larger metal box. Holes were cut in the sides of this larger box and a louvred vent mounted over the holes. This provided air flow to the sensors while shielding them from rain. The outer box was painted white to reflect as much sunlight as possible so as to enable a more accurate air temperature measurement. As the sensors were now inside two layers of box, the light sensor required the light to be channelled to it in some way. To do this, an LED 'light pipe' was mounted on the light sensor and passed through both boxes. This allowed light to be channelled from outside the boxes to the light sensor while protecting the sensor from the weather.

The 6 V 7 Ah lead acid battery was located underneath the inner plastic box and was fully contained within the outer metal box. This shielded it from sunlight and moisture. The bottom of the outer metal box had holes cut into it to allow air circulation and allow any moisture that condensed inside the box to drain out. A solar panel and mounting bracket were attached to one side of the outer case. An image of the design and a deployed node can be found in Figures 4.2 and 4.3.

To mount the sensor nodes securely and in the same location between experiments, steel scaffold poles were driven into the ground where nodes were to be placed. These poles did not move between experiments. Each node had an aluminium flange with an inner diameter slightly larger than the scaffold tubes outer diameter attached to the bottom of its case. This allowed the each node to be mounted on top of a scaffold tube quickly and easily. Three gripping screws in the flange could be loosened to allow the node to be rotated to the correct orientation and tightened to clamp a node in that orientation.

4.1.4 Base Station

The base station was comprised of an EEEPC 701 laptop connected via USB to an XBee Series 1 transceiver. The laptop was powered from a mains socket and connected to the

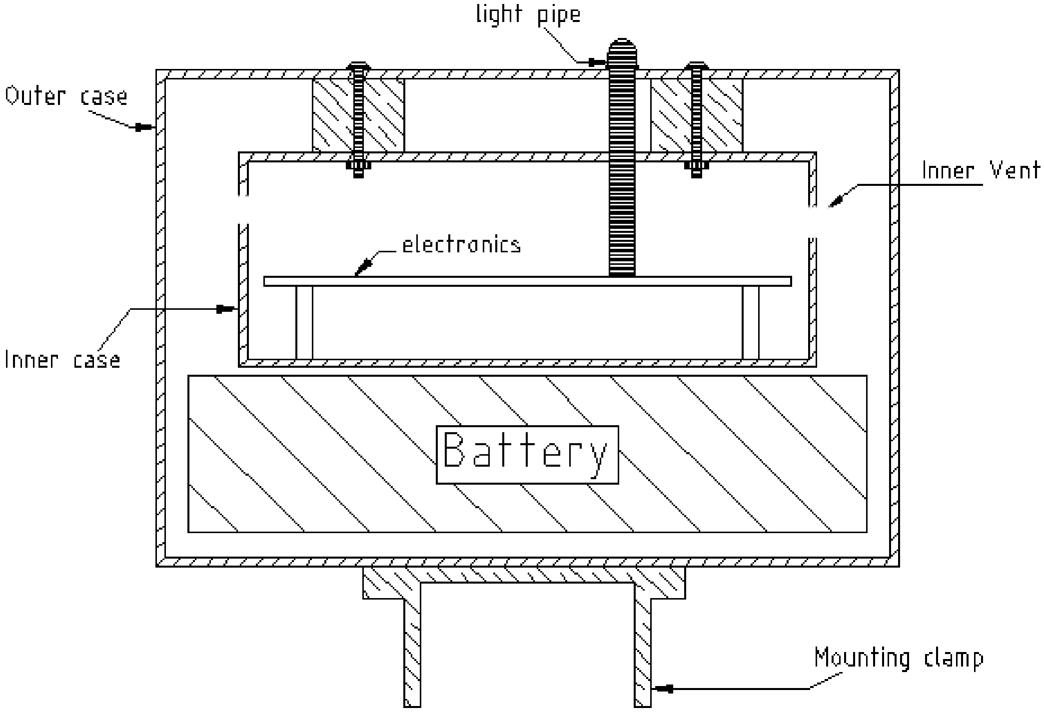

Figure 4.2: Layout of the node enclosures. The position of the battery, internal enclosure, electronics, light pipe and air vents can be seen.

internet for remote access. The base station software was written in Java and provided the ability to log all of the incoming data to disk and transmit certain packets to the network. These packets included:

- Mute/unmute packets to start and stop nodes in deployment mode from broadcasting their neighbour information.
- A 'clear EEPROM' packet to reset nodes.
- A deploy packet that instructed nodes to deploy and start normal operation.
- A neighbour packet so that nodes were able to detect the presence of the base station.
- Centre and wind hormone packets used in later experiments.

The base station also provided the ability to monitor data as it was received so as to enable a user to quickly check on how the network was performing.

Figure 4.3: Image of a deployed node. The solar panel, antenna, outer case and scaffold mount are visible.

For the last experiment a ROWind Wind sensor was attached to the base station so that the wind conditions at the base station could be monitored and used to generate the wind hormone discussed in Chapter 6.

4.2 Control System and Duty Cycling

The control system for the node is based around a duty cycle with a sleep component and wake component. During sleep power is conserved but no actions can be taken, no sensors read and no data transmitted or received. Whilst awake the reverse is true. The percentage of time spent awake vs sleeping determines the average power consumption of the node. Spending more time sleeping decreases power consumption but results in a node spending less time participating in network routing. Spending more time awake increases power consumption and facilitates network routing. A default sensing rate for a set of sensor readings has been chosen to be once per minute. This is similar, or more frequent than many of network deployments discussed in section 2.4.1.1 in Literature Review chapter. The power consumption of the module using this control system is discussed below.

The current consumption of the XBee module in receive mode and the microcontroller combined was approximately 72 mA at 6 V which is approximately 0.43 W. The preliminary work described in Chapter 3 showed that this power consumption was too high to sustain, even with a solar panel, for more than a week or two. To lower the power consumption a duty cycle approach was employed. In this approach, some time was spent operating normally and consuming a large amount of power and the remaining time is spent in low power sleep modes. Both the XBee and microcontroller provide support for sleep modes. As detailed in Section 3.3.2 a default duty cycle of 20% was chosen as it was estimated to allow nodes to operate indefinitely on solar input. The period of the duty cycle was chosen to be 500 ms, as this allowed for relatively low latency communication between nodes. The 20% duty cycle resulted in a node spending 100 ms awake and 400 ms in a low power sleep mode. The experiments detailed in Chapter 5 allowed hormones to modify the duty cycle, however the period remained fixed for all experiments at 500 ms. At the 20% duty cycle the current draw of a node was 72 mA for 20% of the time and 12 mA for 80% of the time, for a total of 24 mA or 0.144 W at 6 V, for the Mesh experiments the power consumption was further reduced to 4 mA sleep current and 64 mA while awake for a total of 16 mA or 0.096 W.

A 6 V 7 Ah lead acid battery was chosen to power the node. This provided enough capacity for approximately 2.5 weeks (18 days) of operation using the 20% duty cycle. The addition of the solar panel extended this by, theoretically, 2.5 weeks in poor solar conditions. The battery was directly charged by a 2.5 W 6 V solar panel. A Zener diode was used to prevent the battery discharging through the solar panel at night.

The use of a duty cycle had significant impact on the way that the transmission and routing functioned, as they had to operate around this duty cycle approach.

4.3 Transmission and Routing

The transmission method used in the final design of the node architecture was designed to work around the varying duty cycle of each node. As the duty cycle of any node could not be relied upon to be constant, as it either varied by design or due to oscillator frequency error, the method used needed to be flexible. The broadcast mechanism was as simple as possible and accepted that there was no guarantee that a broadcast message would be received. The packet routing system used was designed to transmit and route data from nodes towards the base station. No functionality was added to route data between nodes or from the base station to a particular node.

4.3.1 Unicast Transmission

To transmit a unicast message all retries were disabled in the 802.15.4 firmware used on the XBee modules so that the number of retries was controllable by the microcontroller. When a unicast packet needed to be transmitted a maximum of 10 attempts are made, 50 ms apart. Therefore the maximum amount of time spent transmitting was 50 ms × 10 attempts = 500 ms. The receiving node, using the 20% duty cycle, was awake and able to receive packets for 100 ms. With the transmission attempts spaced 50 ms apart there should be multiple opportunities for the receiving node to receive a packet. If the duty cycle decreased or increased, there would be fewer or more opportunities to receive a packet. Using 10 transmission attempts 50 ms apart, theoretically, guarantees that the receiving node will be able to receive the packet. 50 ms provides enough time for the receiving node to receive the packet and transmit an acknowledgement back to the transmitting node.

Figure 4.4 shows an example of a node, shown in blue, trying to transmit to another node, shown in orange. The receiving node is in its 20% duty cycle, spending 100 ms awake and 400 ms sleeping. The transmitting node starts attempting to transmit a packet to the receiving node at time 250. It requires 8 attempts before a transmission coincides with the receiving node being awake.

This technique, very similar to the UPMA XMAC protocol discussed in Section 2.5.1.3, allowed nodes to communicate with another node whose duty cycle is unknown. There were some repercussions from using this technique, the most significant being that potentially a large number of extra packets are transmitted when they do not need to be. Each attempt to transmit a packet to a node could result in 10 transmissions of the packet. With multiple

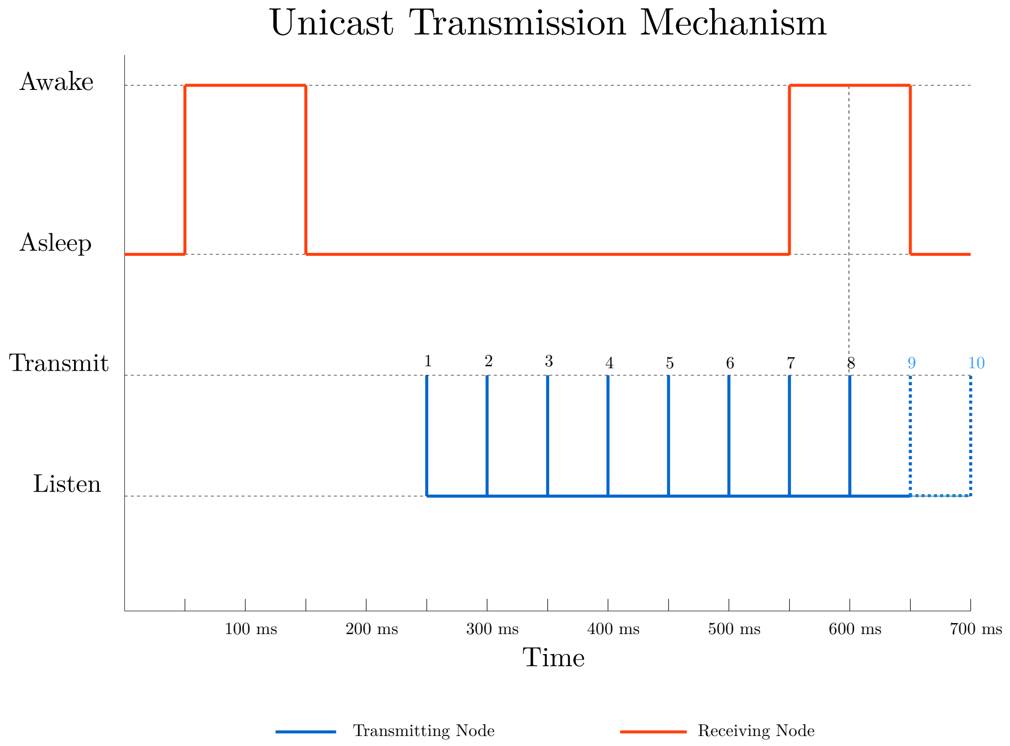

Figure 4.4: An example unicast transmission attempt between two nodes using the implemented unicast transmission method. Eight attempts are made before the receiving node is awake and able to receive the packet. The last two transmission attempts are shown for timing purposes.

nodes in range of each other this could result in , at worst, numNodes x 10 packets being transmitted in a short space of time. This produced congestion for other nodes that wished to transmit. The XBee modules have automatic collision detection with an exponential backoff that can result in a packet being transmitted and received multiple times, making the situation worse. This was due to the XBee internally waiting for some random amount of time before attempting to transmit a packet if it detected activity on the channel. If this random amount of time exceeded 50 ms, the microcontroller assumed that the packet was not received and requested that another packet was sent.

4.3.2 Broadcast Mechanism

Unlike the unicast transmission mechanism, there were no attempts made to retransmit a broadcast multiple times. This was due to broadcast messages not requiring an acknowledgement. As a result, it was possible for nodes to miss a broadcast packet. Most broadcast messages are intended to propagate throughout the whole network. When this was the case, the sequence number of the last broadcast packet that was received was stored so that a node could avoid retransmitting the same packet multiple times. The fact that broadcast packets were rebroadcast by any node that receives them means that there was multiple opportunities for a sleeping node to receive a broadcast packet eventually. If a particular broadcast packet was not supposed to be propagated throughout the network, the receiving node simply stored that sequence number and did not retransmit the broadcast packet.

4.3.3 Packet Routing

A very simple packet routing system was developed to allow nodes to transmit data across multiple hops or routes to the base station. The system was only designed to handle packets being routed towards the base station not between two specific nodes or from the base station to a node. All data was intended for the base station so making this assumption simplified the routing system substantially.

When the nodes were initially deployed at the start of an experiment, they entered a ‘deployment mode’. In this mode nodes left the XBee radios powered up and checking for packets constantly. Every 20 seconds each node would broadcast a message containing the addresses of any neighbouring nodes it had detected. By leaving the nodes in this mode while the network was being set up, each node was able to discover any neighbouring nodes. Once all of the nodes were deployed the NetDeployer software described in Section 4.4.2.1 was used to manually create routes between nodes. Each route was assigned a number which dictated in what order routes were used. The same routes with the same priorities were assigned to the same nodes in each experiment. These routes were then uploaded to the nodes and the experiment started.

The stored routes were used as a simple routing table in the event that a packet was received to be forwarded towards the base station or when the node generated a packet to transmit. When a packet needed to be transmitted, the node attempted to transmit it to the first route in the routing table using the unicast transmission method described in Section 4.3.1. If this failed then the node attempted to forward the packet using the next route in its routing table. This process was repeated until the node was out of possible routes at which point the packet was dropped.

The very limited memory, 2 Kbytes RAM, on the microcontroller meant that storing large routing tables or buffering packets was not an option. The simplicity and small memory footprint, 4 bytes of address and 1 byte of route priority,of this routing system made up for the fact that packets could be dropped.

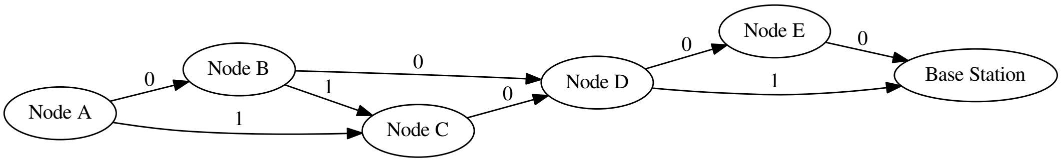

Figure 4.5: Example of routing between nodes. The number next to each link represents the order in which that node will use the links. For example, Node A can transmit via Node B or Node C but will attempt to transmit to Node B before Node C.

Figure 4.5 shows an example of routes between nodes using the routing system described above. Each node has one or more routes to use for transmitting data and each route has an associated number that represents the order in which it will be tried. If Node B were to forward a packet from Node A, or had some data to transmit to the base station, it would attempt to transmit to Node D before trying Node C.

The manual design of the routing between nodes allowed for specific scenarios to be created and replicated between experiments. While the result is not a full mesh network; each node has a preselected set of nodes it can communicate through. As a result, there are links between nodes that could be used but are not. This was in an effort to keep the behaviour of the network as similar as possible between experiments. While not a fully adaptive mesh, the routing method chosen does provide the ability for the network to cope with node failure as long as the nodes that rely on the failed node have alternative routes defined.

4.4 Deployment

The location chosen for deployment of the wireless sensor network was near the village of Ystumtuen in mid Wales. The location contains a diverse selection of sites that would be subject to different levels of exposure to the elements. These include exposed ridges, sheltered ditches, lightly wooded and flat areas. Also of interest, are a house and the track leading to the house. The site is accessible by road unless weather conditions are particularly bad and access around the site is by foot only. There is one location, the house, that is able to provide mains power for a base station unit and a slow internet connection for remote work. It is believed that the location provides an interesting test bed for micro-climate monitoring. Macro weather information is available from the met office at nearby sites at the Gogerddan, Cwmystwyth and Trawsgoed weather stations.

4.4.1 Network Topology

To provide a suitably diverse test bed, the network topology was designed with several scenarios in mind. In recognition of the difficulties of real world deployments, the topology was designed to represent difficult situations. Two different topologies were used, one for the initial five node experiments and another for the 20 node experiments. The 5 node topology was a star topology with the base station in the centre. This enabled experimentation to be carried out without routing being a determining factor. Once the number of nodes increased the topology was changed and multi-hop routing was employed to better represent a real world deployment.

4.4.1.1 5 Node Topology

The network layout for the five node experiments consisted of a simple star network. The base station node was situated indoors and had a constant mains power source. The five sensor nodes were all situated within 100 m of the base station. One node was placed at a high elevation on an exposed ridge, another at the bottom of the hill. One node was placed on the ground near the house (that the base station was located inside), another among a tree plantation on a nearby hill. The final node around 10 m from the house on a fence line.

4.4.1.2 20 Node Topology

The topology of the wireless sensor network remained fixed for the duration of 20 node experiments, so as to have as similar experimental set up as possible. Figure 4.6 shows the position of nodes and the routes between them. Where a node has multiple routes a yellow dot marks the route tried first. The locations chosen for the nodes provided a number of interesting scenarios likely to be encountered in a real world deployment. These scenarios are as follows:

- A cluster of four nodes, two hops away from the base station. This topology was intended to represent a situation where an area of interest was out of direct range from the base station node and a link to the cluster would require 1 to 2 nodes to bridge the distance.
- High density areas, with 15 or more nodes able to interfere with each other and cause congestion problems. This sort of topology is common in deployments aiming to provide a high geospatial density, for example crop monitoring or intruder detection and/or tracking.
- A "chain" of nodes without redundancy to stress test individual nodes involved in routing messages. A topology that may be relevant in areas that exhibit poor signal propagation. Dense woodland, rivers or rocky areas with high metallic content.

The signal quality between several nodes was poor, a received signal strength of around -90 dBm or worse, which is very close to the -92 dBm receiver sensitivity. There are many situations where sensor nodes would not be able to be deployed in optimal or "perfect" locations, as such it is felt that the network topology utilised in the 20 node experiments represents real world use cases well.

The routing remained fixed throughout all of the experiments. The routing layout was decided upon initially and before each experiment, every node was redeployed and the routing data re-uploaded. Where a node had a choice of routing data through multiple paths, the priority of the paths was also constant throughout all experiments.

The number of nodes was limited by the available resources and time available for the construction, testing and deployment of the nodes. As it was, more than 20 nodes would have made deployment and retrieval a multi-day process and costs required for equipment, construction and maintenance would have been too high for the scope of this thesis.

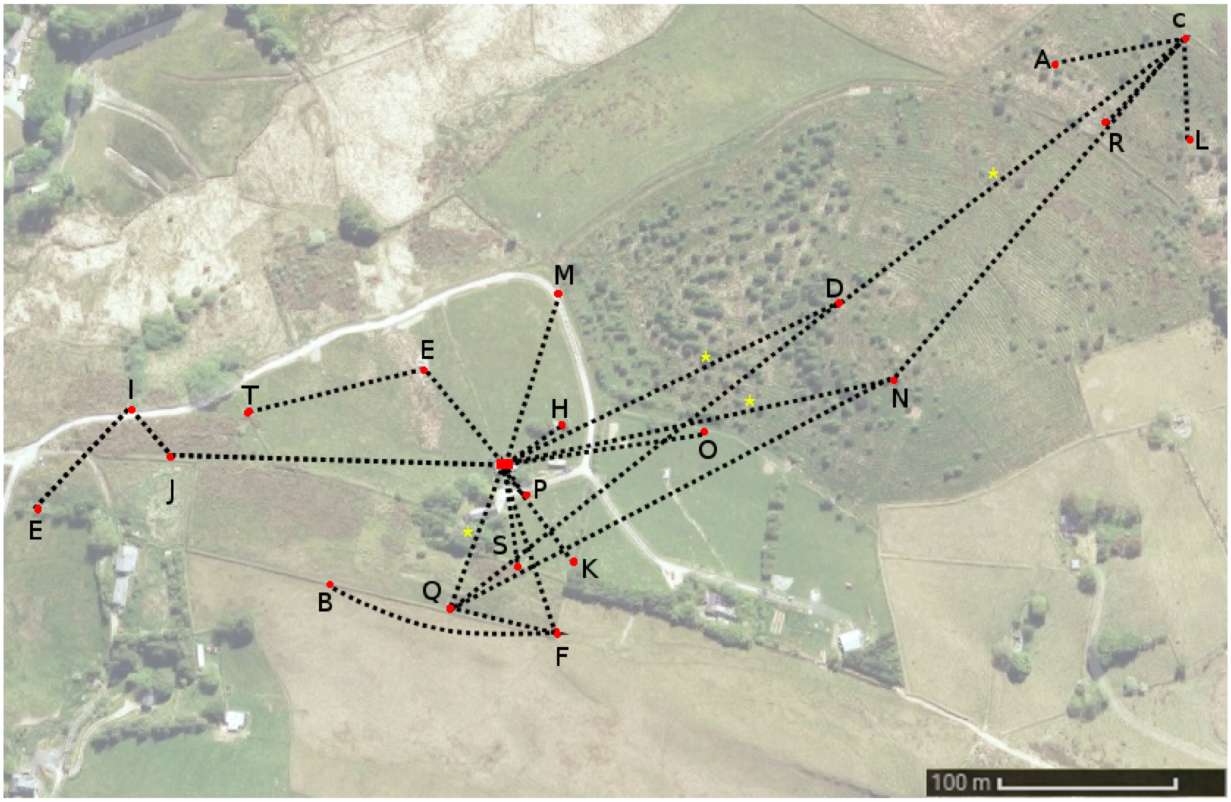

Figure 4.6: Map showing the locations of the 20 nodes (in red) and the routes between nodes. Where a node has two possible routes the one tried first is marked with a yellow dot. Map data ©2015 Google, Getmapping plc.

4.4.2 Deployment Procedure

The procedure for retrieving and deploying the network is detailed in the following section. The same procedure was carried out for each experiment so as to provide as similar environment as possible. Each node would be entirely "fresh". Routing tables would be reset and batteries replaced if necessary. Each node was put in a specific location so that any differences in behaviour due to the physical hardware or quirks of that particular node would remain consistent across all the experiments and minimise differences between experiments due to equipment.

The mounting hardware was set up just once and consisted of steel scaffold tubing either sunk into the ground or secured to trees or fences. This provided solid, stable and non deteriorating mounting point for the sensor nodes.

The procedure for retrieving the nodes and data from one experiment and re-deployment of the nodes for the next experiment is detailed below.

- Data was retrieved from the base station and copied to two USB flash drives and then deleted from the base station. The base station was reset.
- All nodes were collected and brought back to a central point, where they could be worked on in shelter. Here they were disassembled, the batteries were removed and the nodes checked for any physical damage.
- The nodes were reprogrammed with the new experiment code. A small group of the nodes were then fully deployed in the immediate vicinity of the base station to test the deployment process with the new experiment code. If successful, these nodes were reset and their EEPROM cleared.
- Any nodes whose batteries were not nearly full (less than 6.4 V) had their batteries replaced with fully charged batteries.
- The nodes were reconnected to their batteries and reassembled. When powered, the nodes went into a listening state awaiting, the command to enter the deployment state.
- Nodes were returned to their locations and fastened to their mounting poles. Antennae connections were checked.
- The base station was started and began broadcasting neighbour packets. Once all nodes were in their correct locations, a broadcast packet was transmitted that caused the nodes to enter their deployment state. In this state, nodes listened for neighbour packets to form their own lists of neighbours and periodically broadcast the neighbour information so other nodes could do the same.
- Using the NetDeployer software, a connectivity map was built up, allowing any problems to be found and resolved.
- The routing information for the network was created using the NetDeployer software and a routing plan. The routing tables for each node were generated and sent to the nodes, where they were processed and sent back to the NetDeployer software to confirm it had been received successfully. Once confirmed, the routing table was stored in EEPROM.

- When all nodes had their routing information uploaded, the base station neighbour broadcast was stopped. A “deploy” packet was broadcast using either the base station or the NetDeployer software and the nodes entered their normal operational state.

- The base station was used to check that all nodes were transmitting data and the experiment was left to run.

The whole procedure required a full day to complete. Collection of nodes took around 2 – 3 hours for one person. The disassembly, reprogramming and testing took 2 hours and putting the nodes back in their positions another 2 – 3 hours. Setting up the routes, checking that they were correct and deploying the network took 1.5 – 2 hours. On a good day, with good weather and no problems, an 8 hour turnaround was achievable by one person, conversely poor weather and a few problems could extend this by a further 4 – 6 hours.

4.4.2.1 NetDeployer Software

To aid development and testing of the sensor node hardware and software, a piece of helper software was written. Named NetDeployer, it provided functionality to enable deployment of nodes as well as visualizing the network topology. NetDeployer was written in Java and used the popular RxTx serial communications library for cross platform serial support. It has been tested, successfully, on OSX 10.7 (Lion), Windows 7, Windows 8 and several Linux distributions including Linux Mint 14. While there are existing pieces of software that offer similar functionality, custom written software better suited the needs of the project.

The following is a list of the main features available in the software.

- Designed to be easy to use on a touchscreen device, in the field.

- Render a map, using georeferenced images from google maps with nodes in their correct locations.

- Allow the addition, modification and deletion of nodes from a map.

- Store information about a node, including:
    - Latitude and Longitude.

  - Address.
  - Name.
  - Links between nodes, including signal quality.

- Display the links, with associated signal quality, between nodes and whether they are uni or bi-directional.
- Connect, via serial, to an XBee module.
- Mute and Unmute an individual or all nodes.
- Clear the EEPROM of an individual or all nodes.
- Set up and visualise routing between nodes.
- Transmit routing information to nodes and verify the information was correctly received.
- Deploy the network.
- Saving a map with all associated data.

4.5 Calibration

Documentation for the meteorology sensors utilised by the nodes, states that they were calibrated in-factory. Whilst factory calibration may have occurred, the devices used did not appear to be calibrated with respect to each other. The following section discusses this and demonstrates an attempt at post sampling calibration. This is a strategy that could be employed to calibrate the data gathered by the sensor network. The data to be calibrated is taken from one of the preliminary development deployments. Figure 4.7 shows a large discrepancy between the pressure measurements recorded by each sensor. A cursory examination of the graph indicated that the changes in pressure are the same between each node, however there was an offset. It was not possible for the difference, almost 40 mBar, to be attributed to a difference in height as the range of deployed altitudes of the nodes was approximately 20 m. Similar offsets were apparent in temperature, humidity and light levels.

Post calibration of the data was carried out in the following manner. On the 5/6/13, between 14:00 and 14:45, the five sensor nodes used to collect the data were deployed next to each other in a shaded area of an office. They were left for a period of 45 minutes, to allow the sensors to stabilise. Windows and doors were kept closed and the room was not equipped with air conditioning or heating equipment.

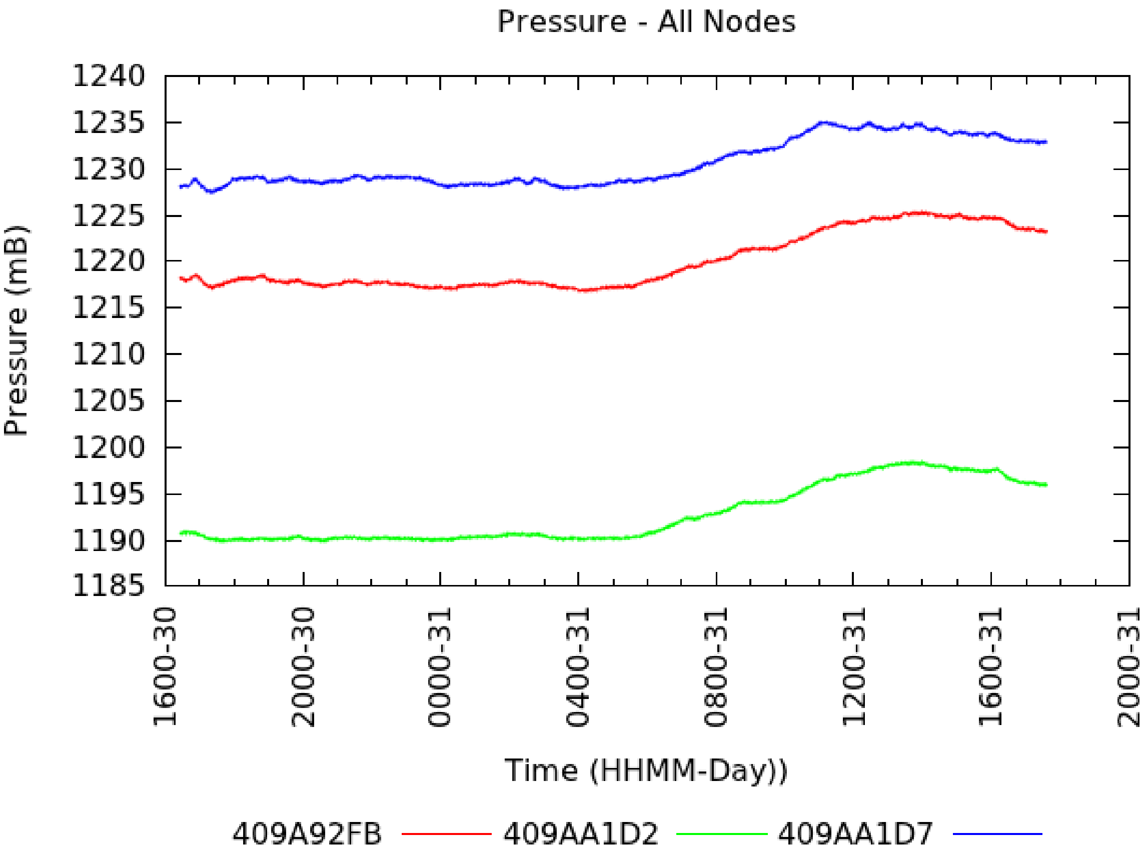

Figure 4.7: Measured pressure over a 24 hr period on 30-31 May 2013.

4.5.1 Pressure

During the preliminary deployment, the sensor nodes were all within 250 m of each other. The resulting difference in recorded atmospheric pressure should have been very small, as the maximum height difference was approximately 25 m. As a result, there should have been a difference of approximately 2.4 mBar between the highest and lowest node.

Table 4.2: Calculated pressure offsets for each node. The official pressure was 1021 mBar and an altitude correction of 7.8 mBar was used.

Node	Recorded Pressure	Required Offset (mBar)
409A92FB	1258.54 – 7.8	229.74
409A9301	1233.97 – 7.8	205.17
409A9307	1220.33 – 7.8	191.53
409AA1D2	1231.97 – 7.8	203.17
409AA1D7	1272.12 – 7.8	243.32

Current and historical pressure data is available from the Met Office. The offset for each pressure sensor was then calculated by subtracting the recorded pressure from the official pressure measurement. The official pressure measurement at sea level[2] was 1021 mBar. The building in which the calibration was undertaken was 65 m above sea level. Using the low altitude conversion rate of 1.2 kPa per 100 m, pressure measurements taken in the office were lowered by 7.8 mBar. This resulted in a sea level pressure measurement, from which the official measurement can be subtracted.

There was a large variance in measurements from the uncalibrated pressure sensors. The calculated offsets are shown in Table 4.2.

4.5.2 Temperature

Temperature offsets could be calculated by leaving each node on long enough for the temperature sensor to stabilise. A very accurate, pre-calibrated sensor was then used to record the temperature and calculate an offset.

The temperature, as measured by a calibrated Fluke 233 multimeter, using the temperature probe, was 24.7 °C. The calculated offsets are shown in Table 4.3.

4.5.3 Other Sensors

The light level sensors used would have been harder to calibrate as the measurement output is not in any unit. The sensor was simply connected to an ADC (Analog to Digital

---

[2] Taken from the Met Office Trawsgoed weather station 12 miles from the calibration location

Table 4.3: Calculated temperature offsets, using 24.7 °C, for both temperature sensors on each node.

Node	Temp BMP085	Temp SHT15	Temp BMP085 Offset	Temp SHT15 Offset
409A92FB	30.24	29.65	-5.54	-4.95
409A9301	30.86	29.51	-6.16	-4.81
409A9307	31.67	30.47	-6.97	-5.77
409AA1D2	31.3	30.08	-6.6	-5.38
409AA1D7	31.63	30.1	-6.93	-5.4

Converter) pin on the microcontroller. One method would have been to place each sensor in a uniformly lit environment and calibrate against the lumen value.

Voltage can be measured with a digital multimeter to a high degree of accuracy.

Humidity is harder to calibrate as a uniformly humid environment is required. One method is to seal the humidity sensor in an airtight container with saturated salt solutions. These solutions maintain an approximately constant humidity in a sealed container. Sodium Chloride will, once stabilised, give a reading of approximately 75% RH. This would be a fairly time consuming process, as an alternative, one sensor could be picked as the baseline and offsets for the other sensors calculated.

4.5.4 Results of Calibration

The calibration data was used to post-process the data from one of the preliminary deployments. An example of the calibrated data, in this case pressure, can be seen in Figure 4.8.

The differences between each node are now considerably smaller and are now explained by differences in altitude. For example, node 409AA1D7 was the highest altitude node, some 20 m higher than the other nodes. Accordingly, it can be seen that it has recorded a lower pressure, approximately 2 mBar lower, than the other nodes.

Using official pressure at 12 noon on the 31st of May 2013 (1020 mBar) and compensating for altitude (350 m), the expected pressure is 978.2924 mBar. This compares to the 994 mBar seen the data, an error of almost 16 mBar. This is possibly due to calibrating indoors, but also that the Met Office data is from 12 miles away.

The author feels that given the low cost nature of the sensors, to obtain reliable and

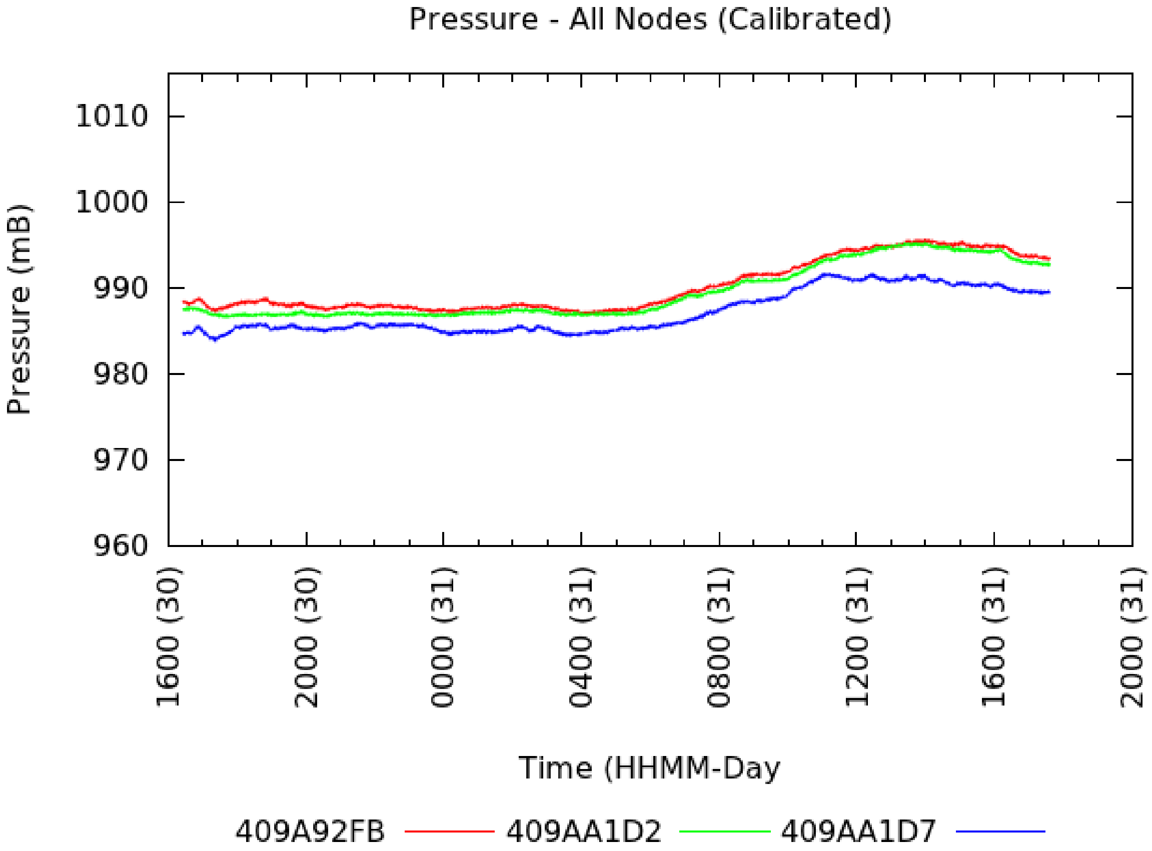

Figure 4.8: Calibrated pressure recorded during the proof of concept deployment. The differences in pressure are due to altitude.

accurate data, each type of sensor would need to be carefully profiled and pre/post calibrated. The locations at which these types of sensors would be deployed would likely be too remote to be able to calibrate the sensors frequently. As obtaining reliable and accurate meteorology data is not the focus of this work and the sensor data presented throughout will be un-calibrated. However, the methods discussed above may provide a template for calibration for other, or future, work.

4.6 Meteorology

There were several different factors in the decision to monitor meteorological conditions rather than another scenario. The most important was that it encouraged deployment of the wireless sensor network outside, in a real environment with harsh, non-optimal conditions. This would more closely represent the environment monitoring scenario presented in the introductory chapter. It was felt that any successes achieved as a result, would be more significant and applicable to real sensing scenarios. Another useful attribute of environmental sensing is that regardless of location, it can be carried out in the same manner. This should allow environment monitoring using the methods in this thesis to be performed elsewhere, with the expectation of similar performance. Although high precision meteorology sensors are expensive, there are cheaper versions available that are simple to use and still provide valuable data. This makes a real deployment feasible from a cost perspective and helps cut down on the amount of time required to get an initial system running. The data provided by the sensors can also be calibrated by and compared to local met office weather stations, if necessary.

It was also felt that high resolution environmental sensing as a service was in high demand in other research areas. There are numerous locations which are remote, harsh or expensive to survey. As a result only limited data is collected. Robust and autonomous solutions to these sensing problems could have a significant impact.

4.7 Hormone Methodology

This section discusses how the human endocrine system was used as inspiration for the work in this thesis. The hormones used in experiments are discussed as well as how they were produced and decayed. The methods used to combine multiple hormones and transmit hormones are presented and discussed. Whilst inspiration was taken from how hormones in the human endocrine system behave, the decision was made from the beginning to not attempt to replicate specific hormones, hormone cascades or regulatory systems. Some components of the endocrine system, such as hormone cascades are not used at all. This was not because they are considered unimportant or useless, but because the complexity involved in using them in a wireless sensor network is beyond the scope of this thesis. The behaviour of each hormone is designed to try to produce a particular effect.

For this body of work, the endocrine system was considered to include paracrine and autocrine signalling. These are forms of cellular chemical signalling that behave in similar ways to endocrine signalling. Endocrine signalling involves the transport of chemical messages, called hormones, using the circulatory system. In autocrine signalling, a cell producing a hormone, or 'autocrine agent', is receptive to that same hormone. Paracrine signalling is a form of cell to cell communication whereby a cell can produce some chemical, or 'paracrine factor', that can cause changes in nearby cells. These systems all involve the production of some signalling molecule, often referred to as a hormone, which can affect cellular behaviour.

The hormones presented in the following sections are layered upon each other one at a time in the order that they are presented in this chapter. The first, non-control, experiment will utilise the selfish hormone. The next experiment will utilise the selfish and anger hormones. This process will repeat until all five hormones are utilised simultaneously. Due to the amount of time required to deploy and run real world experiments, it is not possible to run each hormone in isolation before integrating it. It is also not possible, for the same reasons, to run repeat experiments. As such, the results of this work are expected to be indicative of the real world performance. The integration of all of the hormones into the control system is intended to mimic the behaviour of hormones in the human endocrine system. Each hormone is able to affect parts of the system at the same time. Hormones may produce similar effects in some system or may even be antagonistic. As each hormone decays, so too does its effect on the system. This allows for soft switches in behaviour and the combination of multiple behaviours in a simple manner.

This work does not provide a rigid artificial endocrine framework in which a node represents some exact equivalent from the human endocrine system. The concept that only endocrine glands can produce hormones is one that is no longer considered accurate [73]. As a result it is perhaps best to consider nodes in a wireless sensor network to be cells capable both of producing and being receptive to hormones. The transport mechanism for hormones in a wireless sensor network is radio waves, with proximity to a node being dictated by signal strength rather than geographical proximity.

4.7.1 Hormones Used

In total, five different hormones were used to modify the behaviour of a wireless sensor network. Given the time available it was felt that investigating the effects of more than 5 hormones in real world network deployments would not be possible. They were organised into two groups; hormones that affected power consumption and hormones that affected data quality. The selfish and anger hormones were considered to be 'power' hormones, whilst the light, centre and wind hormones were considered ‘data’ hormones. This section provides an overview of each hormone. The selfish hormone was thought of as working in an autocrine manner as the hormone that a node produced affected itself. The anger, centre and wind hormones were transported throughout the network in an endocrine-like manner. Lastly, the light hormone was distributed in a paracrine like way, with only ‘nearby’ nodes receiving the hormone.

To investigate these hormones each hormone will be added to the node control system and the network deployed. The order in which the hormones are presented in this chapter is the order in which they are added to the control system. The first, non control, experiment will have just the selfish hormone. The next will have the selfish and anger hormones. By the final experiment all five hormones will be enabled at the same time.

4.7.2 Selfish Hormone Mechanism

In order to allow hormone based control of a node’s power consumption and power management a hormone that responded to available power was required. This hormone is referred to as “Selfish Hormone” as it only takes into account the node’s battery state, not the state of other nodes. The goal of this hormone is to enable an individual sensor node to better use its power with no regard for the battery state of any other sensor node. This is an important aspect of the environmental monitoring scenario, as each node is considered important rather than disposable.

The following factors are considered when designing the operation of the Selfish Hormone. The duty cycle approach to power management allowed for continuous and fine grained control of the power consumption of each node. The “Selfish Hormone” is designed to exploit this by increasing or decreasing the duty cycle based on the power available to each node. In a similar fashion to the human endocrine system, the amount of this hormone produced in the control system is related to an input, in this case the battery terminal voltage. The battery terminal voltage provided an approximation of remaining energy. As in the human body, the hormone had a maximum production rate at which no more could be produced even if a higher input was given.

According to the battery’s data sheet a terminal voltage range of 5.6/5.8 V to 6.4/6.55 V maps approximately linearly (0 to 100%) to the remaining capacity of the battery. The valid range of battery voltages chosen for hormone production was 5.9 V to 6.6 V.

The reasons for choosing this range were as follows. The lower bound, 5.9 V, is higher than the fully discharged battery voltage, which is around 5.6 V. Fully discharging lead acid batteries can damage them and hinder future charging, as the charging employed was fairly primitive it was desired that this be avoided. By using 5.9 V as the minimum acceptable battery voltage, long term battery health could be promoted. A battery voltage of 6.6 V was chosen as the upper bound for hormone production. This voltage is higher than a fully charged battery (6.5 V) however the attached solar panel was, in periods of bright sunshine, capable of pulling the terminal voltage higher than this. Thus, while the battery may be "full", there is excess energy from the solar panel that can be used. This range, 5.9 V to 6.6 V is the range of voltages that input to the selfish hormone production system. Voltages above 6.6 V are considered to be 6.6 V and voltages below 5.9 V resulted in no hormone production in an attempt to save power.

The effect of the Selfish hormone is to modify the power duty cycle of the node. The range of duty cycle the "Selfish Hormone" is able to control was chosen to be 5% to 50% duty cycle. This range is large enough that it allows a strong degree of control over power consumption while allowing a significant portion of the duty cycle range to be controlled by other hormones or hormone systems. The minimum of the range 5%, or 25 ms, results in a small chance, 50%, of being able to forward messages from other sensor nodes.

The relationship between battery voltage and hormone release was a simple second order polynomial scaled to the range 25 to 250. These values correspond to times in milliseconds that the node remains awake. Since the duty cycle period is 500 ms, 25 ms corresponds to 5% and 250 ms to 50%. The hormone takes the battery terminal voltage as input and outputs a quantity of hormone. The polynomial behaviour was chosen as it was considered desirable that as the amount of available energy reached 100% battery capacity the energy expenditure should increase significantly so as to bring the battery voltage down. This, again, promotes battery longevity as long term storage at high charge levels degrades battery capacity. Figure 4.9 shows the resulting plot of battery terminal voltage against "Selfish Hormone" production. The selfish hormone was calculated as follows:

$$V = \begin{cases} 6.6, & \text{if } V > 6.6 \\ V, & \text{otherwise} \end{cases} \tag{4.1}$$

The input voltage V is limited at 6.6 V.

$$x = (\mathrm{V} - 5.9)^2 \tag{4.2}$$

x is the second order polynomial component that creates the desired response to the input voltage.

(4.3)

Calculate Hormone Scale Constant (HSC):

$$\mathrm{HSC} = \frac{\mathrm{H_{max}} - \mathrm{H_{min}}}{\mathrm{V_{range}}^2} \tag{4.4}$$

The HSC value is calculated using H_{max} the desired maximum hormone value and H_{min} the minimum hormone value and Vrange is the range of input voltage $(6.6 - 5.9)$.

$$\mathrm{H_{selfish}} = \begin{cases} 25 + (x \times \mathrm{HSC}), & \text{if } V >= 5.9 \\ 0, & \text{otherwise} \end{cases} \tag{4.5}$$

H is the quantity of hormone produced which is determined by multiplying x (Equation 4.2) by HSC (Equation 4.3) and adding it to 25 to bring the hormone output into the desired output range of 25–250. If the input voltage falls below 5.9 V no hormone is produced in an effort to prevent over discharge of the battery.

To provide an example, if the voltage was within the acceptable voltage input range of between 5.9 V and 6.6 V then 5.9 V was subtracted from it and the result squared. The amount of hormone produced was calculated by applying the scaling constant HSC and adding 25 so that the possible range of output values lay within 25 to 250. HSC was calculated by dividing the desired hormone range by the squared voltage range. In this case the desired hormone range was $250 - 25$, which was divided by $(6.6V - 5.9V)^2 = 0.7^2$ and resulted in $HSC = 459.18$. As an example, to calculate the hormone produced at the maximum input voltage of 6.6 V and the minimum input voltage of 5.9 V:

$$\begin{aligned} \mathrm{H}_{\mathrm{max}} &= 25 + ((6.6V - 5.9V)^2 \times 459.18) \\ \mathrm{H}_{\mathrm{max}} &= 25 + 225 \\ \mathrm{H}_{\mathrm{max}} &= 250 \end{aligned} \tag{4.6}$$

$$\begin{aligned} \mathrm{H}_{\mathrm{min}} &= 25 + ((5.9V - 5.9V)^2 \times 459.18) \\ \mathrm{H}_{\mathrm{min}} &= 25 + 0 \\ \mathrm{H}_{\mathrm{min}} &= 25 \end{aligned} \tag{4.7}$$

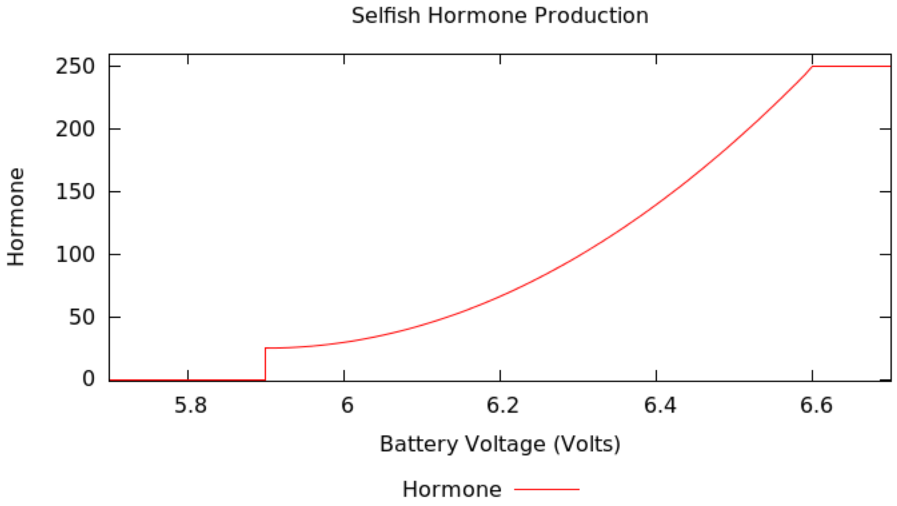

Figure 4.9: Selfish hormone production plotted against battery voltage.

The “Selfish Hormone” was consumed at a rate of 1 unit per ms whilst the node was awake. Once the remaining amount of hormone reached 0, a node entered sleep mode for the remaining portion of the duty cycle period. Thus, a node producing 100 units of hormone consumed all of the hormone in 100 ms and therefore remained “awake” for 100 ms, then slept for 400 ms, a duty cycle of 20%. The system produced a minimum of 25 units and a maximum of 250 units of “Selfish Hormone”, which resulted in the previously mentioned range of 5% to 50% duty cycle. A voltage below 5.9 V resulted in 0 hormone being produced and therefore a duty cycle of almost 0%. This gave a node a chance to “hibernate” and if it were to completely run out of power, a chance to recover a little.

This process occurred every half a second, so the node was able to respond very quickly to changes in power availability so as to be able to best exploit the available power. For example, on a cloudy day, if the sky were to periodically clear due to cloud movement, the node would be able to adapt its power consumption as more or less power was available from the solar panel.

4.7.3 Anger Hormone

The second hormone designed to affect the use of power is the "anger" hormone. It is designed to counter the selfish hormone which provides a node with the ability to selfishly manage its own power consumption, potentially to the detriment of other nodes. This has the potential to result in other nodes struggling to transmit data due to one node selfishly conserving power. To counteract this, the anger hormone is designed to allow a node to affect the power consumption of another node on which it relies for transmitting data to the base station. This is achieved by increasing the duty cycle of the receiving node thus making it more available to route data. This allowed the needs of a particular node, and nodes that relied on it, to be traded off against each other. For example in a situation where 3 nodes all rely on one node to route their data, this 'key' node is able to lower its power consumption using the selfish hormone. This would be detrimental to the nodes that rely on it and they could, using the anger hormone, make the key node increase its power consumption.

Due to the transmission and forwarding technique used, there is a chance that a message will not be able to be successfully transmitted. This is especially the case when combined with the Selfish Hormone approach, as the period of time the receiving node has to receive a message is very low. To generate anger hormone a node attempts to transmit a data message as normal. However, if the message is not successfully transmitted and all attempts to route the packet via an alternate route fail, anger hormone is produced. As more messages fail to be transmitted successfully, more anger hormone accumulates. Each failed transmission produces 3 units of hormone as shown in Equation 4.8. The hormone builds up linearly and slowly so as to have a gradual effect on the target node. This quantity of hormone is considered to be the "local anger hormone" and represents the connection quality in terms of how many failed transmissions have occurred. This local anger hormone value is included in the data packets transmitted to the base station and does not affect the node that produces it only a receiving node.

$$\mathrm{H}_{\mathrm{anger}} = \begin{cases} \mathrm{H}_{\mathrm{anger}} + 3, & \text{if } \mathrm{H}_{\mathrm{anger}} < 250 \\ \mathrm{H}_{\mathrm{anger}}, & \text{otherwise} \end{cases} \tag{4.8}$$

As the anger hormone also affects the duty cycle of a node it is limited to between 0 and 250. The quantity of hormone represents the amount of duty cycle to remain awake. The hormone decays at a rate of 1 per attempt to transmit. Thus a node that failed to transmit once, then succeeded in transmitting would have an anger hormone level of 2 units.

During the routing process, a node that is routing a packet towards the base station checks the hop count of the packet. If the hop count is 0, then before incrementing the hop count and routing the packet towards the base station, the node extracts the local anger hormone value. This is due to the fact that the node performing routing is the first node to encounter this packet. Any local anger hormone contained within was produced as a result of the sending node being unable to communicate with the routing node. As such, the anger hormone was "directed" at the routing node. The quantity of anger hormone is added to an internal "remote anger hormone" value. The remote anger hormone value allows the anger hormone of multiple "child" nodes to be accumulated. The remote anger hormone is limited to the same range and decays in the same manner as the local anger hormone. This remote anger hormone value controls 50% of the duty cycle of a node. The remote anger hormone value is added to the amount of time spent awake and listening for transmissions produced by the selfish hormone. Thus if the duty cycle was previously 25% (125 ms awake) and the remote anger hormone value is 10 then the duty cycle for that node becomes 135 ms or 27%. This mechanism allows a node who's data packets are not being reliably received to attempt to modify the behaviour of the receiving node in an attempt to decrease packet loss.

Figure 4.10 shows the remote and local anger hormone levels, in the format "Local-Value:RemoteValue", for two nodes over time. At t0 NodeA has been unable to successfully transmit packets to NodeB for the last three attempts. It's local anger hormone level is, therefore, 7. Each attempt to transmit decays the hormone by 1 and each failure increases

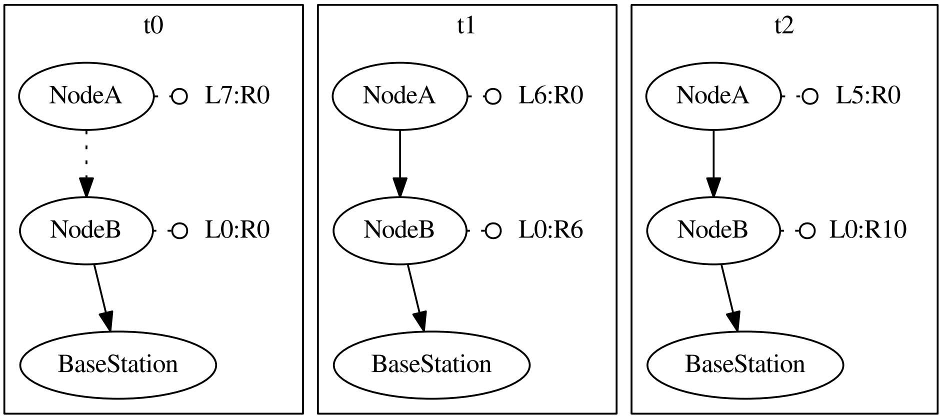

Figure 4.10: Example of local and remote hormone levels over time, for two nodes communicating with the base station. Anger hormone levels are represented in the form LocalValue:RemoteValue.

it by 3. Therefore NodeAs hormone level can be calculated as follows:

$$\begin{aligned} H_{anger}(NodeA) &= 0 - 1 + 3 - 1 + 3 - 1 + 3 \qquad (4.9) \\ H_{anger}(NodeA) &= 7 \end{aligned}$$

As the hormone level cannot fall below 0 the result of Equation 4.9 is 7. NodeB has not encountered any problems transmitting packets and therefore has a local anger level of 0.

At time t1, NodeA's local anger hormone level has decayed by 1 to a level of 6 and has successfully transmitted a packet to NodeB. NodeB has received the packet from NodeA and incorporated the 6 local anger hormone into it's remote anger hormone level.

By t2 NodeA's local anger has decayed again by 1 to 5 and NodeB's remote anger, having successfully transmitted a packet, has also decayed by 1 to 5. NodeA has also successfully transmitted another packet to NodeB, which has incorporated NodeA's new local anger level into it's remote anger level increasing it by 5 to a level of 10.

Once NodeA's local anger becomes 0, NodeB's remote anger level, and hence it's duty cycle and power consumption, will start to decrease. In the case of nodes communicating directly with the base station the anger value cannot be used by the base station as it has a fixed 100% duty cycle already.

4.7.4 Light Hormone

The first hormone aimed at improving data quality is the Light Hormone which to tries to improve the temporal resolution of the data recorded. When something being measured changes rapidly, it would be beneficial to increase the sensing rate to increase the data resolution of the event. Conversely having a higher sensing rate when nothing is changing is of no real use as a large amount of the data is redundant. The sensor node designed in Section 4.1 is able to measure the following; temperature, humidity, air pressure, battery voltage and light. Of these, the light is the most likely to change quickly. There were several reasons that the light value may change suddenly. Obstruction of the light sensor, moving cloud cover and shadows cast by the terrain or clouds. Ideally the light sensor would never be obstructed, however it was possible that leaves or other detritus could become lodged above the sensor. On a relatively clear day, moving cloud cover results in quickly moving areas of darkness on the ground. The transition from light to dark occurs quickly. If the network were to have a high enough temporal resolution, it may be possible to estimate the speed and direction of cloud movement. This would be an example of providing data that contained more information, or be of “better quality”. The last possibility, of the sensor node being quickly cast into shadow, could be due to changing cloud cover or changing inclination of the sun. In either case, the rapid change in conditions could well elicit changes in other measured parameters. Temperature, in particular, could change rapidly in response to a quick change in light level.

The light hormone is produced in response to a rapid light change detector, described in Section 4.7.4.1. When the detector detects a rapid light change it produces the maximum amount of light hormone, 11. The reason for the maximum to be 11 is due to the hormone combination method discussed in Section 4.7.9. The default sensing period (60 seconds) is divided by the light hormone value to determine the current sensing rate. The result of a light hormone value of 11 is that the node will sense and transmit a packet roughly every 5.5 seconds, the default sensing rate divided by 11. When all of the light hormone has decayed, the sensing rate returns to the default of once every 60 seconds. The light hormone decays at a rate of 0.08 units per second and therefore takes 2.5 minutes. This results in an increased rate of sensing that decreases back to the default sensing rate over 3.5 minutes. The decay time is a trade-off, the longer it is the longer the sensing rate of a node is elevated. However if the change is short then the increased sensing rate will produce a large quantity of data that is the same. It was decided that this relatively short amount of time is sufficient to capture the results of the rapid change in light level in a higher temporal resolution while not creating a large amount of redundant data.

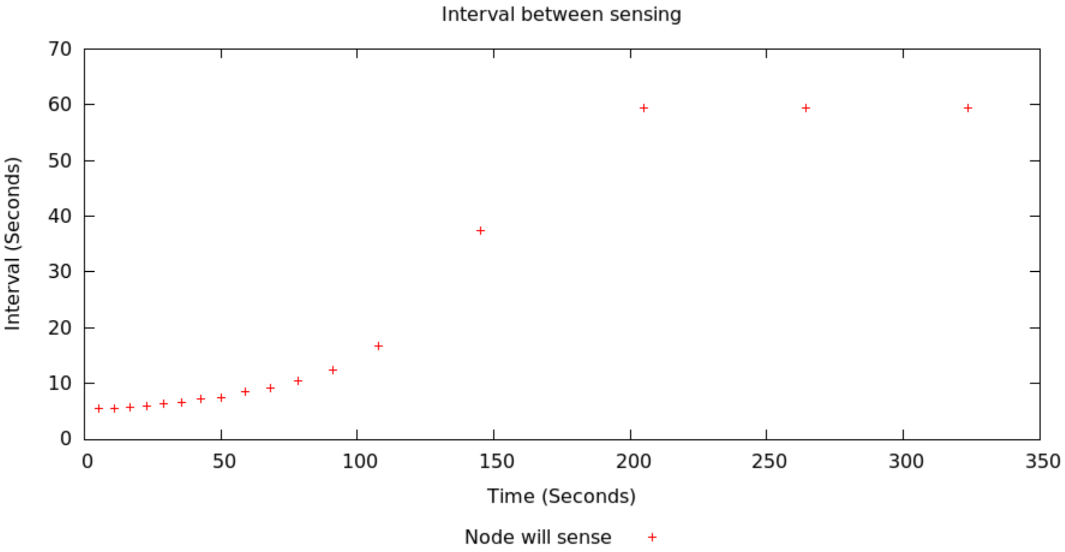

Figure 4.11: Graph showing the times at which a node would transmit if its light hormone level were set to maximum and left to decay. The resultant time, in seconds, between sensing events is shown on the yaxis.

Figure 4.11 shows the times at which a node would read its sensors if its light hormone level were to be set at the maximum and left to decay. The time between sensing events was very low for the first minute, relatively low for the second minute and after 3.5 minutes (210 seconds) returned to the base sensing interval of 60 seconds. Thus a gradual return to the default sensing rate is achieved. The rapid method for detecting rapid light change that produces light hormone is presented below.

4.7.4.1 Rapid Light Change Detection

There are several possible options for detecting "rapid" light changes. As the compute power on each node is very low, the method used needs to be simple and quick. One option would be to compare the current and previous light measurements. Due to the noisy nature of the light sensor, it is likely that this approach would lead to a huge number of detections. To explore the ideas further, it made sense to use some real world light data. Environmental data from a previous deployment was used to determine the values for the detector. The data was captured at 5.5 second intervals which lead to the decision to make one sample per 5.5 seconds the maximum sensing rate. This provided a good data set on which to trial the detection method.

A rapid change in light is considered to be a significant change in the light value, either an increase or decrease, that lasts longer than 22 seconds (4 * 5.5 seconds). The selection of the 22 second target is arbitrary it was considered to be a sensible value. This should rule out very rapid spikes or dips in the light value due to noise or periodic changes due to the environment, such as moving foliage. Two running averages of the light level were calculated, one short term and one long term. The short term average provides a smoothed approximation of the current light level, removing noise. The longer term average provides an estimate of the light level, without short term influences. Each average is updated every 5.5 seconds due to the frequency of the captured data. The short term average converges after approximately 27 seconds and the long term after approximately 2 minutes. Equations 4.11 and 4.10 detail the averaging functions used, where L is the sensed light level.

$$\mathrm{Avg}_{\mathrm{short}} = 0.95 * \mathrm{Avg}_{\mathrm{short}} + (0.05 * L) \tag{4.10}$$

$$\mathrm{Avg}_{\mathrm{long}} = 0.8 * \mathrm{Avg}_{\mathrm{long}} + (0.2 * L) \tag{4.11}$$

After updating the long and short term averages the absolute difference between the two is calculated. If this difference is greater than the threshold value of 10, 60 units of hormone are added to the light detection hormone level as described in Equation 4.12 where H_{ld} is the light detection hormone. The hormone decays at a rate of 30 units every 5.5 seconds. Once the light detection hormone exceeds 90 units the sensor node produces the maximum amount of light hormone, 11, and consuming all of the light detection hormone in the process. This results in the sensing rate of that node increasing, as detailed in the previous section. The values for this hormone were chosen so that they produced the desired effect of taking 22 seconds of light change to trigger the light hormone.

$$\mathrm{H}_{\mathrm{ld}} = \begin{cases} \mathrm{H}_{\mathrm{ld}} + 60, & \text{if } abs(\mathrm{Avg}_{\mathrm{long}} - \mathrm{Avg}_{\mathrm{short}}) >= 10 \\ \mathrm{H}_{\mathrm{ld}}, & \text{otherwise} \end{cases} \tag{4.12}$$

This system was tested on the previously mentioned data set. Days with particularly obvious light profiles, such as highly changeable or very constant light values, were examined in greater detail. The threshold value was tuned until the number of detection events and the conditions causing a trigger matched expectations.

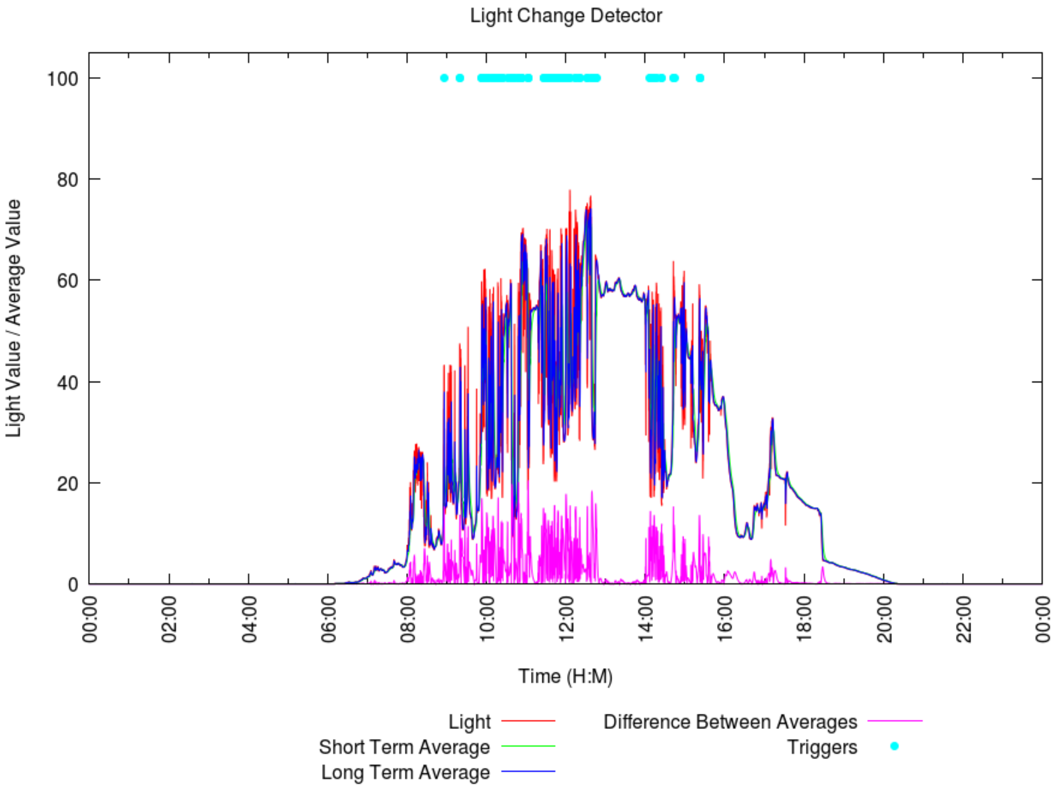

Figure 4.12: Graph showing the performance of the light change detector over one day, using data from a single node from the Anger Hormone experiment. Appendix B shows the performance of the detector over a longer period of time.

Figure 4.12 shows the performance of the detector over one day, using data from a single node. The light blue dots are detected events. During periods with high light variability a high number of events were detected. Between 13:00 and 14:00 and from 16:00 onwards the light was changing much more slowly and no events were detected. In total, this node detected 111 events over this one day. While the detector is not, by any means, perfect, it is able to detect large rapid changes in light level. It is felt that this is sufficient in this scenario.

The detecting sensor node, in addition to producing light hormone locally, broadcasts a packet to all of its neighbouring nodes. Any node that receives this packet will also produce the maximum amount of light hormone, thereby increasing its sensing rate. The result of this system is that a node takes 22 seconds to respond to the rapid light change event. After this it stimulates itself and any neighbours to sense the environment more frequently. This behaviour is analogous to a cell that uses autocrine and paracrine signalling (see Section 2.8.2).

4.7.5 Centre Hormone

The next hormone designed to affect data quality is the centre hormone. This is designed to allow a user to affect the behaviour of the wireless sensor network in an endocrine inspired manner.It is important that the sensor network be able to be directed to change its behaviour on command. A good example of a situation in which this would be useful, would be the knowledge that some event would be occurring at a given time. In the case of meteorological sensing, this could be an interesting weather front, a geological event, an astronomical event (such as a solar eclipse) or numerous others. It may be advantageous to increase the sensing rate of the network to better cover this period in time, even at the cost of power consumption. Alternatively, it may be desirable that the sensing rate should be decreased for some period.

This hormone should allow the testing of whether it is possible, within the endocrine framework, for a user to dramatically influence the behaviour of the sensor nodes. It would be possible to create a mechanism that provides the ability to overrule the existing hormones. In the spirit of the research here and to minimise the complexity of the control system we would like to design a system that fits within the endocrine framework. In keeping with this it was decided that control should be achieved through the use of the centre hormone.

The ability to generate centre hormone is added to the base station node. Any node would be able to produce centre hormone, however the base station is connected to a laptop which enables simpler human interaction. Upon instruction, the base station broadcasts a quantity of centre hormone to its neighbouring nodes. The quantity is selectable by the user and could be any integer value in the range -100 to 100. Values below 0 suppress the sensing rate and values above 0 promote, or increase, the sensing rate. The values of -100 and 100 were chosen so as to be easily interpretable by a human. The limits of the sensing rate is from a measurement every 5.5 seconds to every 12 minutes. This range is based on the default sensing range (60 seconds), from 11 times smaller to 11 times bigger. This allows a wide range of sensing rates from around 650 measurements an hour to 5. The effect of centre hormone on sensing rate is shown in Figure 4.13.

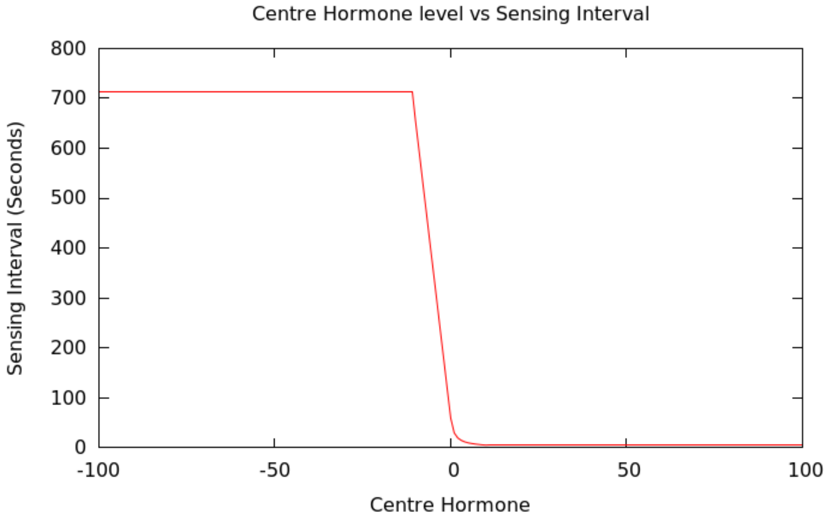

Figure 4.13: Interval between sensing events for a given centre hormone value. At a value of 0, ie no hormone, the sensing rate is the default 60 sensing events per hour.

To prolong the effect of the hormone, the hormone is designed to 'saturate' very quickly. Any hormone level less than -11 or more than 11 results in saturation. The practical upshot of which is that while the centre hormone is greater than 11 or less than -11 it will have maximal impact on the sensing rate. As a result with a constant hormone decay the effect can be prolonged by sending more than 11 or -11 centre hormone, resulting in a greater length of effect. Figure 4.14 shows the decay curve of the centre hormone over time. It can be seen that the maximum, either 100 or -100, hormone level decays to just less than 1 or -1, at which it had no effect, in 12 hours[3]. The hormone decays at an inverse exponential rate, being multiplied by 0.999975 four times a second. It takes around 7 hours for the hormone level to become unsaturated, at which point the dropping level of hormone results in the sensing rate gradually, over the course of the next 5 hours, returning to the "normal" rate of once per minute. This allows a user to increase or decrease the network sensing rate for up to 12 hours at a time, with maximal effect for around 7 hours before the sensing rate gradually returns to the default. 12 hours is considered to be long enough to

---

[3] The programming language used, C, truncates when casting a float to an integer, therefore any value below 1 becomes 0.

capture an interesting event, however the effect of the centre hormone could be prolonged by transmitting more at regular intervals.

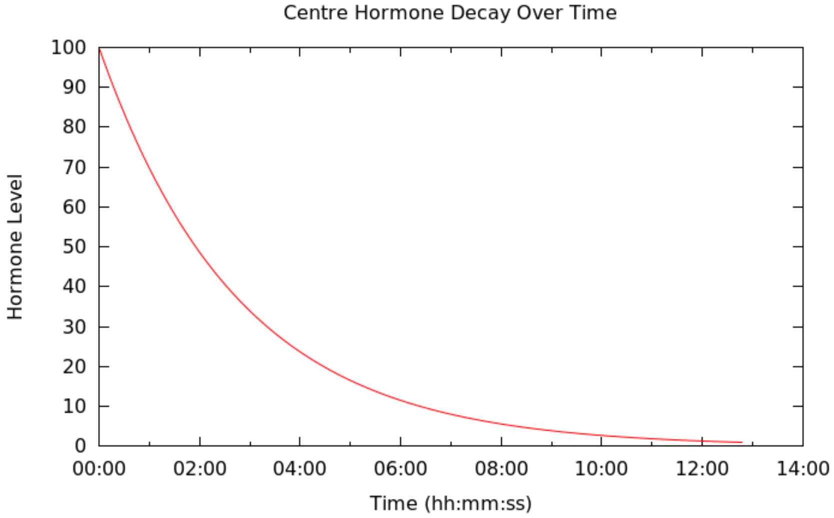

Figure 4.14: Value of the Centre Hormone over time as a result of it's decay rate.

The centre hormone, wind hormone and light hormone all combine in the manner described in Section 4.7.9.

Any nodes that receive the broadcast containing the centre hormone rebroadcast the packet unless they have already done so. This allows the hormone to propagate throughout the network and, theoretically allows each node a chance to receive the hormone.

4.7.6 Wind Hormone

The final hormone is designed to attempt to improve the quality of the data captured by using the wind speed and direction to modify sensing rate.

Two cases of wind speed and direction that may impact environment have been chosen. High wind speed and low wind direction variability. Higher wind speeds are typically related to changeable weather conditions and as such changes in the environment occur more quickly during periods of higher wind speed. Conversely, when the wind direction remains relatively constant, the environment changes more slowly, the wind acting in a constant manner.

A Furuno RO-Wind wind sensor is attached to the base station to capture the wind state. The RO-Wind is an ultrasonic wind speed and direction sensor, with an angular resolution of 1 degree and speed resolution of 0.1 knots. This is used to record the wind speed and direction at a rate of 1 Hz continuously.

The wind speed is prone to noise due to mounting location and local eddies due to the surrounding environment. A short term average of wind speed is taken to smooth out the values before the next step.

The Beaufort scale is a scale that relates the wind speed to observed conditions on land or sea. 2 on the scale (4 – 6 knots) marks a light breeze and as such a wind speed higher than 5 knots is the threshold above which it is considered to be windy. When the wind exceeds the wind speed threshold or 5 knots, the wind hormone is incremented by 1. The change in wind direction, in degrees, is also logged and the standard deviation of the change in wind direction over 100 seconds is calculated. This provides an estimate of the variability of the wind condition in the last 100 seconds. The amount of time that this standard deviation is calculated over was chosen arbitrarily as a reasonable value. The threshold at which the wind direction is considered "stable" is a standard deviation of 5 degrees over 100 seconds. If the wind is below this threshold 1 is taken away from the wind hormone.

These two parameters, short term averaged wind speed and standard deviation of the change in wind direction, are calculated every 1 second and are antagonistic as they affect the wind hormone in opposite directions. Equation 4.13 is calculated every every second on the base station. $\mathrm{H_w}$ is the wind hormone, ws is the wind speed and wd is the wind direction. The hormone is subject to a small decay as it is also multiplied by 0.99 each second.

$$\mathrm{H_w} = \begin{cases} \mathrm{H_w} + 1, & \text{if } ws > 5\text{knts} \\ \mathrm{H_w} - 1, & \text{if } wd < 5\text{deg} \end{cases} \tag{4.13}$$

This wind hormone needs to be sent to the nodes in the network to be able to have an effect. Therefore once per hour, the wind hormone is broadcast, by the base station, to the network. Once received by a node, the wind hormone works in the same way as the centre hormone. The only difference being the much shorter decay time of 1 hour to coincide with the 1 hour period the base station calculates the wind hormone over.

The effects of the light, centre and wind hormones are combined, as described in Section 4.7.9. Thus, a node that is promoted to transmit by the centre hormone and light hormone but equally suppressed by the wind hormone has the default sensing rate of once per minute.

The wind and light hormones would, however, decay much more quickly than the centre hormone.

4.7.7 Hormone Production

Several different methods for determining the quantity of a hormone produced are used in this work. The first, used by the selfish hormone, uses an exponential function and a varying value to generate hormone. In the case of the selfish hormone, the amount of energy in the battery was used as the input for an exponential function. Thus the linear change in the amount of energy in the battery resulted in an exponential change in the amount of selfish hormone produced. This is desirable when a linear change in some input should result in an exponential change in hormone level. For the selfish hormone, once the battery terminal voltage is near or above its fully charged voltage there is clearly an excess of energy and a more extreme behaviour to expend the energy is acceptable.

The second method uses a fixed increase in hormone. This approach is used by the anger, light, light detection and wind hormones. Some trigger causes the production of a fixed amount of hormone. The anger hormone, for example, produces 3 units of hormone in response to a failed transmission attempt. This approach is also used by the light hormone. When the rapid light change detection system is triggered a very large amount of light hormone is produced and transmitted throughout the network.

The quantity and rate of centre hormone produced is not determined by any system, rather it chosen by the user. For purposes of testing, the only quantities used are the maximum or minimum amount.

4.7.8 Hormone Decay

There are two methods used for decaying the hormones used in experiments. The first method involves the decay of a fixed quantity of hormone. The second an inverse exponential decay in hormone quantity. This is shown in Equations 4.14 and 4.15 where $Hormone(t)$ is the amount of hormone at time t and c is the decay constant and satisfies $0 < c < 1$.

$$Hormone(t+1) = Hormone(t) - c \tag{4.14}$$

$$Hormone(t+1) = Hormone(t) * c \tag{4.15}$$

All hormones decay to a value of 0, representing that none of that hormone was present.

The selfish, anger and light hormones all decay by a fixed amount. The selfish hormone decays by one unit every millisecond and the anger by one unit at every successful transmission attempt. Finally the light hormone decay by 0.04 units per duty cycle period (500 ms).

The centre and wind hormone decay at an inverse exponential rate which is achieved by multiplying each hormone by a value slightly less than 1, as shown in Equation 4.15. The closer the value is to 1, the longer the hormone takes to decay.

The decay values are chosen so that the maximum hormone level decays in the desired amount of time. For example if the hormone should decay linearly, has a maximum value of 100 and should decay fully in 100 seconds; a decay rate of 1 per second would be used.

The choice between a fixed unit of decay and inverse exponential drop is entirely dependant on the type of transition desired from the target system. The specific decay rates chosen for each hormone are arbitrary. However, they are chosen so as to produce the desired effects in the systems being controlled for the desired amount of time.

Hormone decay is needed as a hormone is intended to produce a behaviour, or have an effect, for a specified amount of time. The amount of time is determined by the designer of the system and so a decay rate and type must be chosen such that the effect of the hormone fits this requirement. Often the effect of a hormone on the control system is quite extreme and the hormone decay helps ensure the behaviour of the system tends back to a stable state maintaining homeostasis. For example, the centre hormone can increase the sensing rate of a node to as much as 11 times as often, which is unlikely to be sustainable, from a power perspective, or useful, from a data quality perspective. As such the hormone decays so that this behaviour also gradually decays back to the default level.

This mimics the behaviour of hormones in the human body, where hormones have a half-life due to being metabolised or binding to target receptors.

4.7.9 Hormone Combination

When multiple hormones are added to the node control system it is necessary to combine their effects. In particular the effect of hormones that affect the same part of the control system need to be combined. There are two aspects of the control system affected by hormones, the amount of time spent 'awake' in the duty cycle and the sensing rate of the node.

The simplest case of hormone combination is when a node combines hormone of the same type. This quantity of hormone is added to the nodes existing quantity of that hormone type. So a node with centre hormone level of 10 that receives another 5 units of centre hormone simply adds the two, resulting in 15 centre hormone. If the resulting quantity is above the maximum for that hormone then the extra hormone is discarded. T

The selfish and anger hormones both impact the proportion of a node's duty cycle spent awake and consuming energy. As there is no reason for the hormones to be considered unequally, each hormone is able to affect an equal portion of the duty cycle (50%). This has the effect of allowing each hormone to equally impact the energy consumption while not allowing either hormone to independently force a node to be constantly on or off.

Both the Selfish and Anger hormones are capped at a maximum quantity of 250. Thus the duty cycle of a node and therefore the combination of the selfish and anger hormones is defined as follows:

$$DutyCycleOn = SelfishHormone + AngerHormone \tag{4.16}$$

$$DutyCycleOff = 500ms - DutyCycleOn \tag{4.17}$$

Where 500ms is the duty cycle period.

The light, centre and wind hormones all affect the sensing rate of a node. To explain how these hormones were combined the operation of the sensing event timer must be explained. The duty cycling mechanism is used to count time to the next sensing event, with a counter being incremented in each part of the duty cycle. The duty cycle has a period of 500 ms with an off and on portion, resulting in 4 'clock ticks' per second. 240 of these ticks represent a minute, after which a sensing event is triggered.

The light hormone is integrated into this mechanism by adding the light hormone to the clock increment amount, making it count faster, increasing the sensing rate. The centre hormone is checked to see if it is promoting (positive) or suppressing (negative) the sensing rate. If it is promoting the sensing rate, then the centre hormone is added to the clock increment along with any light hormone. The clock increment was limited, so that regardless of how much light or centre hormone there is, the sensing rate can not exceed one event per 5.5 seconds. If the centre hormone is suppressing the sensing rate, then the threshold for the sensing event counter is multiplied by the centre hormone. This value is also limited, so that the maximum time between sensing events is 12 minutes. As a result, if the node is fully promoted by the light hormone and fully suppressed by the centre hormone, the sensing rate remains at the default of once per 60 seconds. The wind hormone is then combined by adding the wind and centre hormones together before performing the above calculation. This allows the light, wind and centre hormones to be combined equally while keeping the sensing rate within the desired bounds of every 5.5 seconds to every 12 minutes.

This combination method is formalised below. First the wind hormone, H_w, and the centre hormone, H_c, are limited (Equations 4.18 and 4.19) to between -11 and 11 ensuring that they are considered equally to the light hormone, H_l.

$$H_{wl} = min(11, max(-11, H_w)) \tag{4.18}$$

$$H_{cl} = min(11, max(-11, H_c)) \tag{4.19}$$

The light hormone, limited centre and limited wind hormones are then added and the result limited again to between -11 and 11, shown in Equation 4.19, to produce H_{Total} the total hormone quantity. Thus, all the hormones are combined but their effect on the sensing rate limited to between 11 times slower and 11 times faster than the default.

$$\begin{aligned} H_{Total} &= H_l + H_{wl} + H_{cl} \\ H_{Total} &= min(11, max(-11, H_{Total})) \end{aligned} \tag{4.20}$$

The amount of time between sensing events, or 1 over sensing rate SR can then be calculated as shown in Equation 4.21.

$$\frac{1}{SR} = \begin{cases} \frac{60}{H_{Total}}, & \text{if } H_{Total} > 0 \\ 60 * H_{Total}, & \text{if } H_{Total} < 0 \end{cases} \tag{4.21}$$

The time between sensing events is the default, 60 seconds, divided (if positive) or multiplied (if negative) by the total hormone quantity.

In experiments in which not all hormones are present, the above combination methods are still used but the level of the unused hormones are assumed to be 0. As a result, they do not impact the control system but the combination technique remains constant for all experiments.

4.7.10 Hormone Transmission

In the human endocrine system there is never any guarantee that a hormone, once produced, will be received by the intended cell. The hormone systems used in this work operates in the same manner. Hormones that are used internally, such as the selfish hormone, to a node are always received by the node itself. Adding the ability for a node to not receive hormones it itself produces is not considered beneficial as there is no reason for them not to be received. However, any hormones that are transmitted to another node have no guarantee that they are received. It would be possible to track whether hormones included in unicast packets, such as the anger hormone, have been received due to the use of acknowledgement packets. This is not done as the human endocrine system does not have an analogous hormone delivery system. Once a hormone is transmitted it is considered gone, if the transmission fails that hormone is ‘lost’. For hormones that are transmitted via a broadcast mechanism there is not even an acknowledgement packet that can be used to know whether the hormone had been received.

The only check made on hormones received by a node is to check the packet ID to ensure that the packet has not already been received. This stops the same hormone packets from being processed multiple times when they should not have been.

4.8 Summary

The hardware choices and decisions were presented. This included the electronics, enclosure and node mounting system. The power duty cycle approach that the control system is based on order to allow control over power consumption was shown and the impact on the packet transmission and routing systems discussed. Unicast transmissions are retried 10 times, every 50 ms, in an attempt to ensure reception. The routing system provides the ability to route packets via alternate routes towards the base station, although the result is not a full mesh network. The topology of the network used in experiments is constant so as to

Table 4.4: Summary of the limits, decay rate and production rate of the hormones used in this work.

Hormone	Limits	Decay Rate	Production Rate
Selfish	$0 \leq H \leq 250$	Linear, -1/ms	Exp. see Section 4.7.2
Anger	$0 \leq H \leq 250$	Linear, -1/tx attempt	+3/tx fail
Light	$0 \leq H \leq 11$	Linear, -0.08/second	+11/light change detected
Light Detection	$0 \leq H \leq 90$	Linear, -30/5.5secs	+60/5.5secs see Equation 4.12
Centre	$-100 \leq H \leq 100$	Inv Exp, 0.999975/250ms	user selectable
Wind	$-100 \leq H \leq 100$	Inv Exp,0.997/250ms	+1, if wind speed >5kts -1,if wind dir stdev <5

maintain similar conditions between experiments. A short discussion on post calibration of the meteorology data provides reassurance that the data produced by the network is valid and therefore useful.

The concept of endocrine inspired systems in the context of this work was explained and detailed. The hormones that are used, how they work and the effect that they have on the node control system was presented. Table 4.4 shows a summary of the limits, production rate and decay rate of each hormone.

Many of the choices made in this chapter are made as a result of testing and the trial deployments which were detailed in the previous chapter, Chapter 3. The next two chapters detail the experiments conducted, metrics used for analysis and the experimental results.

Chapter 5

Power Usage

Using the methodology described in the previous chapter a series of experiments were run to study the first two research questions presented in Chapter 1, Section 1.1. Namely, whether endocrine inspired control methodologies could be used to adapt power consumption of nodes in a wireless sensor network and whether the network lifetime could be increased. Six experiments were conducted using two network layouts, a small 5 node network and a larger 20 node network. There are several key areas that it is felt, could be improved through the use of these techniques.

- The life time of individual nodes and the network as a whole could be improved, ideally to indefinite operation.
- Power could be better utilised, using more when it is available and less when it is scarce.
- The health of the batteries could also be considered.

To do this the amount of power a node is capable of using must be variable. There must be some mechanism that can adjust the amount of power used and some activity that can be better performed through the expenditure of more energy. In some sensor network scenarios modulating the amount of power available for the sensor payload would be the first choice. However, the meteorological sensors installed on the nodes used in this work consume very little power and increased measurement or processing would be insufficient to utilise excess power. Instead, the transmission and routing system is modulated to use excess energy when it is available. By increasing and decreasing a node's duty cycle, as described in Section 4.2, it can be made to consume more or less power.

The Selfish hormone and Anger hormone were enabled and compared against a control experiment for both network types. The Selfish hormone is produced at a rate governed by the battery level to allow each node to "selfishly" extend its lifetime at the expense of network routing performance. The Anger hormone, attempts to reduce network routing problems by generating hormone in response to failed transmissions. Both of these hormones affect the portion of a node's duty cycle that is spent awake listening for packets and, therefore, its routing behaviour.

The real world nature of the data collected during the experiments presents a challenge when it comes to analysis. Calibration of the battery voltages with respect to the temperature must be performed if it is to be used as a metric. The calibrated battery voltage enables the calculation of the change in voltage from start of a day to the end of a day to be performed on a node by node basis. This provides an estimate of amount of power consumed by a node. Due to the expectation of a large element of noise in the data, a simple model is generated to explore the expected outcome. Filtering of the data was carried out to eliminate data that was too incomplete to facilitate proper analysis.

5.1 Uncontrolled variables

While significant effort was expended in an effort to keep experimental conditions constant, there were some variables between experiments that were unable to be controlled. One of the most significant of these was the weather. This, quite obviously, cannot be controlled between experiments and there were several possible effects that differing weather between experiments could have. Humidity levels can affect the signal quality between nodes, making it easier/harder for messages to be received between nodes. Fluctuations in temperature can result in changes in oscillator frequency which in turn may alter the rate at which messages are sent, various timing systems and sleep times.

Long term weather predictions are not sufficiently accurate to be able to plan experiments to take place at certain times, so as to minimise difference in weather conditions. There are also inescapable changes in weather due to the time of year. The length of an experiment, several weeks, dictated that the last experiment run would have to be run a significant amount of time after the first. Even if a whole year were left between experiments so that the next experiment could be run at the same time of year, it would require significant luck to encounter the exact same weather conditions. In any event, testing in this manner was not feasible given the time frame and number of experiments to be performed. Using a separate set of sensors to try to establish a ground truth would have been an option. However, as the location of a node plays a significant role in the environmental conditions it experiences a reliable and accurate ground truth would be very difficult to obtain.

The second uncontrolled variable was linked to the weather, but could be considered a separate issue. The available light, which could change due to either the weather, length of a day or obstruction of the solar panel, directly impacted the energy input to the system.

The primary method for managing the effect of variable day length and available light was to run the control experiment and hormone experiments for the five node network and 20 node networks in a different order. The control for the five node experiment was run during a time of year with longer days and more light than the hormone enabled experiment and vice versa for the 20 node experiments. Similarities between control and hormone experiments should, therefore, not be due to the higher availability of light (and therefore energy) in one experiment with respect to another. Lastly the hardware, electronics and batteries, experienced some level of degradation as they were exposed to a harsh environment. The circuitry, whilst as protected as possible, experienced high levels of humidity, variations in temperature and ingress of particulates and wildlife. The batteries, in addition to the aforementioned factors, also experienced a variety of non ideal charging situations such as overcharging, undercharging and repeated and frequent charge/discharge cycles. Batteries were recharged between experiments and any that had very low terminal voltages or appeared damaged in some way were swapped with pre charged batteries. Hardware was inspected at the beginning of each experiment, any detritus removed and if necessary replacements made. Attempting to quantify the effect of these variables would be extremely difficult and hard to justify.

5.2 Experiments

In total 6 experiments focussing on power consumption and utilisation were run. Two of these were control experiments and four used endocrine inspired mechanisms to attempt to make better use of available power. The first two experiments were run on a small

Table 5.1: Summary of the six experiments run to determine whether endocrine inspired control methodologies can be used to adapt power usage in a wireless sensor network. E1 and E3 are control experiments as they have no hormones enabled. E6 has the same hormones enabled as E5 but the sensing rate is greatly increased.

Experiment Number	Selfish Hormone	Anger Hormone	Deployment Type	Notes
E1	No	No	5 Node	Control for 5 node
E2	Yes	No	5 Node	
E3	No	No	20 Node	Control for 20 node
E4	Yes	No	20 Node	
E5	Yes	Yes	20 Node	
E6	Yes	Yes	20 Node	Increased sensing rate (every 5.5 seconds)

network of five sensor nodes and a base station, arranged in a star network. Nodes were not able to route messages from any other node, there were no mechanisms for the network to "self heal" and any adaptivity was therefore restricted to each individual node. A control experiment and a "Selfish Hormone" experiment were run in this set up. The latter four experiments used 20 sensor nodes and a base station. In this configuration certain nodes were configured to forward messages towards the base station as described in the Methodology chapter. Some nodes had alternative routes so as to be able to adapt to failures, others did not. This routing configuration remained constant for each experiment so as to provide as similar a test environment as possible. The experiments are summarized in Table 5.1.

Experiment E1 provides the baseline for experiment E2 as it implements the control system described in Section 4.2 of the Methodology chapter. Experiment E2 then adds the selfish hormone to the control system.

Experiments E3 to E6 all use a network consisting of 20 nodes. E3 is the control, implementing the same control system as E1 with a larger number of nodes. The selfish hormone is then enabled in E4 and the anger hormone in E5 and E6. In experiment E6 the sensing rate of the network is increased from one sensing event every 60 seconds to one every 5.5 seconds.

5.2.1 Control Experiments

Both control experiments used the same principles. The only difference was that nodes in the 20 node control experiment were able to route messages along pre-defined routes. This allowed the network to have some flexibility and the ability to cope with the loss of a node, whilst maintaining the same experimental set up between experiments.

The control experiments used the fixed 20% duty cycle control system discussed in Section 4.2 of the Methodology chapter. The duty cycle period was 500 ms and each node was tasked to read its sensors and transmit that data once every 60 seconds.

The selfish hormone was added to this control system in Experiments E2, E4,E5 and E6. The anger hormone added in Experiments E5 and E6.

5.2.1.1 5 Node Power Consumption

The nodes used in experiments E1 and E2 had a slightly higher power consumption than the nodes used for experiments E3 to E6. The difference was approximately 8 mA of constant current draw which was due to two LEDS on the microcontroller board and not sleeping the microcontroller. When the 20 node experiments were run, one LED was removed from the board and another disabled in software. The 5 node control experiment also had time synchronisation accidentally enabled. This resulted in one extra transmission per duty cycle period. The current consumption of the node hardware in the 5 node experiments was 12 mA while sleeping and 72 mA while awake. With a 20% duty cycle this equates to $(0.8 \times 12) + (0.2 \times 72) = 24$ mA. The extra packet contained around 20 bytes and at 250 kpbs took $\frac{1}{250000} \times 20 \text{ bytes } \times 8 \text{ bits} = 0.00064$ seconds to transmit. The XBee datasheet states the current consumption will increase by 125 mA during transmission. The constant current draw then becomes $24 + (0.00064 \times 125) = 24.08$ mA which, assuming no solar input at all, results in the expected node lifetime decreasing from 291.7 hours to 290.7 hours. Given that this is a worst case calculation as the transmission current consumption figure is based on the peak consumption the effect is considered to be negligible.

5.2.2 Selfish and Anger Hormone Experiments

Experiments E2 and E4 both enable the selfish hormone. In experiments E5 and E6 the anger hormone is enabled in addition to the selfish hormone. E6 has an increased sensing rate so as to better study the effects of the anger hormone. The anger hormone proliferates more quickly due to the increased sensing rate. The results of these experiments are used to analyse whether endocrine inspired control systems can adapt power usage in a wireless sensor network. The available data on power consumption and generation is the light level each node is exposed to and its battery voltage. The light level can be used to estimate the amount of energy generated by the node and the change in battery voltage to estimate the energy consumed by the node. This leads to the hypothesis $H1$:

> $H1$: There is a positive correlation between integrated light and change in battery voltage.

Without a correlation between the estimates of energy input and output from a node, further analysis on whether power usage is being adapted is impossible.

If we can accept this hypothesis then we can test hypothesis $H2$:

> $H2$: The selfish hormone approach decreases the influence of available energy on the battery voltage.

The metric for whether power consumption has been improved is the model described in Section 5.5. Accepting this hypothesis provides strong evidence to support the main hypothesis stated in Chapter 1; that endocrine inspired control methodologies can adapt power usage and increase the quality of data from a wireless sensor network in response to environmental factors.

Hypothesis $H1$ is applied to all six experiments and $H2$ to experiments E2 and E4 to E6 using E1 and E3 as baselines for comparison.

5.3 Temperature Calibration for Battery Data

Like most types of battery chemistry, lead acid batteries are subject to changes in battery terminal voltage in relation to temperature. To calibrate for this, an experiment was run using 10 of the same lead acid batteries used for the sensor nodes. Each battery was connected to a load that simulated the power consumption of the 20% duty cycle used in the control experiments. The voltage and temperature of all the batteries was recorded before placing five in a refrigerator and the other five outside of the refrigerator. The temperature and voltage of all 10 batteries were periodically measured for a period of six hours, before leaving the batteries to continue to discharge overnight. The next morning, the batteries placed inside the refrigerator were removed and left to warm back up to room temperature while continuing to record their voltage and temperature.

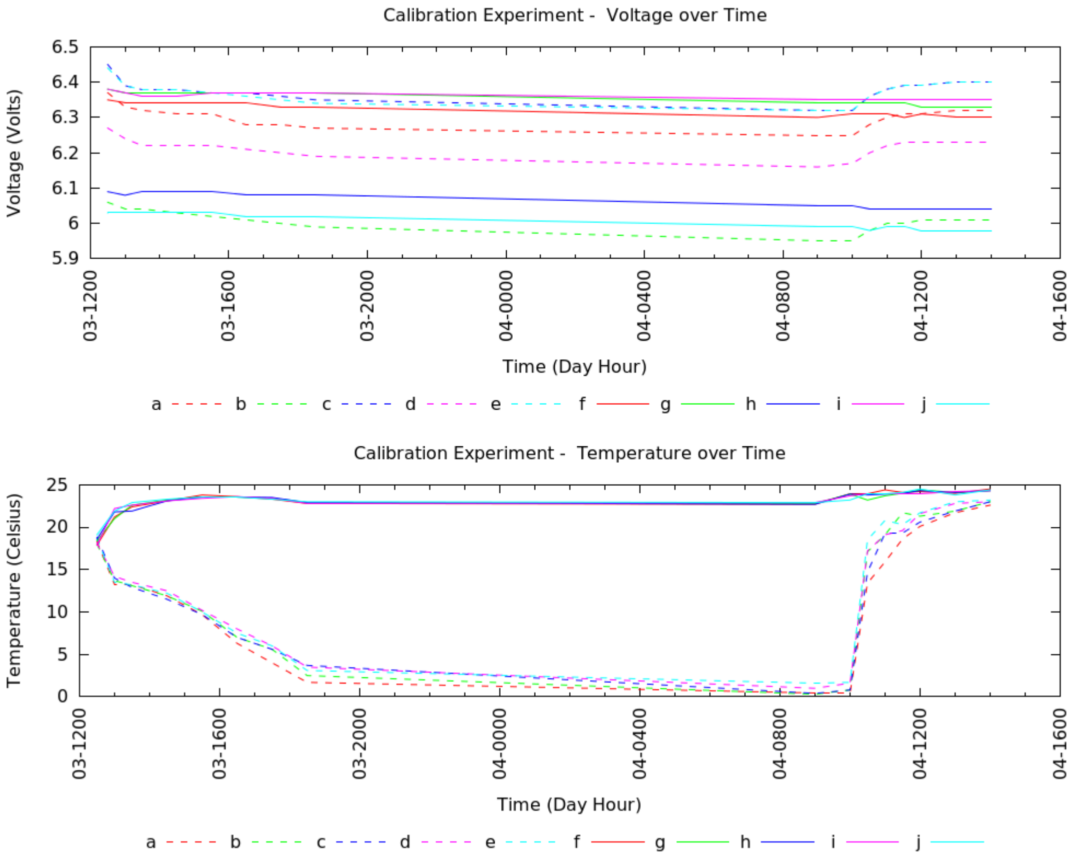

Figure 5.1: Graphs showing the voltage and temperature changes experienced by the batteries in the calibration experiment. Batteries a,b,c,d and e were cooled in a refrigerator whilst f,g,h,i and j were left at room temperature.

Although the batteries start from different initial terminal voltages, it can be seen from Figure 5.1 that they experience similar changes in both voltage and temperature. As the temperature of a battery drops its terminal voltage under the simulated load also drops. Batteries that were chilled in the refrigerator experienced a voltage drop of around

0.11 V from the start of the experiment to their removal from the refrigerator. The non chilled batteries experienced a voltage drop of around 0.04 V in the same time. Once the batteries were removed from the refrigerator, their terminal voltages increased as they warmed, despite the constant power consumed by the dummy load.

Table 5.2: Temperature and voltage changes for batteries a,b,c,d and e from room temperature to removal from the refrigerator.

Battery	Range (T)	Range (V)	Temperature Change	Voltage Change
a	18.3-0.05	6.37-6.25	-17.8	-0.12
b	18.0-0.08	6.06-5.95	-17.2	-0.09
c	18.8-0.80	6.45-6.32	-18.0	-0.13
d	18.1-1.60	6.27-6.17	-16.5	-0.10
e	18.6-1.70	6.44-6.32	-16.9	-0.12
		Average	-17.28	-0.112

Table 5.3: Temperature and voltage changes for batteries a,b,c,d and e from removal from the refrigerator to room temperature.

Battery	Range (T)	Range (V)	Temperature Change	Voltage Change
a	22.6-0.5	6.32-6.25	22.1	0.07
b	23.0-0.8	6.01-5.95	22.2	0.06
c	23.0-0.8	6.40-6.32	22.2	0.08
d	23.0-1.6	6.23-6.17	21.4	0.06
e	23.2-1.7	6.40-6.32	21.3	0.08
		Average	21.84	0.07

To calculate a calibration value of voltage per degree the experiment was split into two sections. The first was the gradual decrease in temperature (Table 5.2) and the second the gradual increase in temperature (Table 5.3). The average of all five batteries range in voltage change and range in temperature change were calculated. Finally the average voltage change was divided by the average temperature change to give the calibration

Table 5.4: Temperature and voltage changes for batteries f,g,h,i and j left at room temperature.

Battery	Range (T)	Range (V)	Temperature Change	Voltage Change
f	24.5-17.9	6.35-6.30	6.6	-0.05
g	24.3-18.4	6.38-6.33	5.9	-0.05
h	24.3-18.6	6.09-6.04	5.7	-0.05
i	24.3-17.9	6.38-6.35	6.4	-0.03
j	24.4-19.1	6.03-5.98	5.3	-0.05
		Average	5.98	-0.046

value. The temperature and voltage changes for the control group of batteries are shown in Table 5.4.

$$\text{Average Temperature Change} = \frac{17.28 + 21.84}{2} = 19.56$$
$$\text{Average Voltage Change} = \frac{0.112 + 0.07}{2} = 0.091$$
$$\text{Voltage change per degree} = \frac{0.091}{19.56} = 0.004652352$$

For every 1 degree of temperature change there was, on average, a voltage change of 0.0046523517 volts. Looking at temperature data gathered from the control, selfish hormone and anger hormone experiments a set point 10 °C was chosen. This represented the mid point of range of temperatures experienced by nodes in these experiments. Values below 10 °C were increased by the calibration value and above 10 °C were decreased by the calibration value.

Data was post calibrated using the calibration value derived above. The calibrated battery voltage was calculated to 2 decimal places.

5.4 Filtering

Due to the nature of the communications mechanism, there is no guarantee that a packet that is transmitted will be received and stored at the base station. Node transmission periods drifting in and out of phase with each other, high contention rates for airtime, marginal network connections and even changing weather conditions can affect the number of packets received from any particular node. While the theoretical gap between packets should be 60 seconds, in reality packets are lost en route to the base station. As there are no attempts made to detect lost packets or retransmit packets this results in gaps of varying sizes in the data from each node. During these gaps the state and conditions of a node is not known. As the gap grows larger is becomes difficult to justify any sort of estimation of what occurred in the gap. The analyses of power usage and consumption presented in Section 5.5 requires an estimate of the energy both entering and leaving a node. To ensure these estimates are realistic the filtering techniques described below are used.

The filtering systems consider each day for each node in an experiment separately. If any of the conditions discussed below are found then that particular day for that node is marked as invalid. Any days marked invalid are not included in the output from the filtering program and are therefore not used for the data analysis.

5.4.1 Voltage Change Filtering

The energy expended by a node can be estimated by the difference in its battery voltage from the start of a day compared to the end of the day. To be able to calculate the change in voltage from the start to the end of the day, a sensor node's battery voltage must be sampled from the data at a specific time. Midnight was chosen as there would have been no sunlight on the solar panel for a number of hours and the battery voltage would be unaffected by the solar panel, increasing accuracy. It is, therefore, necessary to sample the battery voltage as close to midnight as possible to minimise the error incurred by the battery discharging. A maximum time of 30 minutes after midnight was considered to be the limit for a battery measurement. During this time it was very unlikely for the battery to have decreased by 0.01 V, the maximum resolution for the battery voltage measurement.If there are voltage measurements from a particular node, within 30 minutes of midnight both at the start of the day and the start of the next day, then that day is considered to be valid. If either the start or end of day battery voltage measurement is more than 30 minutes after midnight the day is marked as invalid due to the imprecise nature of the battery voltage measurement.

5.4.2 Valid Voltage Filtering

The aforementioned filtering method removes a days data if the battery voltage, used to calculate the change in battery voltage over a day, is potentially inaccurate. It is, however, necessary to consider cases where the battery voltage is considered to be near or on the boundary of being fully charged or discharged. The valid range of voltages for t/he chosen lead acid battery is 5.5 V to 6.5 V. The voltage of a battery, when outside this range is non linear and does not relate well to the energy stored.

Days where a node's battery voltage started and ended above the fully charged value of 6.5 V are marked as invalid. Regardless of the input from the solar panel, a fully charged battery can not store more energy. Days that start or end on a battery voltage below 5.5 V are also marked as invalid due to the non-linear behaviour of battery voltage below 5.5 V.

5.4.3 Inter-packet Gap Filtering

Light was chosen to be the metric for the quantity of energy received by each node. For each node, every day's light measurements were integrated over time to result in a value representing the light input for that node on that day. While this method could tolerate gaps in the light data, too large a gap caused the value calculated to be significantly different from the actual value. A maximum time gap between measurements was necessary to eliminate invalid data. Two possible maximum time between packet parameters were chosen, 10 and 30 minutes. While the light value can change relatively quickly, due to obstruction by an object or clouds, it generally changed slowly over the course of the day. If the time between packets from a node over the course of a day exceeded either 10 or 30 minutes, the day was marked as invalid. Two files were output from the filtering process one using the 10 minute parameter and the other using the 30 minute parameter. To further refine this process only gaps that fell during "daylight" hours were considered. This was due to there only being light data during these times. Excluding a day's worth of data due to a 1 hour gap at 2 am would be nonsensical as,regardless of the gap, there was no light to be missed. Official sunrise and sunset times were used and the maximum range for the time period of each experiment was chosen so as to ensure no light data was filtered out.

Table 5.5: The number of "valid" days after applying three filtering options on the data from each experiment. Also shown is the percentage of data that is usable after applying the chosen filtering, 30/30v. The structure of the filter name is 'maximum gap between packets in minutes'/'maximum time from midnight for voltage reading in minutes' where the 'v' indicates that 5.5 to 6.5 voltage range filtering was also applied.

Experiment	Duration	TDS	30/30	30/30 v	10/30 v	%
E1	19 days	85	59	59	52	68.2%
E2	45 days	215	178	177	177	80%
E3	19 days	340	151	45	39	13.2%
E4	29 days	540	475	439	415	81.3%
E5	18 days	320	236	173	160	52.2%
E6	44 days	840	571	392	352	45.6%
E7	6 days	80	63	42	41	52.5%
E8	13 days	220	107	105	85	47.7%
E9	24 days	440	174	160	7	36.4%

5.4.4 Filtering Results

Table 5.5 shows the result of filtering on the data from each experiment. The theoretical data size, TDS, was calculated as the number of days the experiment ran minus 2, multiplied by the number of nodes. The subtraction of two days was due to the fact that the first and last days of an experiment have an incomplete set of data. The amount of data for the three chosen filtering options shown in the columns 30/30, 30/30v and 30/10v. The first value represents the maximum acceptable inter-packet interval in minutes. The second value is the maximum time after midnight for a voltage measurement in minutes. The "v" indicates that the 5.5 to 6.5 voltage range filtering was also applied. For the chosen filtering, 30/30v the percentage of the theoretical data that is achieved by the filter is shown.

5.4.4.1 Light Integration Method

The method of integrating light over the course of a day, on a node by node basis, needed to be robust. There were several constraints applied when calculating the integrated light value. Light was integrated between the day start hour and the day end hour which were constant throughout the experiment. As previously mentioned in the filtering discussion, these values were obtained from astronomical data of the deployment time and location. For example the light integration would be calculated from 6 am up to, and including, 8 pm. Secondly, the initial and final light levels of the defined period are considered to be 0. The assumption is made that each measurement represents the light level for a duration of time. The duration is defined as from the midpoint between the previous light value and current light value to the midpoint between the current light value and the next light value. An visual example of the light integration technique used, can be seen in Figure 5.2. Of particular note are the cases of missing data, for example between the 5th and 6th point in Figure 5.2.

5.5 Naïve Model

In order to better understand and interpret the data from the real world experiments, a naïve model is created. The data available for analysing the power consumption and management of a node is the battery voltage and light level. The filtering discussed in Section 5.4 aims to remove the data for days in which there is insufficient data to be of use in quantifying the energy entering or leaving the node. To help understand the effect of different types of power management on the battery voltage and light data a model is needed. A system that does not manage its power, a system that manages its power and a perfect system are all modelled. The change in battery voltage and associated energy input, which the light level is a surrogate for, can be plotted. The difference between the three systems can be analysed to determine how to interpret the data from the experiments.

A node is modelled in terms of energy input and consumption on a day to day basis. A random distribution of energy input is fed into the model to provide an analogue for solar conditions in the real world.

The assumptions made for the development of the model are presented below:

- For any given day, the amount of energy entering the system is in the range defined as between 0 and the maximum energy possible from the solar panel. In the absence of a better model, the amount of energy entering the system each day, is assumed to be a random value within this range.

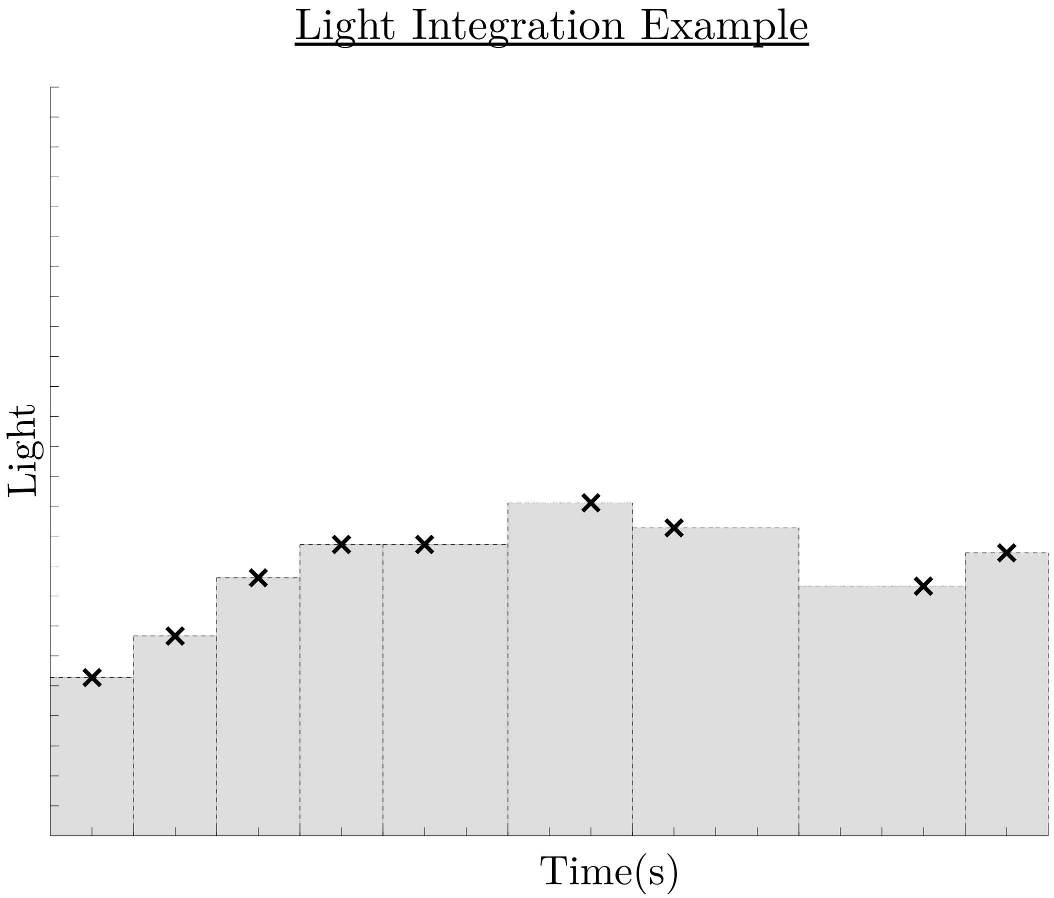

Figure 5.2: A visualisation of the light integration method used to estimate the amount of energy a node received in a day. Light is a unit-less raw sensor value from the light sensor. Each cross marks one received data point.

- Energy into and out of the system can be modelled as an increase or decrease in battery voltage which is linear between 5.5 V and 6.5 V.
- The available power for the next day can be calculated as being today's voltage plus the energy in, minus the energy out.
- It is possible to use the energy entering the system to determine how much energy to expend on any given day.
- All nodes experience the same energy conditions.

These calculations assume the energy entering the system, X is uniformly distributed within the range in_{min} (minimum energy input) to in_{max} (maximum energy input).

$$X \sim U[in_{min}, in_{max}]) \tag{5.1}$$

Where in_{min} is likely to be 0. In a system that manages its power consumption, energy expenditure will be higher if there is more energy available and lower when less energy is available. The system's energy consumption, c, is defined to be within the range $[c_{min}, c_{max}]$. We assume that nodes are able to adjust their power consumption to the maximum, c_{max}, when the maximum power is available and the minimum,c_{min}, when the minimum is available. The size of the energy consumption range dictates how effectively the system is able to use energy.

There are two possible extreme cases. The first, when the minima and maxima of the two ranges are equal. The second, when the minimum and maximum of the consumption ranges are the same.

$$[c_{min}, c_{max}] = [in_{min}, i_{max}] \tag{5.2}$$

and

$$c_{min} = c_{max} \tag{5.3}$$

The first case represents a system that can perfectly regulate its power consumption to match its energy input. The second case represents a system that cannot regulate its energy consumption at all.

To examine the behaviour of this model over time the following equation was used to calculate the amount of energy remaining at the end of each day:

$$c = \frac{(c_{max} - c_{min})(e - in_{min})}{in_{max} - in_{min}} + c_{min} \tag{5.4}$$

$$Energy_n = Energy_{n-1} + e - c \tag{5.5}$$

Where E_n is the remaining energy for a day, E_{n-1} is the remaining energy for the previous day, e is the energy gained for that day and c is the energy consumed during that day. This equation was used to simulate 100 days of operation with a variety of energy consumption ranges. Plots of energy input against change in voltage were made. Figure 5.3 shows the results obtained from running the model. Three sets of data are shown, the first for a simulated control experiment, where the energy consumption range is 0, i.e. power consumption is fixed. The second shows a system able to regulate its energy consumption but not perfectly. The final data set shows the results for a modelled perfect system, able to regulate its power perfectly based on the energy input to it. It can be observed that the steepness of the slope increases as the systems ability to adapt its power improves.

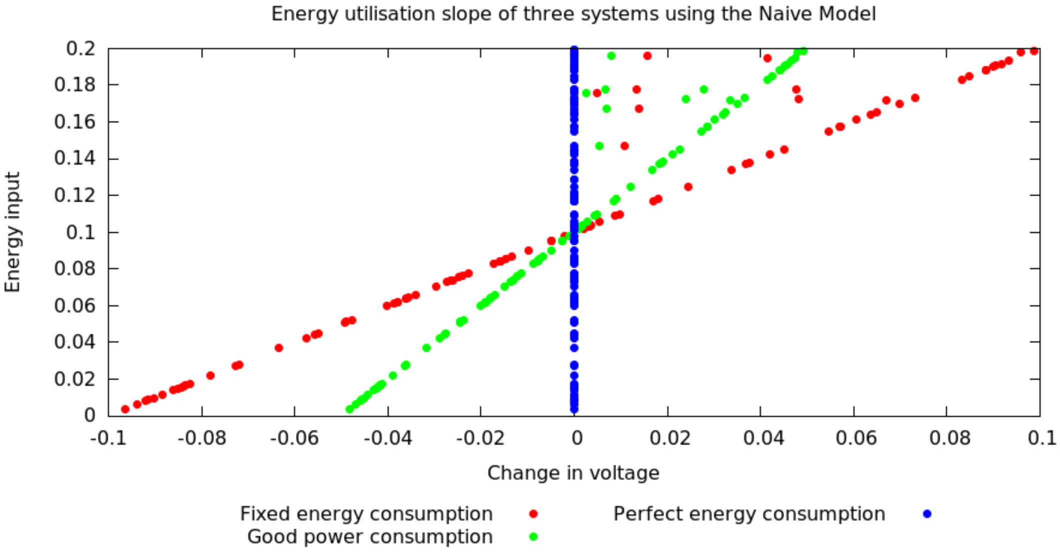

Figure 5.3: Energy utilisation plots for three systems using the Naive model described in Section 5.5.

We expect to see similar behaviour in the experiments performed. The real world measures of both energy input and energy remaining are likely to be noisy. In particular the energy input, which is based on light readings, is expected to have a large error with respect to the battery voltage change. The model also assumes that all of the nodes experience the same energy input, whilst in reality many will not due to their location, orientation and any obstructions of their solar panel. Independently monitoring was not feasible as the light level would need to be recorded at each node location with an entirely separate system.

Despite these factors, **we expect that an increase in the slope on a plot of energy input against change in energy would indicate that a system was able to better use the available energy. See** $H2$ **Section 5.2.2.**

5.6 Results

The integrated light value and the change in battery voltage for each node are used to determine how well the network as a whole uses the available power in accordance with the model described in the previous section. For each day of each experiment, each node's integrated light value is plotted against its change in battery voltage over that same day. Figures 5.4, 5.5, 5.6, 5.7, 5.8 and 5.9 show the integrated light plotted against the change in battery terminal voltage for experiments E1 to E6. The data is filtered in the manner described in Section 5.4. Days where a node's battery voltage started and finished the day above 6.5 V were removed during the filtering process. Thus, if integrated light was a good predictor of energy input, an increase in integrated light will correspond with an increase in battery voltage. Each data point represents one day for one node. Some experiments contain fewer data points than others. In particular experiment E4 (20 nodes, selfish hormone enabled), Figure 5.6. There are two reasons for the smaller quantity of data in this experiment. Firstly, the base station lost power for 2 days during the middle of the experiment. As a result, there is no data for those two days or the days either side of the missing days. This results in

$$4 \text{ days} \times 20 \text{ nodes} = 80 \text{ datapoints}$$

missing as a result. Secondly, the filtering process filtered out a large number of data points due to many nodes consistently having battery voltages above 6.5 volts. These days do not provide a reliable estimate of energy expended by nodes. Figures 5.4, 5.5, 5.6, 5.7, 5.8 and 5.9 suggest that there is a positive correlation between the integrated light and the change in voltage for a day. Plots generated for experiments run as part of the data focus chapter also suggest the same correlation, see Appendix A.

Hypothesis $H1$, as discussed in Section 5.2.2 is tested to determine whether there is a correlation between the energy input and output estimates. As there is no evidence that the data is normally distributed, this correlation can be tested using Spearman's Rank Correlation Coefficient, a non-parametric measurement which does not assume that the data is normally distributed. To make use of the model, discussed in the previous section, there must be evidence to support a correlation between the light level and change in battery voltage.

$H1_0$:There is no positive correlation between integrated light and change in battery voltage.

$H1_1$:There is a positive correlation between integrated light and change in battery voltage.

This forms a one tailed test, as we only test that an increase in voltage change correlates with the quantity of light increasing. Spearman's correlation coefficient was calculated in R using the *cor.test()* command. The results are shown in Table 5.6.

Table 5.6: Spearman's rank correlation coefficient for integrated light and change in voltage for all experiments. E1 is the 5 node control and E3 the 20 node control experiment.

Experiment	Spearmans Rho	N	Alpha	P Value	Reject/Accept H_0
E1	0.4652205	58	0.05	<0.001	Reject
E2	0.626908	172	0.05	<0.001	Reject
E3	0.7045392	45	0.05	<0.001	Reject
E4	0.6011654	439	0.05	<0.001	Reject
E5	0.4800635	383	0.05	<0.001	Reject
E6	0.7137981	167	0.05	<0.001	Reject

At a 95% significance level we can reject the null hypothesis, $H1_0$, and accept $H1_1$ for all six experiments (E1 to E6). The weakest correlation is that of experiment E1, the 5 node control experiment. This may be due to the small amount of data as the deployment lasted only a 2 weeks and used 5 nodes. All experiments show good correlations, with a lowest of 0.465, and p values such that we could accept $H1_1$ at a 99% significance level. As a result of accepting this hypothesis the model described in Section 5.5, can be used to interpret whether the power consumption was being adapted.

To attempt to determine the differences between experiments and try to quantify the difference between the control and endocrine approaches, a linear model is fitted to the data using linear regression. There is no strong justification for fitting a more complex model to the data and the following assumptions are made:

- Over the period of a day, the amount of energy gained from the solar panels is linearly related to the integrated light.
- The battery voltage is linear over the range 5.5 V to 6.5 V.

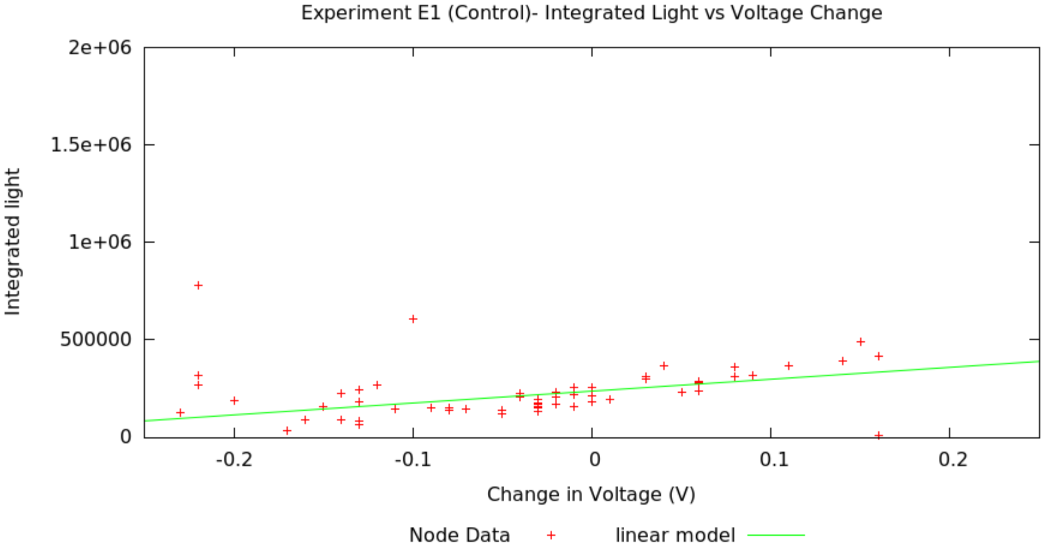

Figure 5.4: Integrated Light over Voltage Change for the Five Node Control Experiment.

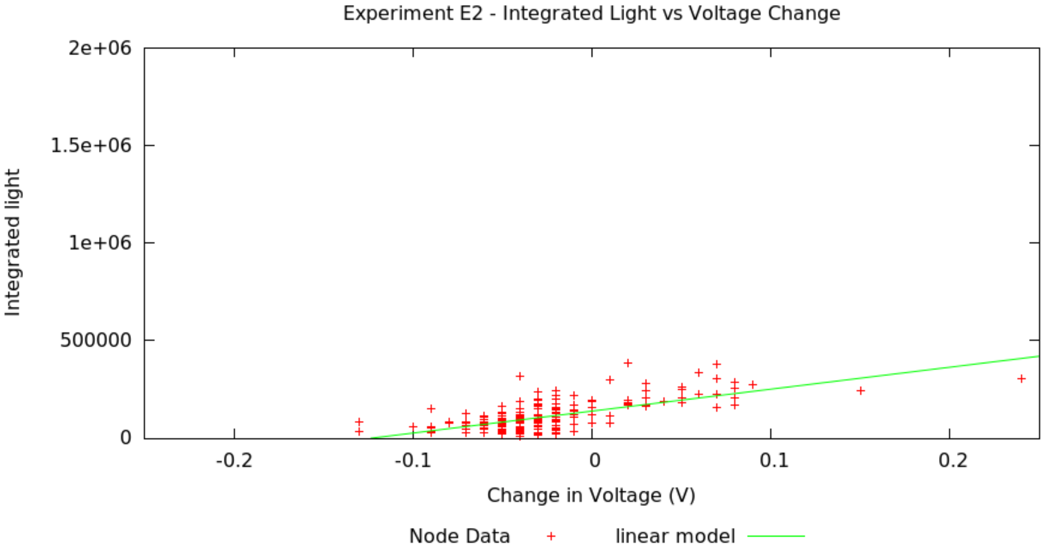

Figure 5.5: Integrated Light over Voltage Change for the Five Node Selfish Hormone Experiment.

- There is a linear relationship between the amount of energy gained and the change in battery voltage.

These are the same assumptions made for the predictive model described in Section 5.5. There are likely to be outliers due to the integrated light not being a direct measure of energy input from the solar panel. It is possible to obscure the light sensor without

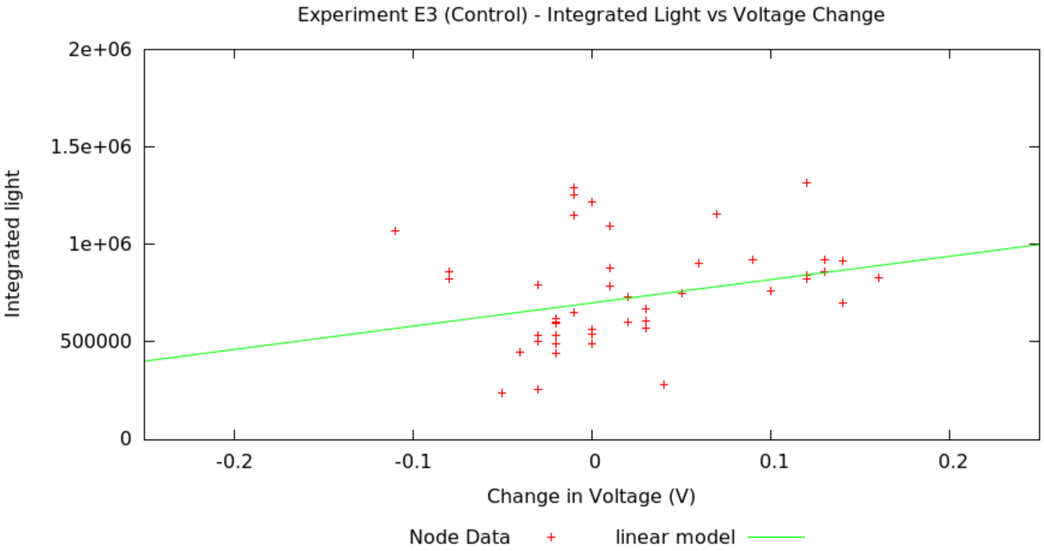

Figure 5.6: Integrated Light over Voltage Change for the Mesh Control Experiment.

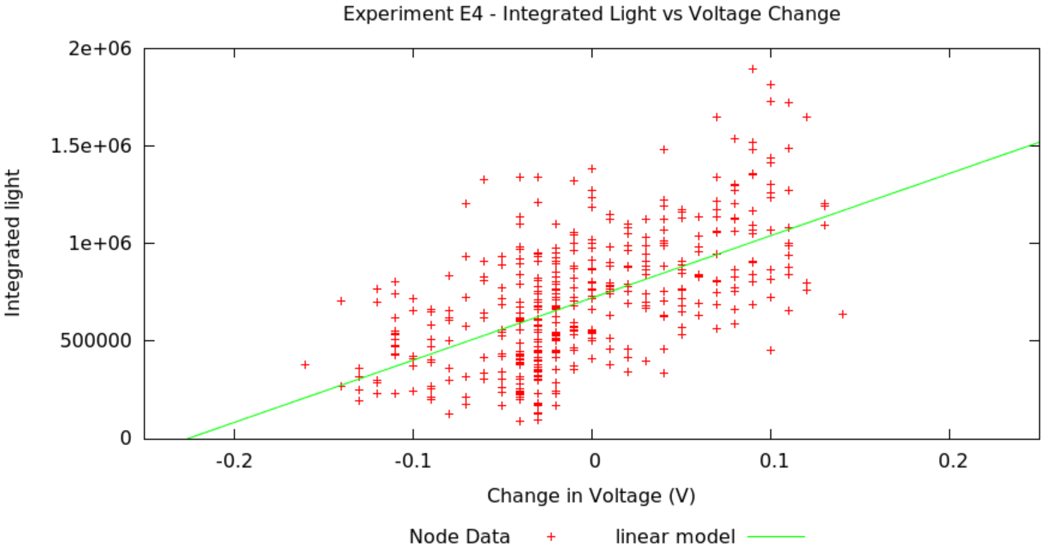

Figure 5.7: Integrated Light over Voltage Change for the Mesh Selfish Hormone Experiment.

covering the solar panel or vice versa and therefore some error is expected. To cope with these outliers, R's robust linear model function, *rlm()*[26], is used to generate the linear model. The results of the linear regression can be seen in Table 5.7.

For a similar amount of energy input, from the solar panels, the predictive model shows that the steepness of the line of best fit represents how well the node adapted its power

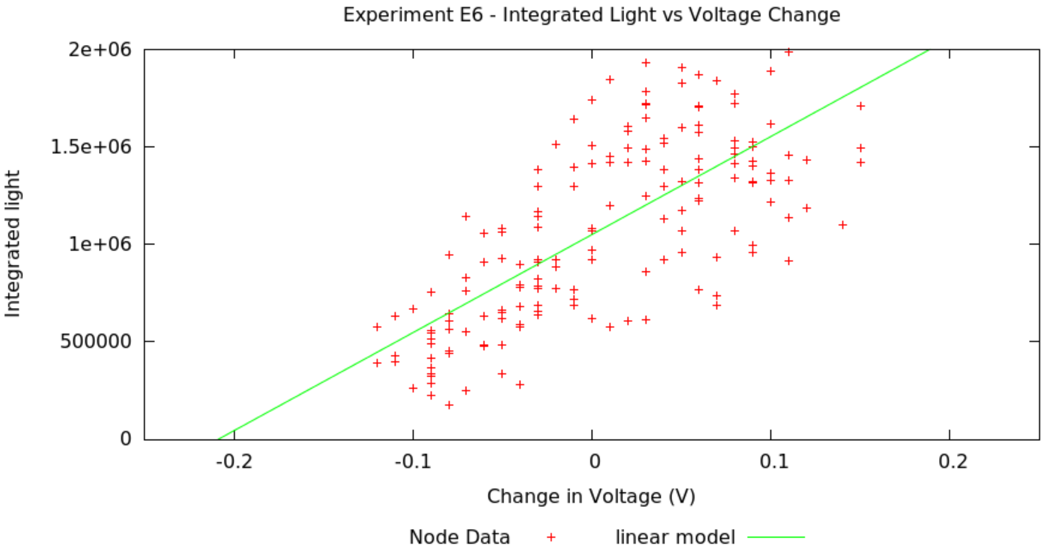

Figure 5.8: Integrated Light over Voltage Change for the Mesh Anger Hormone Experiment with a fast transmission rate.

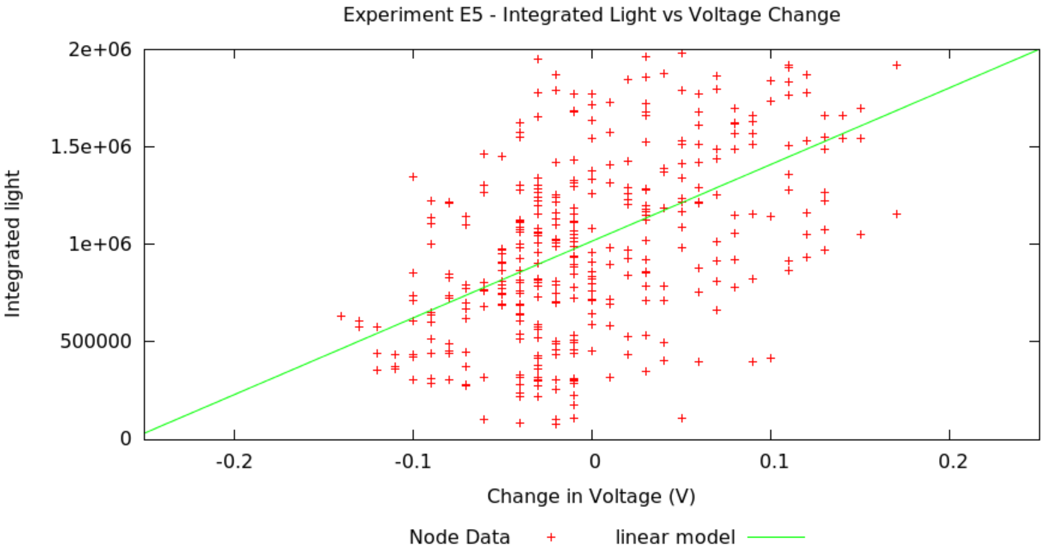

Figure 5.9: Integrated Light over Voltage Change for the Mesh Anger Hormone Experiment with the normal transmission rate.

consumption to the available energy. A more horizontal line of best fit indicates that the battery voltage varied regardless of the energy input. A vertical line of best fit indicates that regardless of energy input the battery voltage never varies as the energy was being

Table 5.7: Slopes of the line fits and average absolute error for all experiments. E1 is the 5 node control and E3 the 20 node control experiment.

Experiment	Slope ($\times 10^5$)	Average Absolute Error of Residuals ($\times 10^5$)
E1	6.10	0.76
E2	11.25	0.44
E3	11.96	2.00
E4	31.94	2.11
E5	39.40	4.15
E6	50.32	2.79

"perfectly" used. In the perfect case, this line of best fit is $x = 0$; regardless of the energy input the battery voltage remains constant. Energy usage adapted perfectly in relation to the amount of energy input.

By comparing the slope of these lines, hypothesis $H2$ is tested:

> $H2_0$:The selfish hormone approach has no effect on the relationship between available energy on the battery voltage.
> $H2_1$:The selfish hormone approach decreases the influence of available energy on the battery voltage.

As shown in Table 5.7, there is a large difference in slope between the control (E1 and E3) and non control experiments (E2,E4,E5,E6). The increase in steepness between control and non-control experiments is calculated by dividing the non-control experiment slope by the control slope. Thus the slope of experiment E2, five nodes with selfish hormone enabled, compared to the E1, the five node control experiment can be calculated as follows:

$$\text{E2} = \frac{11.25 \times 10^5}{6.10 \times 10^5} = 1.84 \text{ times steeper than the control}$$

Table 5.8 shows the increase in line slope for experiments E2, E4, E5 and E6. The value for experiment E2 is calculated using the five node control experiment, E1. For experiments E4, E5 and E6 the 20 node control experiment, E3, is used.

Table 5.8: The increase in line slope, used as an indicator of power consumption adaptation, of experiments E2, E4, E5 and E6. Figures 5.5, 5.7, 5.8 and 5.9 show these linear models plotted against the associated data.

Experiment	Increase in slope from control
E2	1.84
E4	2.67
E5	3.29
E6	4.21

In experiments E2 and E4, the selfish hormone approach yields a steeper slope; 1.84 and 2.67 times as steep respectively. The average absolute error of the fitted linear model is very similar between the 20 node control experiment, E3, and 20 node selfish hormone experiment, E4. There is a smaller error in the fitted linear models for experiments E1 and E2, likely due to the smaller number of nodes. Experiments E5 and E6 used both selfish and anger hormones. The slopes for these experiments are also steeper than the control experiment, E3, with similar error in the linear model fit. The largest contributing factor to this error is likely to be measurement error as a result of using the light level as an estimate of energy received by a node.

In addition to this, the experiments that combined selfish and anger hormones, experiments E5 and E6, result in a steeper line than E4. This suggests that the combination of selfish and anger hormones is more effective at adapting the power consumption than just the selfish hormone.

It is notable that of the two experiments that used the anger hormones E5 produced a steeper slope than E6. Experiment E6 used a higher sensing rate, resulting in a steeper slope, than experiment E5 which used a "normal" sensing rate. The associated larger error in experiment E5s linear model, may have accounted for some of that difference. More likely is that the increased power consumption of nodes in experiment E6 was a stronger factor. The increased number of transmissions, 11 times more frequent, results in higher power consumption and therefore the regulation of power consumption in other areas became more important. The ability of the selfish hormone to reduce the power consumed by spending less time listening for packets allowed the control system in experiment E6 to stabilise even with the higher power consumption. Another factor may be that the anger hormones were produced more quickly due to the increased sensing rate. A node that was unable to communicate for a few minutes at a high sensing rate, in experiment E6, would result in a larger build up of anger hormone when compared to the same situation in experiment E5. This, in turn, may impact the power consumption of the nodes which receive the anger hormone.

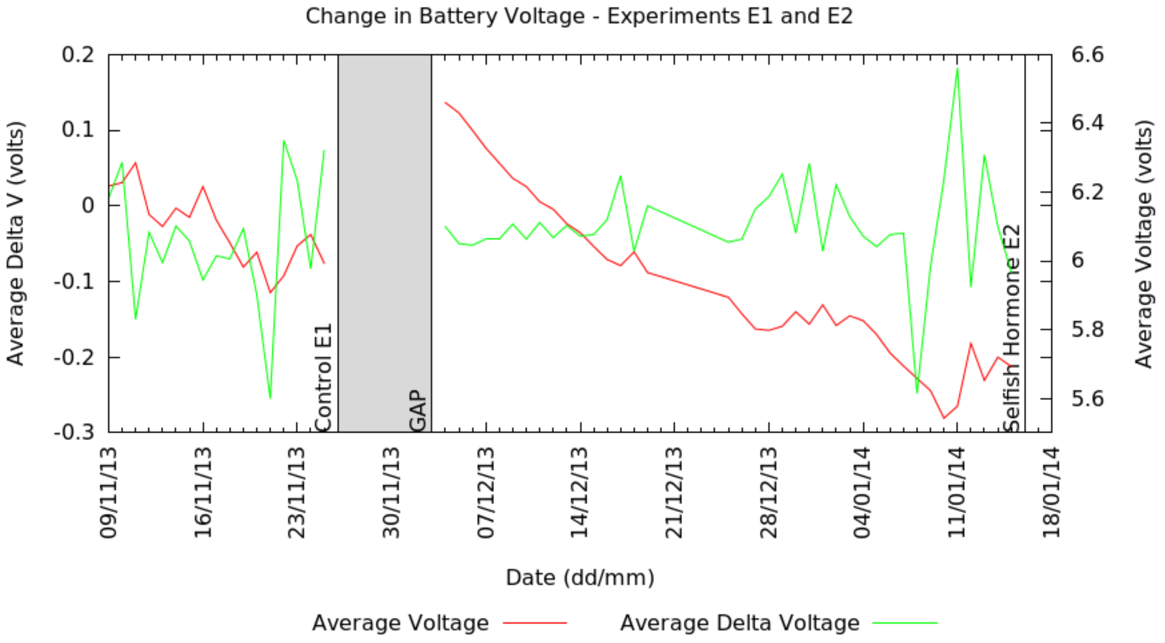

Figure 5.10: Average voltage and voltage change for the five node control experiment, E1, and selfish hormone experiments, E2.

Based on the predictive model and the slope and error of the fitted linear models (Table 5.7) we can accept $H2$.

Further support for hypothesis $H2$ is that the selfish hormone was capable of better managing power consumption in situations where less energy was available when compared to the control experiment. In the case of the five node experiments, E1 and E2, there was less solar energy available to nodes during experiment E2 than during the control experiment E1. Figure 5.10 shows the average battery voltage and change in battery voltage for the network over time in experiments E1 and E2. In the case of the control experiment, the average network voltage dropped until the 22/11/2013 after which it increased slightly. This slight increase is due to several nodes running out of power entirely, thus raising the average. The average voltage during the selfish hormone experiment drops at a similar rate, despite less power availability due to the time of year. More nodes survived until the end of E2 than the control E1, 80 % vs 60 % respectively.

Figure 5.11 shows the same average voltage and average change in voltage plot but for the mesh control and selfish hormone experiments, E3 and E4. The average battery voltage of the mesh control shows that most of the nodes were fully charged (greater than 6.5 V) which indicated that there was excess power available. This power was not being used for anything by the nodes. The lower average voltage during the mesh selfish hormone experiment, would suggest that the ability of the selfish hormone to increase the power consumption of each node resulted in the excess energy being used.

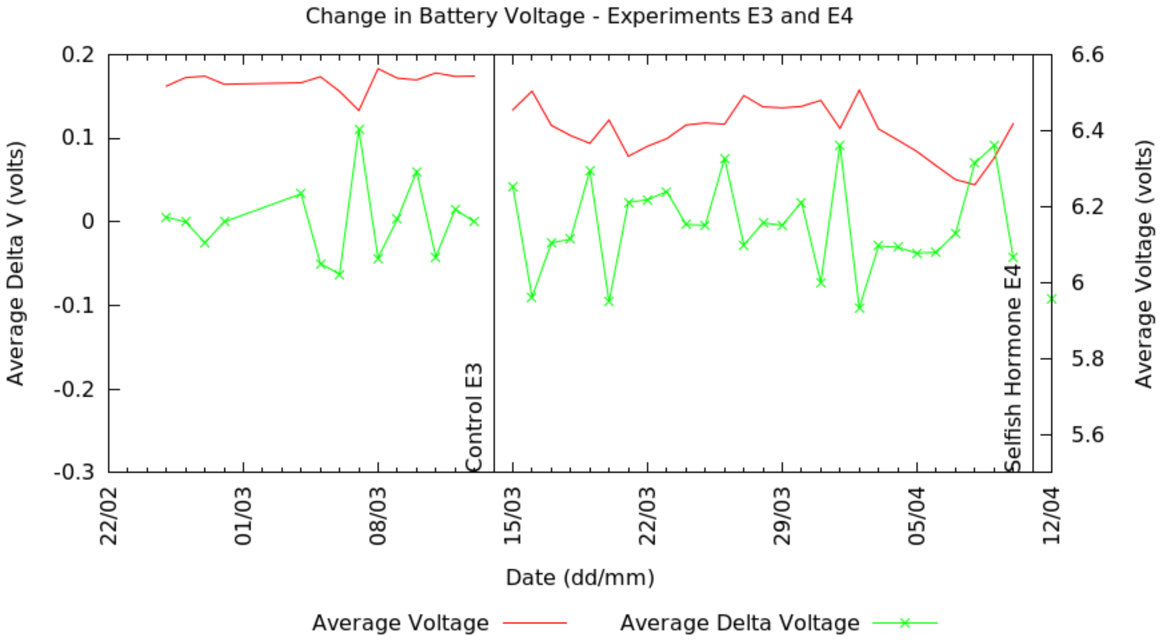

Figure 5.11: Average voltage and voltage change for the control experiment E3 and selfish hormone experiment E4.

The difference in available energy between these experiments is shown in Figure 5.12. This shows four histograms of the network's average integrated light per day. This integrated light value is an estimate for the amount of energy received by the network as a whole from the solar panels. The top two histograms show the five node experiments E1 and E2. It can be observed that during E2 nodes received very low amounts of energy, lower than during the control, E1. This is shown by the higher frequency of smaller amounts of integrated light. The average integrated light per day for E1 is 1.96 times as much as E2. The bottom two histograms show that in the case of the mesh experiments, the control experiment, E3, experienced a slightly reduced energy availability (0.87 times as much), in comparison to the selfish hormone experiment E4.

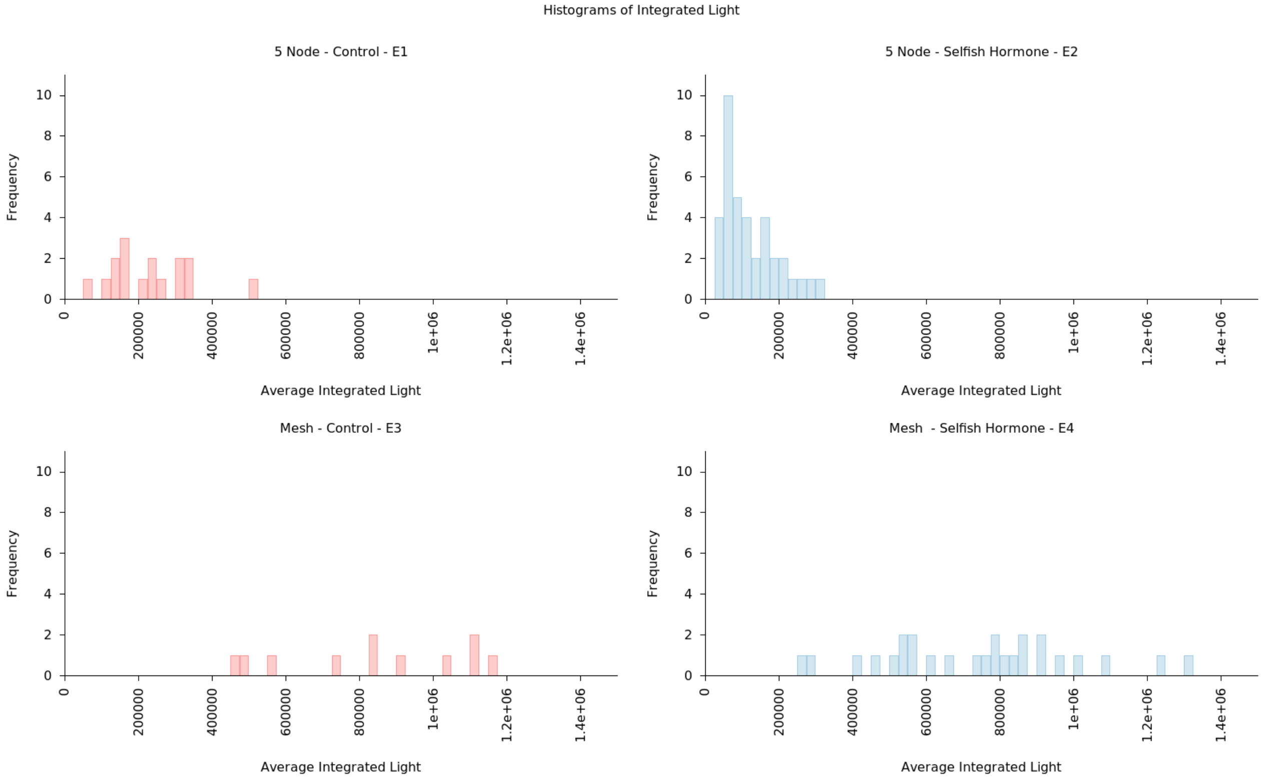

Figure 5.12: Histograms showing the distribution of the amount of integrated light, a unitless value, in each of control and selfish hormone experiments.

The model described in Section 5.5 suggests that an increase in slope for a linear model fitted to integrated light and change in battery voltage data is the result of better power consumption management. The slopes and errors of the linear models fitted to this data from experiments E1 to E6 are summarized in Table 5.7. There is an increase in slope, not attributable to the error, from both control experiments to the experiments using the selfish and anger hormones. The change in slope is not the result of a different quantity of energy entering the system, as shown by the histograms in Figure 5.12.

5.6.1 Effects of Anger Hormone

The anger hormone improved the power usage of the sensor network, see Table 5.8. This was achieved by modifying the duty cycle of nodes to improve communication. To determine whether the anger hormone was effective in doing so, an analysis of the quantity of data returned is presented.

The number of unique packets of data received per hour is calculated for each node. This value is expressed as a percentage of the number of packets received in the number of packets that were expected. For experiments E3 , E4 and E5 this is 60 packets per hour. As an example, receiving 58 packets in an hour is expressed as $\frac{100}{60} \times 58 = 96.7\%$. Due to the fact that the experiments ran for differing amounts of time, this hourly packet percentage value is averaged using the first week's data in each of these three experiments. Differences in the number of packets during this period of time will not be due to lack of power.

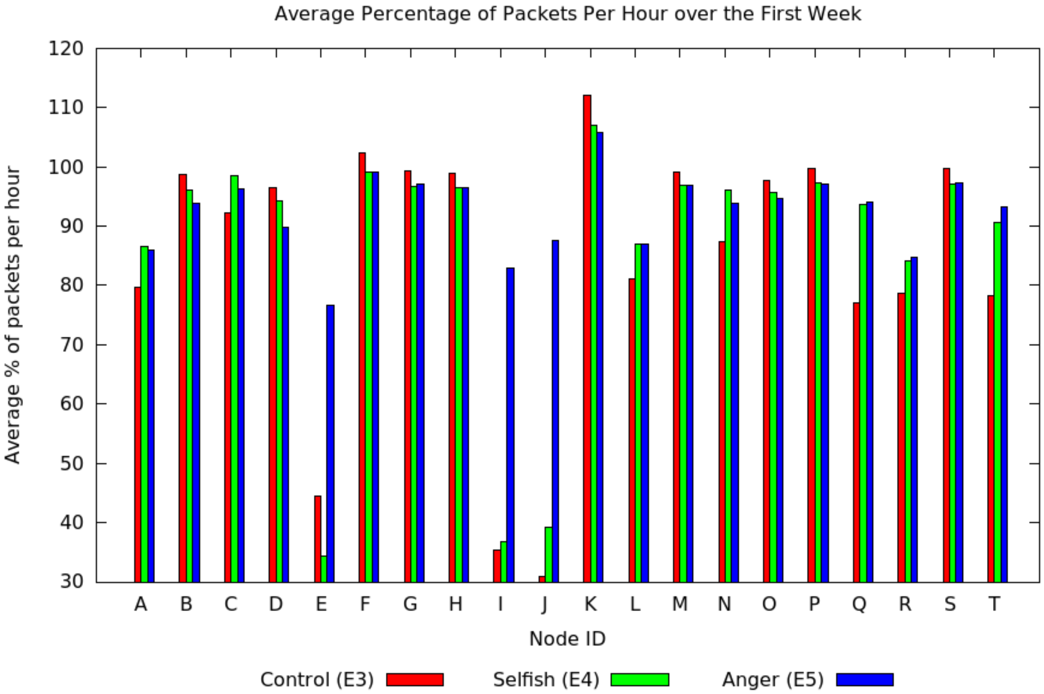

Figure 5.13: Average percentage of packets per hour for the first week of experiments E3 (control), E4 (selfish hormone) and E5 (selfish and anger hormone).

Figure 5.13 summarises this data for each node and each experiment. Certain nodes, K and F, for example, experienced a small decrease in the average number of packets per hour between the control experiment, E3, and experiments E4 and E5. Nodes that experienced this slight decrease were typically close to the base station and transmitted directly to it. This difference may have been attributable to small inaccuracies in the timing system used by the selfish hormone. Typically, this slight decrease in the average percentage of packet count did not improve when the anger hormone was added in experiment E5. This is due to the base station being always awake and listening for data. Any anger hormone produced by nodes that transmit directly to the base station do not have any effect on the base station, as the base station's duty cycle is 100 %. In most cases, the averaged percentage of hourly packets was very similar (within 1 %), between the selfish (E4) and anger (E5) experiments.

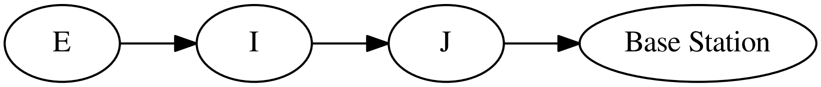

Figure 5.14: Network routing paths between nodes E, I, J and the Base Station.

There were three nodes that are noticeably very different; E, I and J. These nodes experienced a dramatic improvement between the control experiment and the anger hormone experiment. The average percentage of packets received increased from between 30% and 45% to between 75% and 86%. The three nodes were arranged in a line with E routing packets through I which, in turn, routed packets through J to the base station. Figure 5.14 shows the routing path between these nodes. The anger hormone's ability to increase the time a node spent awake and able to service routing requests resulted in an improvement in communication between these nodes. Other nodes showed an improvement from the control experiment (E3), for example Q and T. However, they improved between the E3 and the E4 (no anger hormone) experiments. As a result the improvement could not be attributed to the anger hormone.

Figure 5.15 shows the level of local and remote anger hormones for nodes E, I and J over the duration experiment E5. Node E is the furthest node from the base station and no nodes rely on it for packet routing. As a result its remote anger level remained 0 throughout the experiment, as it never received any anger hormone from other nodes.

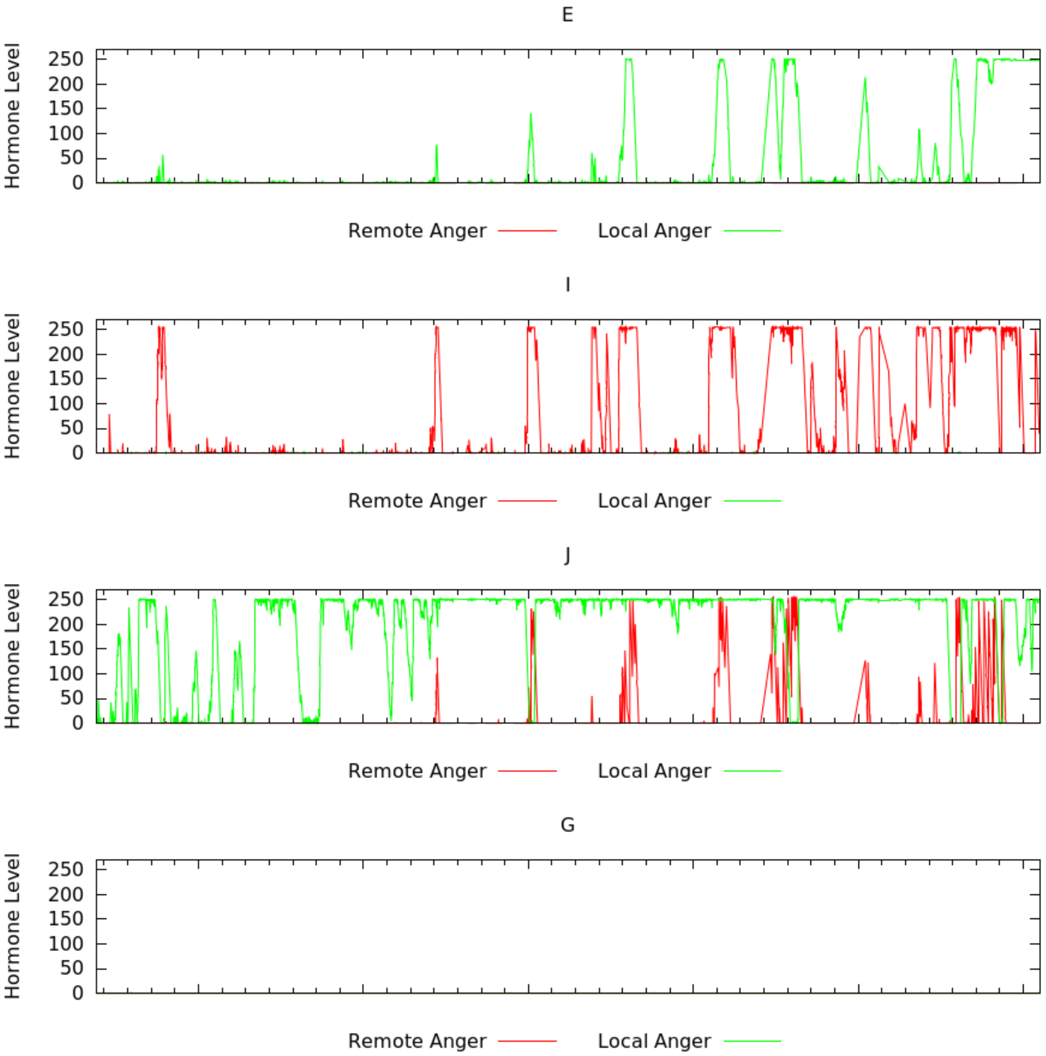

Figure 5.15: Graphs showing the level of local and remote anger hormones for nodes E, I, J and G over the duration of experiment E5.

Node I shows increases in remote anger hormone that correspond with the increases in node E's local anger hormone. This indicates that the anger hormone system is working successfully. Node I also generates small quantities of local anger hormone occasionally but was generally able to successfully transmit data to J. Node J showed amounts of remote anger hormone that correspond with I's local anger hormone. What was evident however, was that the local anger hormone was almost continuously at its maximum level. This indicated that node J consistently struggles to transmit data. Node J has to forward all of the packets from nodes I and E as well as its own. As a result there are more failures to transmit data resulting in an increase in anger hormone level.

Figure 5.15 also presents anger hormone data for node G. This node is responsible for routing packets for another node and transmits directly to the base station, in a similar way to node J. The graph shows that this node and the node it routed packets for are communicating without problem as there is almost no anger hormone present at all.

It can also be seen, in Figure 5.14, that as the number of hops a packet had to make increased the fewer packets eventually arrived at the base station. This is demonstrated best by nodes E, I and J. The closest node to the base station, J, has the highest packets per hour percentage. The furthest, E, has the lowest. The only exception to this is the control experiment where nodes E and J were accidentally swapped during the deployment. As the nodes are exactly the same the only result of this swap is that the names of the nodes is swapped for experiment E3, this should not have affected the outcome of the experiment.

As a result of the lack of replicates, it cannot be definitively proven that the anger hormone was effective. The data is, however, suggestive of the fact that in certain situations, the anger hormone benefits the routing performance of certain nodes.

5.7 Summary

This chapter presents the results of 6 experiments focussing on improving the use of power in a wireless sensor network using endocrine inspired methodologies. All experiments were run on real hardware in a real environment for extended periods of time, a minimum of two weeks. A baseline control experiment (E1) and experiment using the selfish hormone (E2) were run on a simple sensor network of 5 nodes, organised in a star topology. This was then scaled up to a mesh network of 20 nodes with support for multi-hop routing. In addition to the control (E3) and selfish hormone (E4) experiments, the anger hormone was also tested in two experiments. One at the default sensing rate (E5) and the other at an increased sensing rate (E6). The selfish hormone system used each individual node's battery level as the basis for generating a "selfish" hormone. While the level of this hormone was greater than 0, a node remained awake, consuming power and available to service routing requests. Once the hormone decayed the node would enter a sleep state, greatly reducing its power consumption. This cycle was repeated every half a second, continuously. The anger hormone affected the power consumption of each node in the same way as the selfish hormone. It was, however, generated in response to a failure to transmit data. When a packet was eventually successfully transmitted (and received), this anger hormone was contained within the packet and affected the receiving node.

The experiments demonstrated that an endocrine inspired power control system was able to be implemented in a wireless sensor network. A naïve model provided a prediction of how 'good' adaptive power consumption would be observed in the metrics chosen. Analysis of the experimental data showed that there was a good correlation between the amount of light received and the change in battery voltage over a day, validating the use of the predictive model. This result was apparent in all six experiments. The model described in Section 5.5 suggested that an increased line slope, of a linear model fitted to a node's energy input and output data, indicated an increased ability to adapt power consumption. Using this it was shown that the experiments using the selfish and anger hormones better adapted their power consumption when compared to the control with an increase in line slope of 2.67 and 3.29 times respectively compared to the control experiment. The anger hormone experiment (E5) showed a further improvement, compared to just the selfish hormone (E4), in the network's ability to use power. When the sensing rate was increased in E6, the network's energy usage was improved again with an increase in line slope of 4.21 times, compared to the control. Whether this was due to the effects of the anger hormone, or simply due to the potential for a higher power consumption was not determined. Individual node's lifetimes were extended and the energy usage of the network as a whole improved through the use of the selfish hormone.

The lack of data in cases where nodes were unable to transmit their environmental data, combined with design of the transmission and routing system, made determining the cause of failures very hard or impossible. Future hardware platforms would benefit from having some reasonably large amount of local storage to provide logging capabilities to each node. In the future, if time were available, performing multiple runs of experiments would provide better confidence in the data.

The results of this chapter show that the endocrine inspired techniques used would be useful in the sensing scenario presented in previous chapters. With low numbers of sensors, where each sensor is important, the ability to adapt power consumption while still considering the routing requirements of other nodes would be greatly beneficial. This has been achieved using the combination of two simple hormones.

Chapter 6

Data Quality

This chapter focuses on three experiments run using endocrine inspired methods to designed to improve data quality in addition to the selfish and anger hormones discussed in the previous chapter. The experiments were carried out in the same manner as those in the previous chapter. Equipment failures resulted in a subset of nodes ceasing to function and, as a result, a loss of data from these nodes. As the data is analysed on a node by node basis the impact of this is that experiments produced less data for the same run length. Three endocrine inspired approaches were investigated in order to determine whether endocrine inspired systems could improve data quality. They are described in more detail in Sections 4.7.4, 4.7.5 and 4.7.6 of Chapter 4 but can be summarised as follows:

- Detect rapid changes in light level to increase the sensing rate of a node and its neighbours, to acquire higher resolution data of the changing light level.
- Provide an endocrine-like mechanism for a user to influence the sensing rate of the sensor network. In particular it should combine with any other endocrine systems affecting the sensing rate.
- Use the wind speed and direction to promote and suppress the sensing rate of the whole network in differing wind conditions.

The metrics used to analyse these endocrine inspired mechanisms are discussed in Section 6.4. Three metrics are presented which aim to provide insights into the data collected from the sensor network. The first uses the number of packets received from each node to determine how much data was collected by the network and when this data was being captured. This metric is used to determine the effectiveness of the light, centre and wind hormones in increasing the amount of data captured by the network. The second metric analyses the relationship between the number of packets received from a node and the variability of the light level. Of all of the environmental parameters being recorded, the light is able to change fastest. This metric provides a method for determining whether the endocrine inspired systems are able to increase the data resolution during times of greater variability. Lastly the quantity of duplicate data is analysed to ascertain if and when unnecessary data was collected. Unnecessary data is sequential data points that are exactly the same, thus providing no new information. As there is both a power cost and routing cost in transmitting data, reducing this duplicate data is beneficial to network performance and power level. Due to the amount of time required to run each experiment it was not possible to run repeat experiments. It also was not possible to test each hormone in isolation, uninfluenced by other hormone systems.

Finally, the effect of the light, centre and wind hormones used on the power usage of the sensor network, is analysed using the metrics described in the previous chapter. By doing this the ability to adapt power consumption and simultaneously increase the data quality is assessed.

6.1 Data Quality

It is important to detail what is meant by data quality in the context of this work. A larger quantity of data is considered to increase the data quality. Having more data available for analysis is beneficial when examining the behaviour of the environment. This is especially the case as data was collected from a large variety of differing environments in close proximity. More data is better when it comes to analysing the interactions between these micro climates. Duplicate data should, however, be kept to a minimum. With larger quantities of data comes the requirement for more resources for the analysis of the data. Duplicate data does not contain any new information and in fact creates an unnecessary load on the sensor network. Minimising duplicate data is considered to improve data quality. Lastly, efforts to modify sensing behaviour in relation to the conditions being monitored are considered to be beneficial to the quality of the data returned. Ideally the rate at which, and manner in which a condition is monitored should be based on the behaviour of that condition. Higher environmental variability requires more data to reconstruct faithfully and periods of stability require less.

6.2 Experiments

Three experiments examining the effect of the light, centre and wind hormones on improving data quality were carried out. In addition, the two anger hormone experiments detailed in the previous chapter are discussed. It is important to note that the light, centre and wind hormones were added on top of each other and the selfish and anger hormones discussed in the previous chapter. A summary of which hormones were active in which experiment is shown in Table 6.1.

E7 - Light Hormone Experiment

The first experiment adds the light hormone. This uses changes in the light level to produce a light hormone. The hormone is broadcast to neighbouring nodes which, upon receiving the hormone, increase their sensing rate. The change in sensing rate is dependant on the level of the light hormone they received. The light hormone decays over the course of a few minutes, back to 0. At which point it ceases to have an effect on the sensing rate. See Chapter 4 Section 4.7.4 for details. The aim of this experiment is to test whether the light hormone can increase the amount and resolution of the light data when it is varying significantly.

E8 - Centre Hormone Experiment

The second experiment examines the ability for someone to control the behaviour of the sensor network, in an endocrine inspired way. The ability to release "centre hormone" on demand is introduced. By releasing this hormone a user is able to promote or suppress the sensing rate of the network for a period of time, until the centre hormone fully decays. The maximum and minimum sensing rates achievable using the centre hormone are once every 5.5 seconds to once every 12 minutes. The centre hormone fully decays in 12 hours. See Chapter 4 Section 4.7.5 for details. To test the centre hormone the maximal and minimal quantities of centre hormone were manually released on one morning and one evening. The result is a user triggered attempt to decrease the sensing rate at night, decrease the sensing rate during the day, increase the sensing rate at night and increase the sensing rate during the day.

Table 6.1: Summary of the three experiments (E7, E8 and E9) run to determine the effects of endocrine inspired control methodologies on data quality in a wireless sensor network. Also shown is E3, the control experiment.

Experiment Number	Selfish Hormone	Anger Hormone	Light Hormone	Centre Hormone	Wind Hormone	Deployment Type
E3	No	No	No	No	No	20 Node
E7	Yes	Yes	Yes	No	No	20 Node
E8	Yes	Yes	Yes	Yes	No	20 Node
E9	Yes	Yes	Yes	Yes	Yes	20 Node

E9 - Wind Hormone Experiment

The last experiment uses a ROWind ultrasonic wind sensor at the base station node. The variability of wind direction and the wind speed act against each other to produce a quantity of wind hormone. The wind hormone is transmitted throughout the network once per hour. Once received by nodes its effects combine with the centre hormone and light hormone to produce promoting or suppressing effects on the nodes sensing rate. This process allows the sensing rate of the whole network to be modified by the wind conditions as measured by the base station. See Chapter 4 Section 4.7.6 for details. The concept is that different wind conditions will result in the environment being more or less likely to change quickly. The manual triggering of centre hormone carried out in the previous experiment, E8, is repeated.

The same experimental procedure, set out in Section 5.2, of the previous chapter was used. Nodes were placed in the same locations as in previous experiments and the routes used for multi-hop packet routing were consistent between all experiments. Equipment failures resulted in loss of data from around half of the deployed nodes. The cause is still unknown although it was possibly due to hardware faults, antennae damage or a software bug. More than half of the nodes remained fully operational throughout all three experiments. The metrics chosen were calculated on hour long periods on a node by node basis and, as a result, there is still a substantial quantity of data for analysis.

6.3 Hypotheses

There are a number of hypotheses tested in order to test data quality aspect of the main hypothesis set out in the Introduction chapter.

The first hypothesis, $H3$, tests whether the quantity of data from a wireless sensor network can be increased using endocrine inspired techniques. This is considered to constitute an increase in data quality.

> $H3$: The number of data packets from nodes will be increased from the control experiment using the light hormone.

$H4$ tests whether the centre hormone allows a user to trigger an increase or decrease in the network's sensing rate. The ability to have some control over the network is considered important to data quality as it enables an increase in temporal resolution on demand.

> $H4$: The centre hormone can be used to manually increase or decrease the sensing rate of the network.

The light hormone is designed to produce an increase in the quantity of data from the network, as tested by hypothesis $H3$. However, more specifically it should result in an increase in the quantity of data from a node as the variation in light level increases.

> $H5$: The use of light hormone produces a correlation between the standard deviation of the light level and the packet count.

The wind hormone also has the ability to increase and decrease the sensing rate of a node. Thus it is tested with hypothesis $H6$.

> $H6$: The use of wind hormone produces a correlation between the standard deviation of the wind speed and direction and the packet count.

The final hypothesis is $H7$. We have already discussed that duplicate data lowers data quality, this leads to:

> $H7$: Experiments using the light, centre and wind hormones will produce fewer duplicate data packets when compared to a fixed increase in sensing rate.

6.4 Metrics

The following sections detail the metrics used to test the hypotheses in the previous section. How each metric is calculated and the reasoning behind its use is discussed. In all of the metrics used, where packets are counted, duplicate packets were discarded. Duplicate packets are packets are instances where the same packet was transmitted and received multiple times. They are, therefore, not unique pieces of information.

6.4.1 Packet Counts

The simplest metrics used is the counting of unique data packets received at the base station. Each day of an experiment is divided into 24, hour long periods, starting and finishing on the hour. For each of these hours, the number of packets received, per node and total, is counted. The number of packets received during this hour and the expected number of packets is then used to calculate a percentage of actual vs expected packets for this time period. The default sensing rate is set at 60 packets per hour. A node transmitting 59 packets in an hour achieves a percentage of $(59/60) \times 100 = 98.3\%$. The theoretical value of 60 packets per hour remained constant throughout all of the experiments analysed, including ones in which nodes are capable of transmitting more. This provides a method of comparing the quality of data, between experiments. Higher packet counts imply more data, although this metric does not factor in whether the information contained within packets was of use. It is therefore possible to have a high packet count but for many of the packets to contain the same data due to lack of change in the environment.

6.4.2 Standard Deviation against N method

Ideally, to minimise unnecessary data and maximise the resolution at which the environment is sensed, data would only be collected when the monitored parameters in the environment change. When they do change, the change should be captured in as high a resolution as possible. To quantify whether this is achieved the standard deviation of a parameter as sensed by each node is calculated over each one hour period, starting on the hour, for the duration of each experiment. This indicates how variable the value of the parameter in question was over each hour long period. For the same period of time, the number of packets that were received by the base station is calculated. The relationship between these two values is indicative of the quality of the data from that experiment. If the standard deviation is low then the environment was not changing much and the number of packets should be low. When the standard deviation is high, the environment was changing and the packet count should be higher so as to try to better capture these changes.

Light is used as the parameter being observed for this metric. There are several reasons for this. The first, is that light is used by the light hormone system to attempt to improve data quality by promoting sensing rate during periods of rapid change. The second was that, as a parameter, it varies much more quickly and by a greater amount than the other parameters being measured. As such, it is the transitions in light level that would benefit the most from a higher sensing rate.

Standard deviations of a single value can not, and are not, reported. Hours with only one data point are discarded. However, as long as an hour time period contains two or more data points, it is included. The calculated standard deviation of these hours may not be representative of the "actual" standard deviation. While this may result in noisy data, any relationships should be apparent nonetheless and there is not strong justification for an arbitrary minimum threshold.

6.4.3 Duplicate Data

Duplicate data is of little to no value in understanding the behaviour of the environment, ideally data is only required when change occurs. There are likely to be times when the environment is in a relatively stable state and so there will be very little to no change in short time frames. Given a fixed sensing rate, this is likely to lead to periods where data is duplicated due to the lack of change in the environment. Ideally this data would not be transmitted to the base station as it does not provide more information about the environment.

By counting how many subsequent packets were duplicates in hour long periods, a measure of duplicate data is obtained. It also provides data on when duplicate information was being collected.

The method for determining whether a packet contains duplicate data, or not, is as follows. From each packet the following fields are extracted; temperature (sensor 1), pressure, temperature (sensor 2), humidity, light level and battery. The string representation of these values are concatenated to form the "unique packet data". This unique packet data is constructed for the current and previous packets and compared. If equal then the duplicate count for that node, for that hour is incremented. If any one of the parameters were to vary at all then the two are not considered equal. This approach is likely to underestimate the amount of duplicate data. Many of the sensors provided data to two decimal places and are by no means noise free. Thus if, for example, the temperature measurement varied by 0.01 °C between packets, they would be considered different. Given the number of sensors and their quality, it is very likely that there would be variation between subsequent readings.

It may be possible to provide some "equality range" for each parameter, whereby if two values fell within that range of each other they are deemed equal. This is problematic as choosing these values such that they discriminate between genuine change in the environment and sensor noise is difficult and requires strong justification.

It is therefore accepted that duplicate data is likely to be under reported. It is expected that rapid sensing during periods of stability is likely to result in more duplicates and that rapid sensing during periods of variability would result in very few or no duplicates.

6.5 Results

In this section, each metric described in the previous section is applied to experiments E7, E8 and E9 to test the hypotheses set out in Section 6.3. Experiment E3, the control experiment with no hormones enabled, is used as the baseline for comparison. Experiments E4, E5 and E6 which are also discussed in the previous chapter are used for comparison where appropriate. The hypotheses from the previous chapter regarding the ability to adapt power consumption are tested, in Section 6.5.4, against the data obtained from experiments E7, E8 and E9 to determine the effects of the light, centre and wind hormones on power consumption.

6.5.1 Packet Counts

Before analysing the behaviour of the experiments presented in this chapter it is important to apply the packet count metric to previous experiments to see if the selfish and anger hormone impacted the packet count. Ideally 100 % of the expected packets from each node would be successfully received by the base station, due to routing or transmission failures this is not the case.

The behaviour of the packet count over time for experiments E3, E4 and E5 (discussed in the previous chapter) is very similar. For nodes that can transmit directly to the base station the packet count remains at 100 %, ± 1%, for the duration of those experiments. Nodes that require their data to be routed via an intermediate node produce between 60 and 80 % of the expected 60 packets per hour. This behaviour is to be expected as these experiments are focussed on improving power consumption rather than data quality and do not attempt to explicitly increase the sensing rate. Figures 6.1 and 6.2 show the packet count percentage calculated on an hour by hour basis for the control experiment E3 and selfish hormone experiment E4. Gaps, due to the base station crashing and data being lost, can be seen in the control experiment.

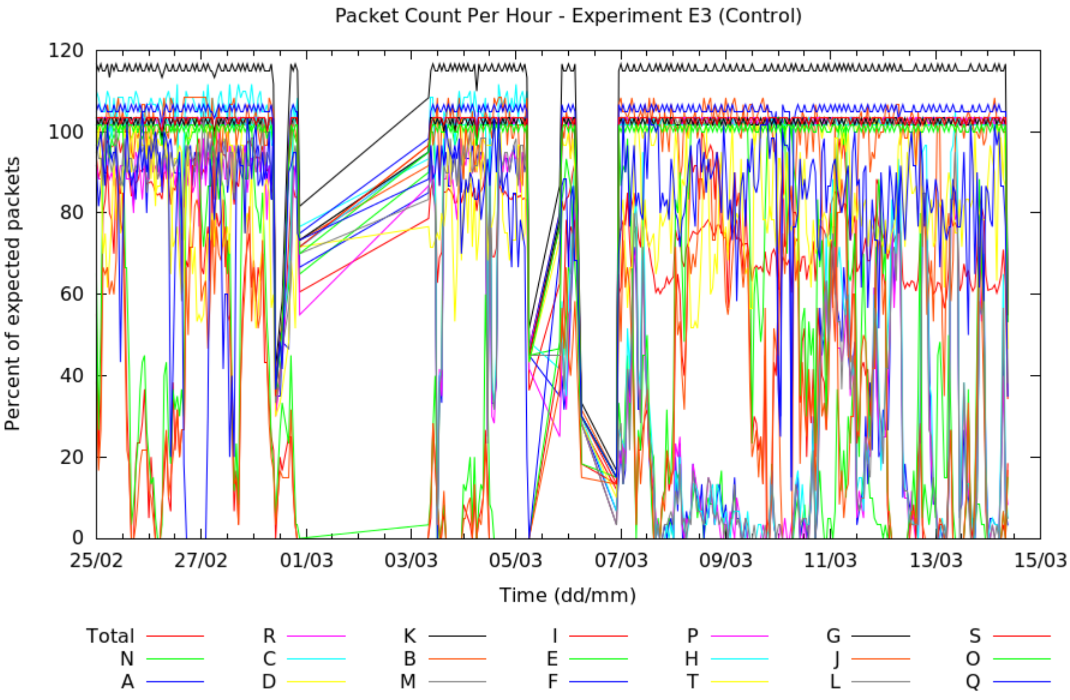

Figure 6.1: Packet count per hour, per node for the control experiment, E3.

Towards the end of the control experiment it can be seen in Figure 6.1 that the packet count of some nodes declines. This could be due to a variety of factors such as battery level or environmental conditions. The design of the network resulted in groups of nodes that relied on another node for their communication. If this key node was unable to transmit

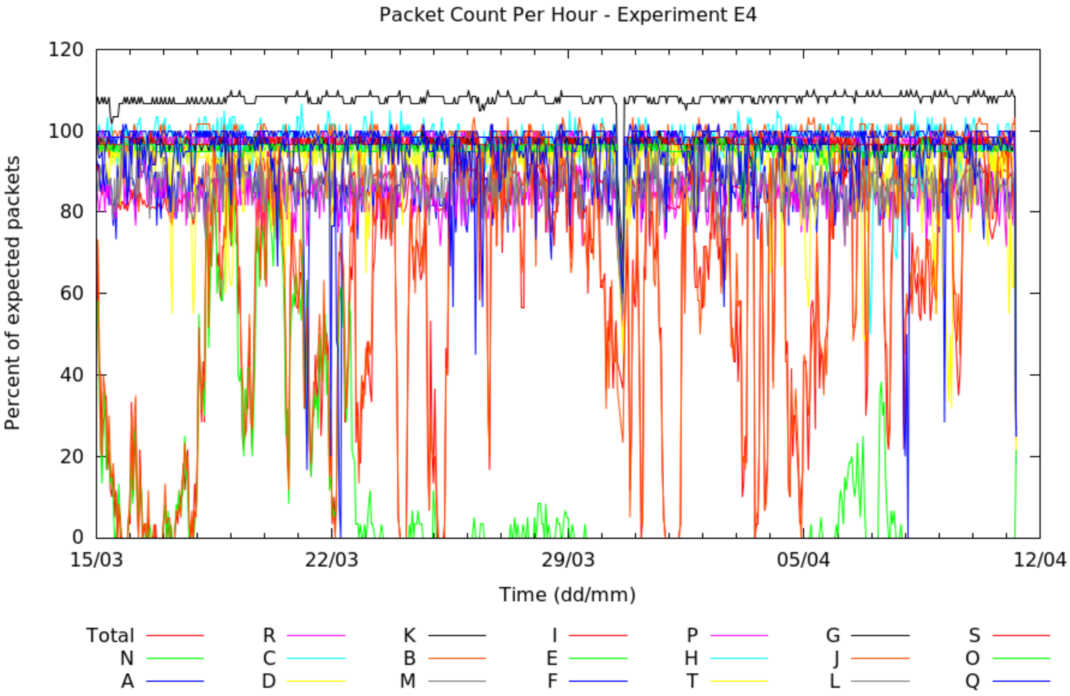

Figure 6.2: Packet count per hour, per node for the selfish hormone experiment, E4.

optimally the packet count of reliant nodes suffered. This could be seen in several places in the graphs but was particularly noticeable in the experiment E4, Figure 6.2. For the first 5 – 7 days the packet count of three nodes; E, I and J, can be seen to track each other. In this case E routed through I which in turn routed through J to the base station. Experiment E5, in which anger hormone was introduced, resulted in the same packet count behaviour as shown in Figures 6.1 and 6.2.

Figure 6.3 shows the packet count graph for experiment E6, which used selfish and anger hormones but with a sensing rate 11 times faster than the default used in all other experiments. The percentage of packets received compared to the expected number is calculated using the default rate of 60 packets per hour, or one per minute. As a result each node should theoretically transmit 1100 % of the expected packet count. There is a subset of nodes that are consistently within 10 % of the theoretical packet count. Others produce a more variable number of packets but stay within 600 – 800 % of the expected number. Figure 6.3 illustrates two further interesting points. The first is that, in Figures 6.1, 6.2 and 6.3, there was one node that consistently transmitted more packets per hour than any other node. This node, K, transmitted directly to the base station and so was not

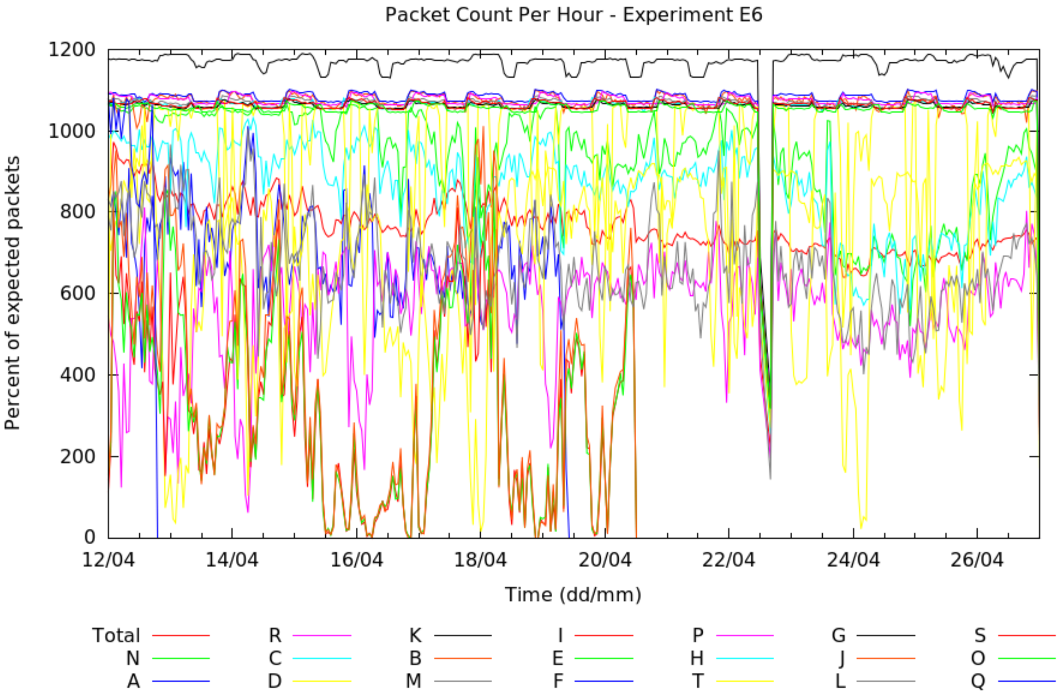

Figure 6.3: Packet count per hour, per node for experiment E6 which used selfish and anger hormones but with an increased sensing rate 11 times greater than the default. The percentage of packets received is still calculated using the default rate of one packet per minute, 60 packets per hour. As a result a perfect node should transmit 1100 % of the expected packet count.

reliant on other nodes to route its packets. This is not unique to this node however. The node was identical to every other node in every way other than its address. The reason that this node consistently transmitted more packets than any other node is the result of either its internal or external oscillator running faster than the other nodes. This was not by design, the ceramic resonators used as the clock sources for the microcontrollers on the nodes, have a high potential frequency error as do the RC timer circuits used for sleep timing. Any timings performed by the node are reliant on the resonator. In this instance, this particular node's resonator ran measurably quicker. As a result this node consistently transmitted 5% faster.

The second point of interest, is that some degree of periodicity can be seen in the nodes with consistently high packet counts.

6.5.1.1 Packet Count Periodicity

It is important to establish a cause for this behaviour as it there was no intention to produce it. Figure 6.3 shows the packet count for experiment E6. There is a clear diurnal cycle in packet count evident in nodes with a high packet count percentage. The nodes with the highest packet counts transmit directly to the base station, which removes the routing system as a possible explanation for this behaviour. The number of packets from these nodes fluctuates by as much as 34 packets per hour between day and night.

A similar behaviour can be seen in the experiment E4, Figure 6.2 but is not apparent in the control experiment, E3. Increasing the sensing rate in experiment E6 resulted in this change in packet count becoming more apparent. The change is diurnal, the packet count increasing at night and decreasing during the day. As previously stated nodes producing this behaviour did not all route their packets through intermediate nodes and some were not involved in routing packets for any other nodes. It is not possible that the routing system is the cause of the behaviour, leaving environmental factors as the most likely cause. The environmental factors that are recorded by each node are temperature, air pressure, humidity and light level. The cause of the periodicity may not be one of these factors, however as this data is available it is possible to test using the following hypothesis.

> H_0: There is no correlation between environmental factors and the packet count.
>
> H_1: There is a correlation between environmental factors and the packet count.

Although the battery level was not considered an environmental parameter, it is able to impact the performance of the node, due to the selfish hormone. It is, therefore, included in the following analysis. The environmental factors and battery voltage are averaged over one hour periods to match the periods that the packet count was calculated over. It is then possible to analyse the relationship between packet count and each of the previously mentioned parameters. All nodes that exhibit this periodic behaviour, according to Figure 6.3, are analysed using Spearman's correlation coefficient using R's *cor.test()*[26] function.

Table 6.2 shows the results of the correlation statistic. The correlation coefficient for each parameter is averaged to produce the average correlation for each parameter. The best correlation is light, -0.686, followed by temperature, -0.622, and battery, -0.588. Humidity shows a positive correlation and pressure weakly negatively correlated. The three

Table 6.2: Correlation between different environmental factors and packet count for experiment E6. Each environmental parameter is averaged on an hour by hour basis to match the packet count. Values are rounded to 3 decimal places. All p values are below $2{\times}10^{-10}$ and are shown in Appendix C. N is 370 for all correlations.

Node	Temperature	Humidity	Light	Battery	Pressure
G	-0.66	0.466	-0.703	-0.569	0.129
O	-0.585	0.384	-0.563	-0.458	-0.08
S	-0.595	0.374	-0.69	-0.516	-0.152
F	-0.589	0.395	-0.764	-0.624	-0.087
P	-0.756	0.571	-0.728	-0.716	-0.127
H	-0.52	0.319	-0.651	-0.472	-0.108
B	-0.485	0.304	-0.608	-0.53	-0.022
M	-0.67	0.464	-0.754	-0.589	-0.138
K	-0.741	0.599	-0.711	-0.821	0.102
Average	-0.622	0.4317	-0.686	-0.588	-0.105

parameters with the highest correlation; light, temperature and battery, are all related. An increase in any one of these parameters was likely to result in an increase in the others. As light increases, during the day, the temperature also increases and the battery voltage is pulled higher by the light. As a result of this we are able to accept $H1$ and say that the packet count is influenced by environmental parameters.

There are several possible explanations for the periodicity. The high correlation with light and temperature could indicate that the cause is the ceramic resonator used as the oscillator for the microcontrollers. An increase in temperature can result in the resonator oscillating more slowly. As this is the source of all timings in the microcontroller, this may explain the decrease in packet count. Another possibility is that the RC Oscillator used for controlling the sleep times of the microcontroller is both temperature and voltage dependant. Increases in temperature and/or voltage could have resulted in inaccuracies in the amount of time the microcontroller spent in it's low power sleep mode. Another possibility is that, due to the inaccuracy of the low power RC Oscillator used to control sleep times, the more time a node spent "sleeping", the more inaccurate the intervals between packets became.

While no definite cause was found, the fact that the packet count correlated to environmental factors that are not directly controlled is worth noting. The difference between day and night is more pronounced with a higher sensing rate. The difference in experiments E3, E4 and E5 which used the default sensing rate of 60 packets per hour is only around $\pm 1 - 2$ packets per hour. It is only when the sensing rate is increased, as in experiment E6, that it becomes more noticeable.

It is important to note that in experiments that did not utilise the light, centre or wind hormones the sensing rate of nodes was never more than 1 – 2 % above the expected number. As a result, observed packet counts of more than 110 % in experiments E7, E8 and E9 are considered to be have an increase sensing rate.

6.5.1.2 Light, Centre and Wind Experiments

The results of the packet count analysis on the light, centre and wind experiments indicate that the control light, centre and wind hormones were able to increase the amount of data received from the network.

The first hypothesis to be tested is $H3$:

> $H3_0$: The number of data packets from nodes will not be increased from the control experiment using the light hormone.
> $H3_1$: The number of data packets from nodes will be increased from the control experiment using the light hormone.

Figure 6.4 shows the packet rate over time for the light hormone experiment, which sought to better capture changes in the environment during rapid changes in light level. The packet count can be seen to increase significantly during the day, at times reaching over 700% of the expected 60 packets per hour. 12/06/14 exhibits the lowest peak of 500 %, which correlates with the fact that there was not much variability in the light level for that day due to clear skies. The diurnal change in sensing rate discussed in the previous chapters was only 1 – 2% using the default sensing rate of one packet per minute. As a result the change in sensing rate cannot be due to environmental conditions and must be the result of the light hormone. Based on this we accept $H3$.

The centre hormone experiment, E8, examined whether it is possible for a user to control the behaviour of the sensor network in a hormone inspired manner. The ability to promote and suppress sensing behaviour, using a centre hormone, was added. During the

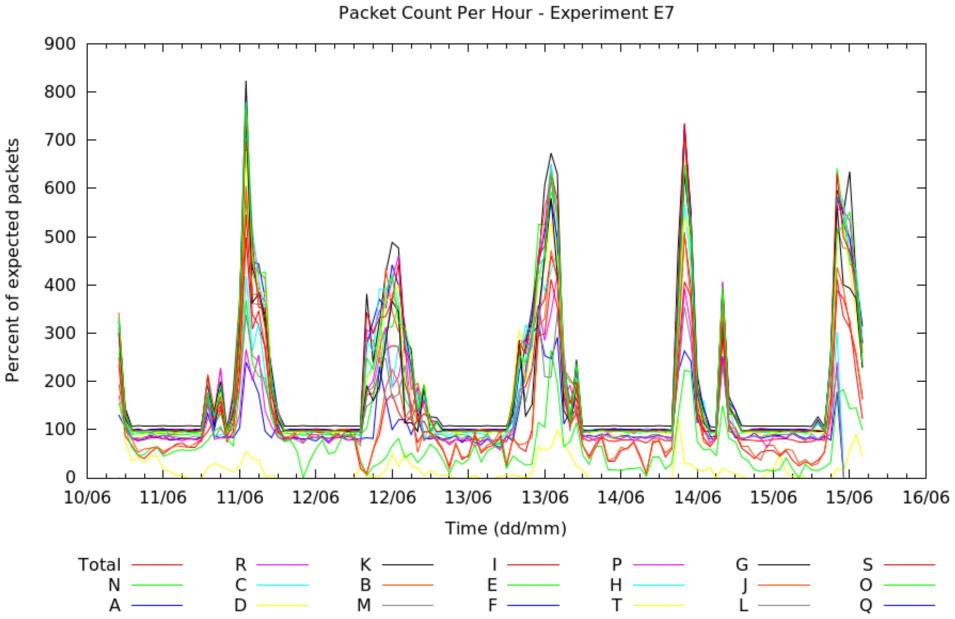

Figure 6.4: Packet count per hour, per node for experiment E7, which used the light hormone.

course of the experiment, the base station was used to broadcast centre hormone to the network to both promote and suppress the sensing rate during day and night. Hypothesis $H4$ states:

> $H4_0$:The centre hormone can not be used to manually increase or decrease the sensing rate of the network.
>
> $H4_1$:The centre hormone can be used to manually increase or decrease the sensing rate of the network.

The packet count for experiment E8 is shown in Figure 6.5. Periods of time when centre hormone was released to attempt to suppress or promote the sensing rate are highlighted with grey.

The first instance highlighted was a test to ensure that the system was working as intended. There are two instances where the centre hormone was released to suppress the networks sensing rate. The first during the day of the 11/7/14 and the second during the night of 17/7/14. Examining Figure 6.5, the attempt to suppress the sensing rate during the day was not successful on all nodes. Multiple nodes still transmit over 600 % of the expected number of packets. Examining the data from these nodes shows that they did not receive the centre hormone. Nodes that did receive the centre hormone show a reduced packet count, less than 100 %, however the sensing rate of these nodes was simultaneously increased by the light hormone. The result is a mixing of the two behaviours, with the sensing rate moving between 50 % and 150 % depending on the light hormone.

The attempt to suppress the sensing rate during the night of 17/07/14 shows a fixed reduction in sensing behaviour until the decaying centre hormone became unsaturated 7 hours later. Once the centre hormone becomes unsaturated, in the early hours of the 18/07/14, the sensing rate gradually returns to the default before being increased during the day by the light hormone.

The two occasions where the centre hormone was used to promote the sensing rate were similar despite one being during the day and the other at night. This is due to the fact that there were no mechanisms to decrease the sensing rate whilst the centre hormone was increasing it. The only other hormone to affect the sensing rate is the light hormone which can only increase the sensing rate. On both occasions where the centre hormone was released to increase the sensing rate, most nodes increased their sensing rate, exceeding 1000%, or 600 packets per hour. It is possible to see during the first promotion period, the night of the 13/07/14, that some nodes did not increase their sensing rate. Their packet count remained at 100%, 60 packets per hour, throughout the night of the 13/07/14. Due to the broadcast mechanism, there is always a chance that a node will not receive a broadcast. Looking at the data returned by these nodes it was found that they had not received the centre hormone. The alternative to the broadcast system used, is a more complicated mechanism to ensure reception by every node.

On three of the four occasions that centre hormone was released, there is an immediate change in the number of packets received from each node. The direction of change corresponded to whether the centre hormone was used to increase or decrease the sensing rate. The release of centre hormone in the morning of the 11/07/14 to suppress the sensing rate was not received by all nodes. The nodes that did receive it exhibited a behaviour corresponding with the combination of light and centre hormones. These change are not attributable to anything other than the centre hormone. Given these facts we can accept $H4$.

The packet count during the wind hormone experiment (E9), shown in Figure 6.6 is much more variable than during any other experiment. The addition of sensing rate suppression and promotion due to various wind conditions results in almost no extended periods of the default sensing rate. The effect of the centre hormone can be seen to still

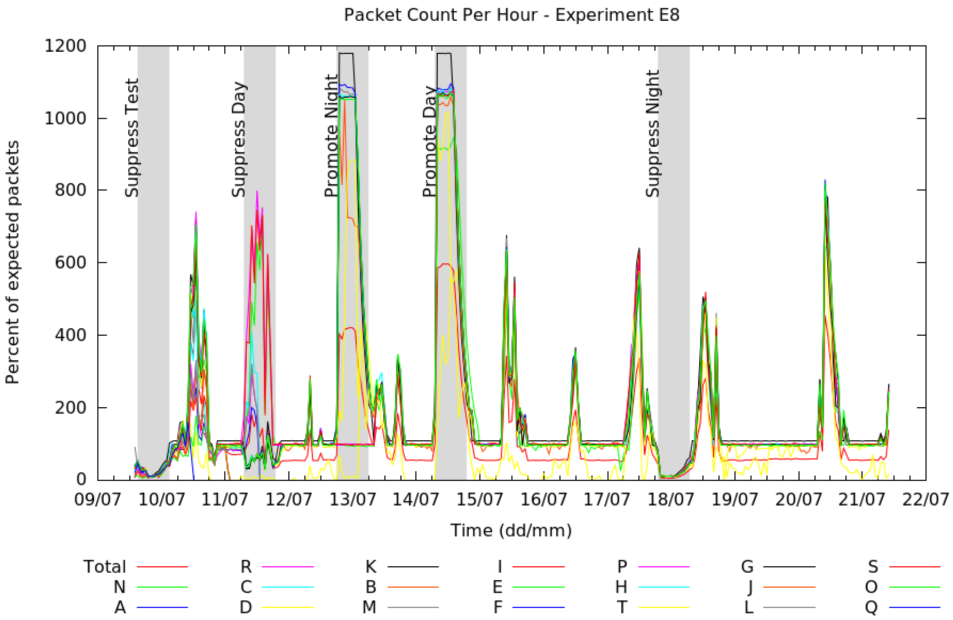

Figure 6.5: Packet count per hour, per node for the mesh centre hormone experiment.

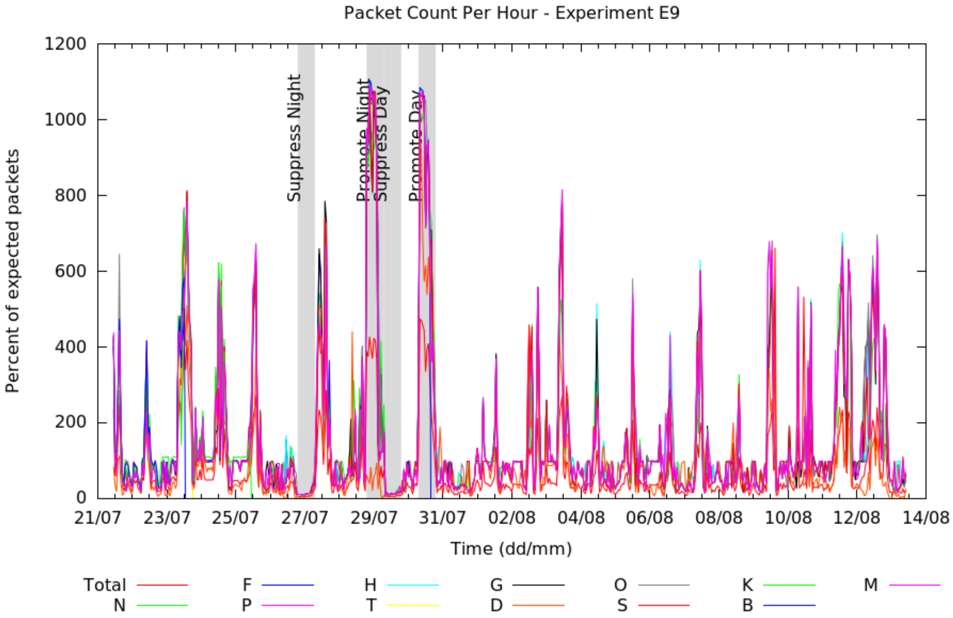

Figure 6.6: Packet count per hour, per node for the mesh wind hormone experiment.

be effective in promoting and suppressing the overall sensing rate. The centre hormone's ability to decrease the sensing rate was particularly effective as the wind hormone's ability to decrease the sensing rate countered the light hormone's ability to promote sensing rate.

Centre hormone promotion produced the same behaviour as in experiment E8, producing an increase in the sensing rate of the majority of nodes of above 1000%. The light hormone continued to promote sensing during the day due to rapid changes in light level. The biggest difference is seen during the night during which, in all previous experiments, the packet count was constant at 100%, unless the centre hormone was being used. During experiment E9, the number of packets received during the night was much more variable and was frequently lower than the default 60 packets per hour.

6.5.1.3 Packet Count Conclusion

The packet count is a useful metric for determining whether the endocrine inspired methodologies used were effective in increasing the amount of data from the sensor network. Figures 6.1 and 6.2 show that the approaches to power management did not negatively impact the packet count and were not responsible for increasing the sensing rate above the default. The light and wind hormones were also shown to be effective in changing the number of packets received per hour based on various environmental factors. In addition, the packet count of the centre hormone experiment showed that it was possible for a user to interact with the network in an endocrine like fashion to increase or decrease the sensing rate. However, the number of packets, while an interesting metric does not tell the whole story. If maximising the number of packets was the overall goal then increasing the sensing rate, as was done in experiment E6, would be the simplest solution. The result of this fixed increase was a very large amount of data, much of which was not useful due to it not containing any new information. Increased amounts of data produces overheads in the routing system and consumes power that could be otherwise used, or could extend node lifetime. As well as these concerns there are problems with processing, analysing and storing the data. The increased sensing rate anger hormone experiment produced over 600 Mb of data in two weeks. If this scaled linearly then 100 nodes would produce almost 80 Gb of data in a year. Deployments to cover larger areas in a higher resolution, sensing a larger number of parameters could easily produce Terabytes of data per year. Ideally data would only be gathered when “something” was happening in the environment. It is advantageous to sense more when the signal being measured is more variable and less when it is constant.

6.5.2 Standard Deviation and Sample Size

The light, wind and centre hormones were designed to improve the quality of the data returned from a sensor network. The light and wind hormones aim to better capture the environment in response to some environmental stimulus. In the case of the light hormone, rapid light changes provide the stimulus and for the wind hormone, it is changes in wind direction and wind speed. The centre hormone exists as a mechanism for a user to interact with the network. This presupposes that there is a predictable event that is cause for wanting the sensing rate to be increased or decreased. The previous section analysed experiments E7, E8 and E9 using the packet count metric. It was shown that more data could be produced using endocrine inspired methodologies. As previously discussed data quality also takes into consideration the number of samples of a signal and the variability of that signal.

To determine whether these hormones have improved the data quality, the data from each node is split into hour long sections. The standard deviation of the light is calculated and plotted against the number of samples in that hour. Where the packet count metric gives an estimate on how much data was collected, the standard deviation metric gives an indication as to how well the data maps the actual light level. The light level is a good environmental parameter to use as it is the most variable in the shortest amount of time compared to the other parameters being recorded. As the standard deviation of the light level increases more samples are required to be able to reconstruct the original signal faithfully. Taking many measurements on a signal that is invariant wastes resources and does not improve the quality of the data.

Experiments E4, E5 and E6 used hormones that were designed to improve power consumption based on the available energy. The graphs of standard deviation plotted against number of samples, per hour, are shown in Figures 6.7 and 6.8.

Experiments E3, E4 and E5 used a sensing rate of 60 packets per hour which resulted in obvious concentrated bands at around 60 data points. Here the standard deviation can be seen to rise and fall, while the number of samples per hour remains, relatively, constant. There are some points in between 0 and 60, however they are distributed evenly and are the result of failures to transmit all of the data. A second smaller collection of points can be seen at around 65 samples. This is due to, as previously discussed, one node transmitting more frequently due to its oscillator. Experiment E6, which used a higher sensing rate,

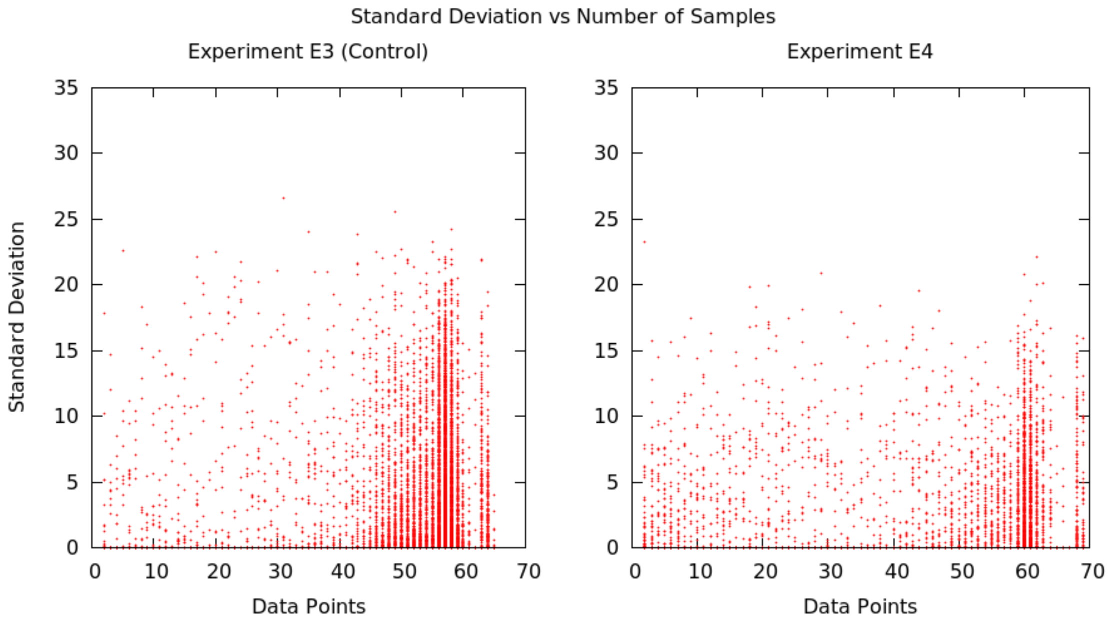

Figure 6.7: Standard deviation of light calculated over an hour plotted against number of packets received in that hour for the control experiment, E3, and selfish hormone experiment, E4.

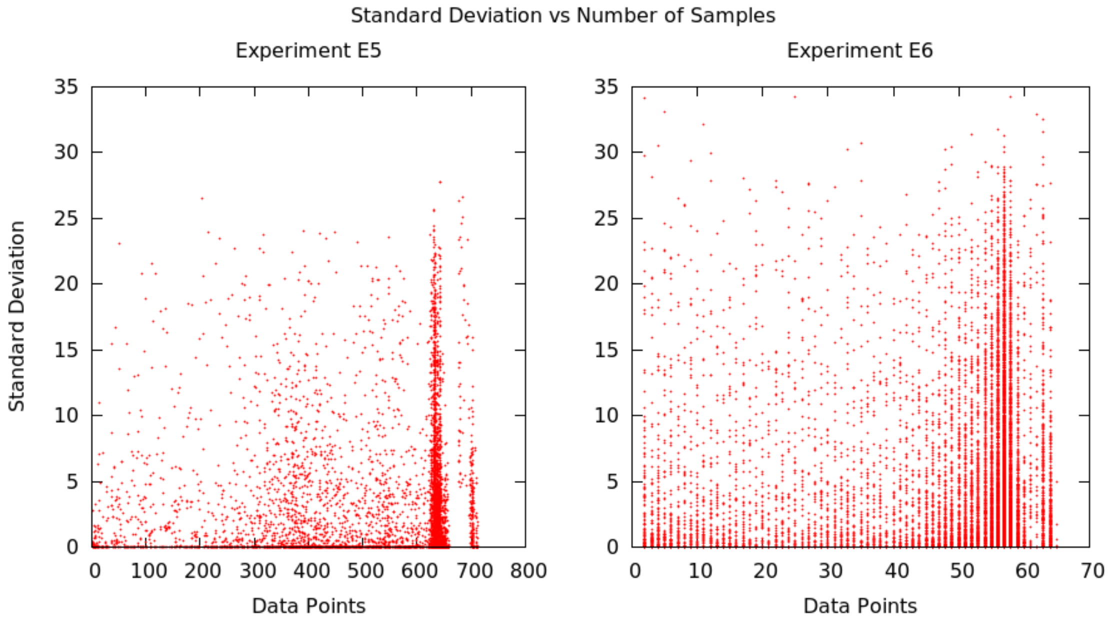

Figure 6.8: Standard deviation of light calculated over an hour plotted against number of packets received in that hour for experiments E5 and E6.

shows similar clusters of points. They are, however, clustered at around 650 data points with the "fast node's" smaller cluster at around 700.

Hypothesis $H5$ is applied to every experiment run, E3 to E9.

> $H5_0$: The use of light hormone does not produce a correlation between the standard deviation of the light level and the packet count.
>
> $H5_1$: The use of light hormone produces a correlation between the standard deviation of the light level and the packet count.

Table 6.3 shows the results of Spearman's correlation coefficient run using R's *cor.test()*[26] function on the standard deviation and number of data points per hour. This forms a two tailed test which does not assume that the underlying data is normally distributed.

Table 6.3: Correlation between the standard deviation of the environmental light and the number of packets received per hour.

Experiment	Correlation	p.value	N	Alpha (two-tailed)
E3	-0.03	0.0377	6054	0.025
E4	-0.03	0.005	12599	0.025
E5	-0.08	<0.001	6392	0.025
E6	-0.07	<0.001	16719	0.025
E7	0.55	<0.001	2189	0.025
E8	0.37	<0.001	3682	0.025
E9	0.39	<0.001	4837	0.025

Given the data in Table 6.3 we can accept $H5$ for experiments E7, E8 and E9. The light hormone produces a positive correlation between the standard deviation of the light level and the amount of data transmitted even with the addition of the centre and wind hormones.

There is a very weak negative correlation in experiments E3, E4, E5 and E6. These experiments do not use hormones that increase or decrease the sensing rate. Instead they use a fixed sensing rate. As such, there is no assumption that there would be a correlation. With the two-tailed alpha value of 0.025 all of the correlations were significant, except for the control experiment. There is little to no correlation between the standard deviation of the environmental light and the number of samples taken in the experiments E3, E4, E5 and E6. Experiments E7, E8 and E9 all show moderate correlation. Their associated p values indicate that the correlations can be strongly accepted as significant.

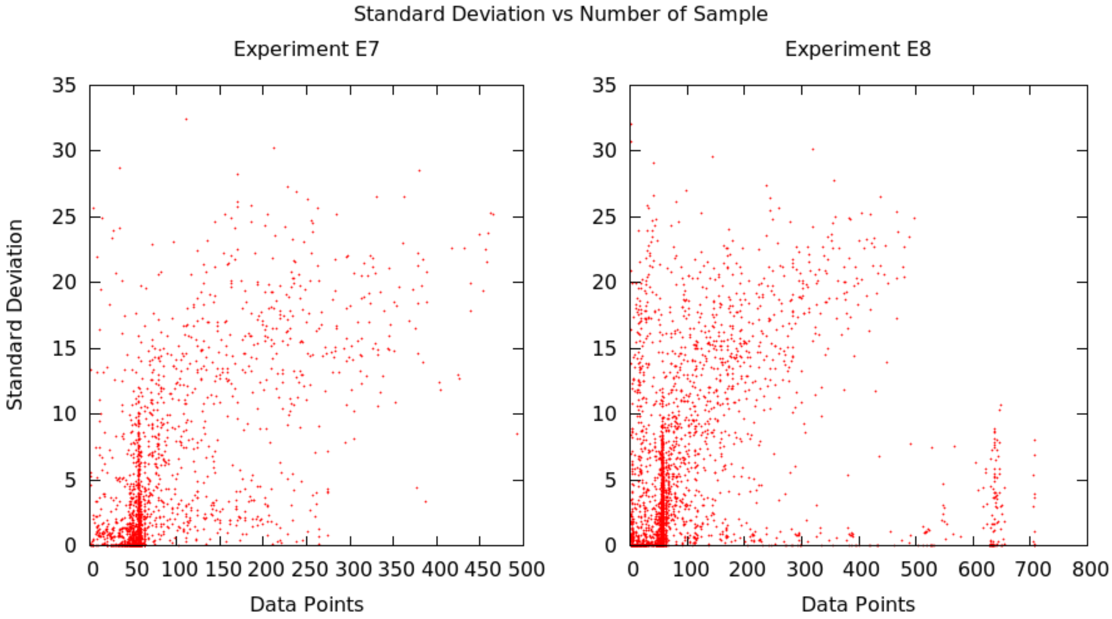

Figure 6.9: Standard deviation of light calculated over an hour plotted against number of packets received in that hour for experiment E7 and E8.

Experiment E5, which introduced the light hormone, shows the strongest correlation, 0.55, between the standard deviation and number of data points. The correlation is positive, which indicates that as the standard deviation increased, the number of data points also increased. The centre hormone experiment, E8, also shows a moderate positive correlation. Figure 6.9 shows the standard deviation plotted against the number of data points per hour for the light hormone, E7, and centre hormone, E8, experiments. It is possible to observe the positive correlation in both experiments. The lower correlation during the centre hormone experiment, may be due to the centre hormone's suppression and promotion mechanism. The centre hormone was used to promote sensing rate at night, when there was no change in light level, and suppress during the day, when there were changes in light level. If the hours during which centre hormone was present in the network were filtered out, the correlation between the standard deviation of the light level and number of data points increases to approximately 0.51. The associated p-value, 1.26×10^{-187}, is considerably below the alpha threshold of 0.025 and indicates that the correlation is significant. Therefore, the centre hormone, as used in the centre hormone experiment lowered the data quality. This was to be expected as it was not used to promote sensing rate during times of increased light variation or suppress during times of no light variation.

Figures 6.9 show that both experiments E7 and E8 produce the same clustering of data, at around 60 data points, that was present in experiments E3, E4, E5 and E6. This clustering could be explained by the fact that at night and at low light levels, the changes in light level may not have been large enough to be detected by the light hormone mechanism. As a result, the default sensing rate of 60 packets per hour was maintained during these periods. The centre hormone experiment, E8, shows additional clusters at 650 and 700 data points, in the same manner as experiment E6 which used a faster sensing rate. This is the result of promoting the sensing rate using the centre hormone, which maximised the sensing rate for around 7 hours on each occasion. There is, therefore, a number of hours during which the sensing rate of the network was very high without regard for the light conditions.

The last hypothesis tested in this section is $H6$.

> $H6_0$: The use of wind hormone does not produce a correlation between the standard deviation of the wind speed and direction and the packet count.
> $H6_1$: The use of wind hormone produces a correlation between the standard deviation of the wind speed and direction and the packet count.

Experiment E9, the wind hormone experiment, used an endocrine inspired system to promote and suppress sensing rate according to wind conditions. Figure 6.10 shows the standard deviation of the light, wind speed and change in wind direction plotted against the number of data points per hour for experiment E9. Table 6.4 shows the correlation between these two variables. The correlation between the standard deviation of the light and the number of data points, 0.39, is very similar to that of the centre hormone, 0.37. This suggests, that the improvement in data quality due to the light hormone mechanism is not negatively impacted by the addition of the wind hormone. The correlation between the wind speed standard deviation and number of data points is very similar to that of the light, at 0.39. This implies that as the light or wind speed becomes increasingly variable, the sensing rate of the network also increases.

Table 6.4 shows that there is almost no correlation between the standard deviation of the change in wind direction and the number of data points per hour. The wind hormone mechanism used the standard deviation of the changes in wind direction over a few minutes

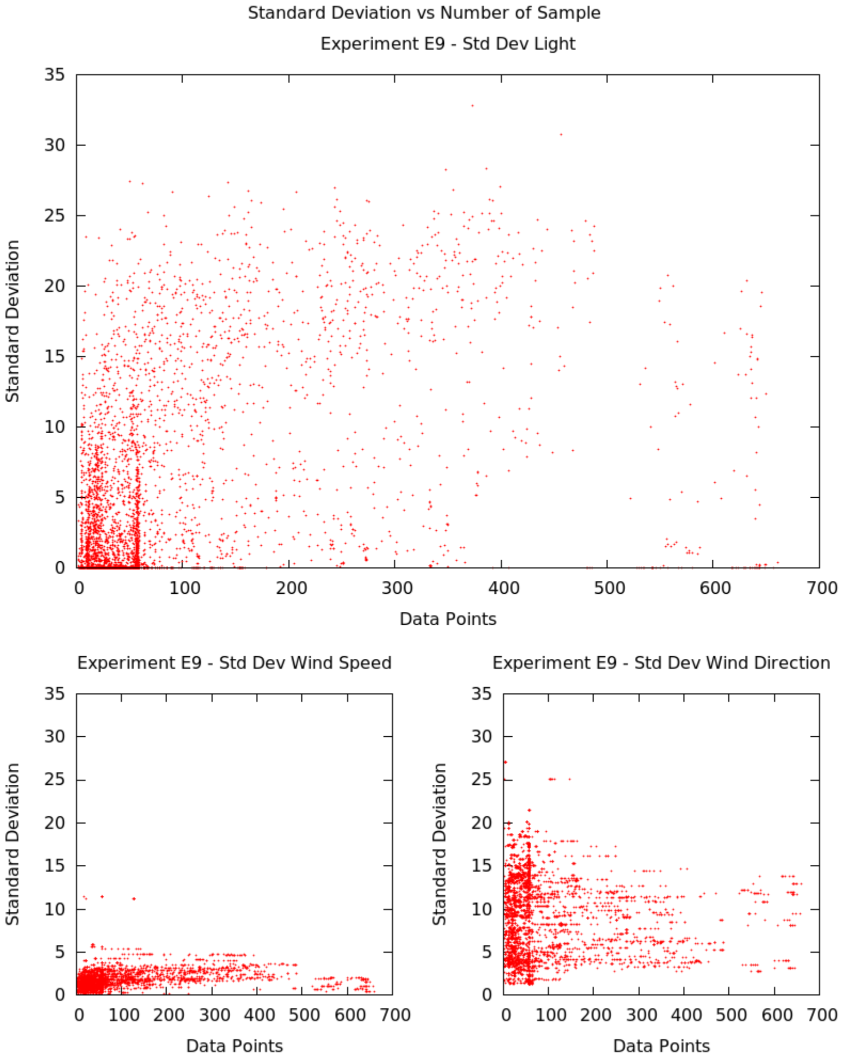

Figure 6.10: Standard deviation of the light, wind speed and change in wind direction calculated over an hour plotted against number of packets received in that hour for the wind hormone experiment, E9.

to produce a suppression hormone. A low standard deviation implied that the wind direction was stable and that the sensing rate should be suppressed as a result. The threshold at which the production of suppressing hormone occurred was a standard deviation of less than 5. This can be seen in Figure 6.10, where there were almost no examples of a high numbers of data points with a standard deviation of change in the wind direction below 5.

We cannot accept $H6$. There is evidence of a correlation between the standard deviation in wind speed and the amount of data transmitted by the network. However Table 6.4

Table 6.4: Correlation between the standard deviation light, wind speed and change in wind direction and the number of packets received per hour for the Wind Hormone experiment, E9.

Experiment	Parameter	Correlation	p.value	N	Alpha (two-tailed)
E9	Light	0.39	<0.001	4837	0.025
E9	Wind Speed	0.39	<0.001	4837	0.025
E9	Wind Direction	-0.04	0.002	4837	0.025

shows that there is no correlation between the standard deviation of wind and amount of data transmitted.

The effect of the centre hormone is visible in all three of the plots in Figure 6.10 with small clusters around 550 and 650 data points. The clustering occurred at a lower number of data points than in previous experiments. This is possibly due to the suppressing effects of the wind hormone. The clustering around 60 data points that was present in previous experiments appeared to be less concentrated and spread over the range 1 to 60 data points. This, again, was likely to be due to the suppressing effect of the wind hormone, it was unlikely that there were many periods of the default sensing rate of 60 packets per hour.

6.5.2.1 Standard Deviation Conclusion

As a metric, the correlation between standard deviation of light level and the number of data points per hour is effective in showing which experiments returned better quality data. The power focussed experiments E4, E5 and E6 all performed similarly showing little to no correlation. There appeared to be little to no degradation in the data quality, using this metric, as a result of the selfish or anger hormones. The light hormone showed a good correlation, suggesting that, with its addition, the quality of data produced by the network was higher than experiments that did not use the light hormone.

The addition of centre hormone, in experiment E8, resulted in a lower correlation than the light hormone experiment. This was likely due to promotion of the sensing rate at night and suppression during the day. This affected the sensing rate and therefore the amount of data generated but in a manner that did not correspond with any environmental parameter. When the hours affected by the centre hormone were removed and the correlation re-run, the new correlation was very similar to that of the experiment E7 which just used the light hormone.

The addition of the wind hormone lowered the correlation for light level, but showed a similar correlation for wind speed. The change in wind direction showed very little to no correlation. It did, however, produce the expected effect of suppressing sensing rate during periods of stability in wind direction.

Increasing sensing rate during periods of high variability is advantageous in improving the quality of the data. However, increasing the sensing rate too much, or unnecessarily would result in duplicate data being recorded. This duplicate information is of little use as it provides no new information. The next metric analyses experiments E7, E8 and E9 in terms of the amount of duplicate data produced.

6.5.3 Duplicate Data

The duplicate data metric described in the Metrics section of this chapter is used to analyse the effects of light, centre and wind hormones on duplicate data. Having no duplicate data may be the result of too infrequent sensing, however, this cannot be determined by this metric alone. A large amount of duplicates indicates that the sensing rate is too high. Figure 6.11 shows a histogram for the experiments E5, E6, E7 and E8. Plots for the remaining experiments can be found in Appendix D. The frequency of the number of duplicate packets per hour for each experiment is shown. In every experiment, the most frequent number of duplicates per hour is 0 followed by one then two duplicates per hour. The figures for experiments E5 and E7 are representative of the results of E3 and E4. Experiments E6, E8 and E9 showed a different distribution. These three experiments produced as many as 10 to 12 duplicates in an hour. The experiment with by far the most duplicates is E6. Experiment E6 did not use any hormones to increase its sensing rate, just the selfish and anger hormones to improve power consumption. However, it did use a fixed increased sensing rate of 11 packets per minute.

Calculating the number of duplicates as a percentage of the number of packets per hour presents a different view. One duplicate packet, with a sensing rate of 60 packets per second, results in a duplicate data point percentage of $\frac{1}{60} \times 100 = 1.67\%$. Experiment E6 transmitted data 11 times more frequently. Twelve duplicate packets in an hour, therefore, results in $\frac{12}{660} \times 100 = 1.81\%$ duplicate data points. As a percentage the number of duplicate

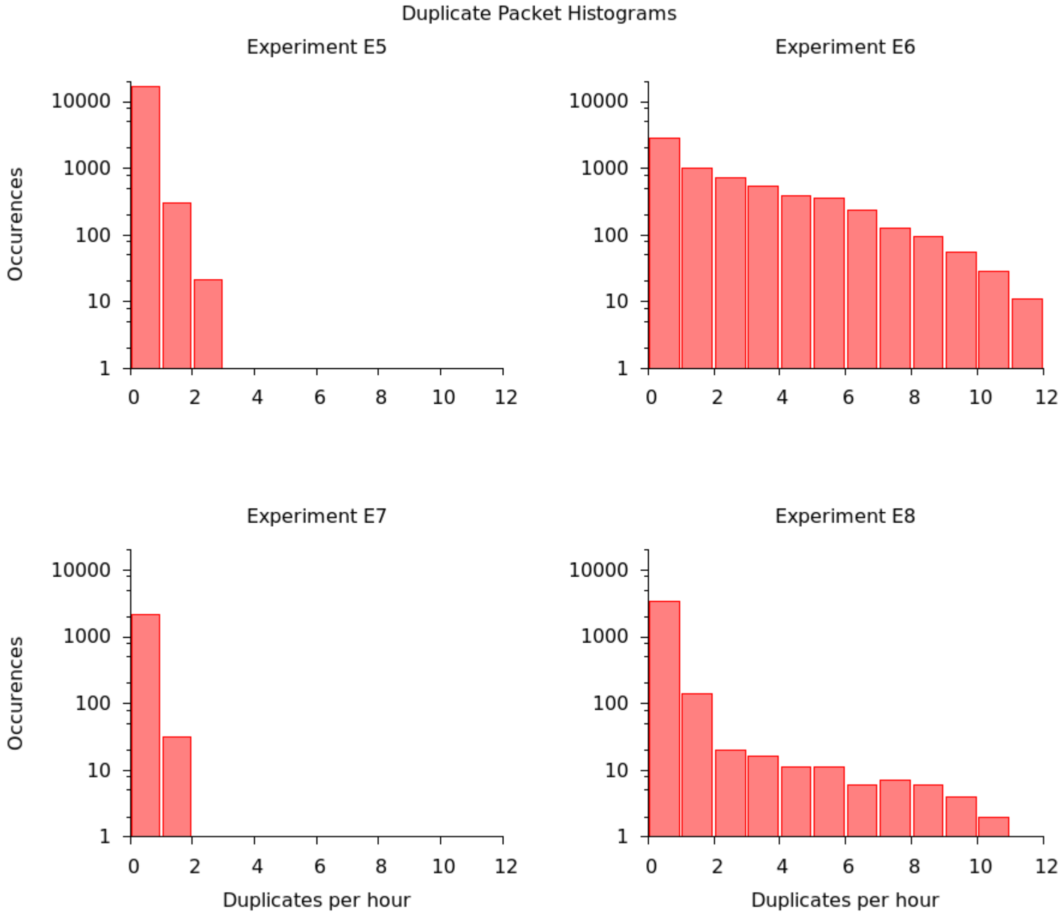

Figure 6.11: Histograms showing the frequency of duplicate data per hour for experiments E5, E6, E7 and E8. Plotted using a log scale. The most frequent number of duplicates is 0 in every experiment. Appendix D contains plots for the other experiments.

data points for experiment E6 is not much higher than the preceding experiments that used the lower sensing rates of one packet per minute. However, while the percentage does not increase by much, the actual number of duplicate data points does. Experiment E5, using the normal sensing rate of 60 packets per hour, produced 344 duplicate sequential data points over the duration of the experiment, 44 days. When the sensing rate was increased elevenfold in E6, the number of duplicate sequential data points produced was 11505, over the 18 days the experiment ran. Assuming the distribution of duplicates is uniform then in 18 days the experiment E5 would have produced 140 duplicates, 82 times less than E6. The fixed increase in sensing rate produced large quantities of redundant data.

The light, centre and wind hormones are all capable of increasing the sensing rate of nodes in the network to the same level as the high sensing rate anger hormone experiment. However, these hormones are produced in response to the environmental conditions, in particular when the environment was changing. The histogram in Figure 6.11 for experiment E7, which made use of the light hormone, shows a similar distribution to previous experiments. However, the sensing rate during the experiment was significantly higher during the day, than previous experiments.

The histograms for experiments E8 and E9 show instances of hours with up to 11 duplicates. The number of hours with high duplicate rates is considerably lower than experiment E6. The use of the centre hormone to promote sensing rate is considered to have resulted in an increase of duplicates when compared to the light hormone experiment, E7. This is due to the environment changing more slowly at night, particularly due to the lack of light. These observations support hypothesis $H7$:

> $H7_0$: Experiments using the light, centre and wind hormones will not produce fewer duplicate data packets when compared to a fixed increase in sensing rate
> $H7_1$: Experiments using the light, centre and wind hormones will produce fewer duplicate data packets when compared to a fixed increase in sensing rate.

Figure 6.12 shows the number of duplicate data points per hour from E6. There is a clear diurnal pattern with increased numbers of duplicates data at night. Promoting sensing rate at night using the centre hormone, as was carried out in experiments E8 and E9, resulted in a similar peak for that particular night of the experiment. The presence of higher numbers of duplicate packets in the centre and wind hormone experiments is due to the centre hormone increasing the sensing rate at night time.

6.5.3.1 Duplicate Data Conclusion

The duplicate data points metric provided insights into the amount of redundant data produced by the sensor network. Most duplicate data points were recorded during night time, which is the result of the environment changing more slowly during this time. In particular, the light level and battery voltage would change more slowly due to the lack of solar input. The method of detecting duplicate data is a very conservative method. It looks for identical sets of measurements and the sensors have some degree of noise in their outputs and are high resolution. Determining acceptable bounds, however, is difficult and

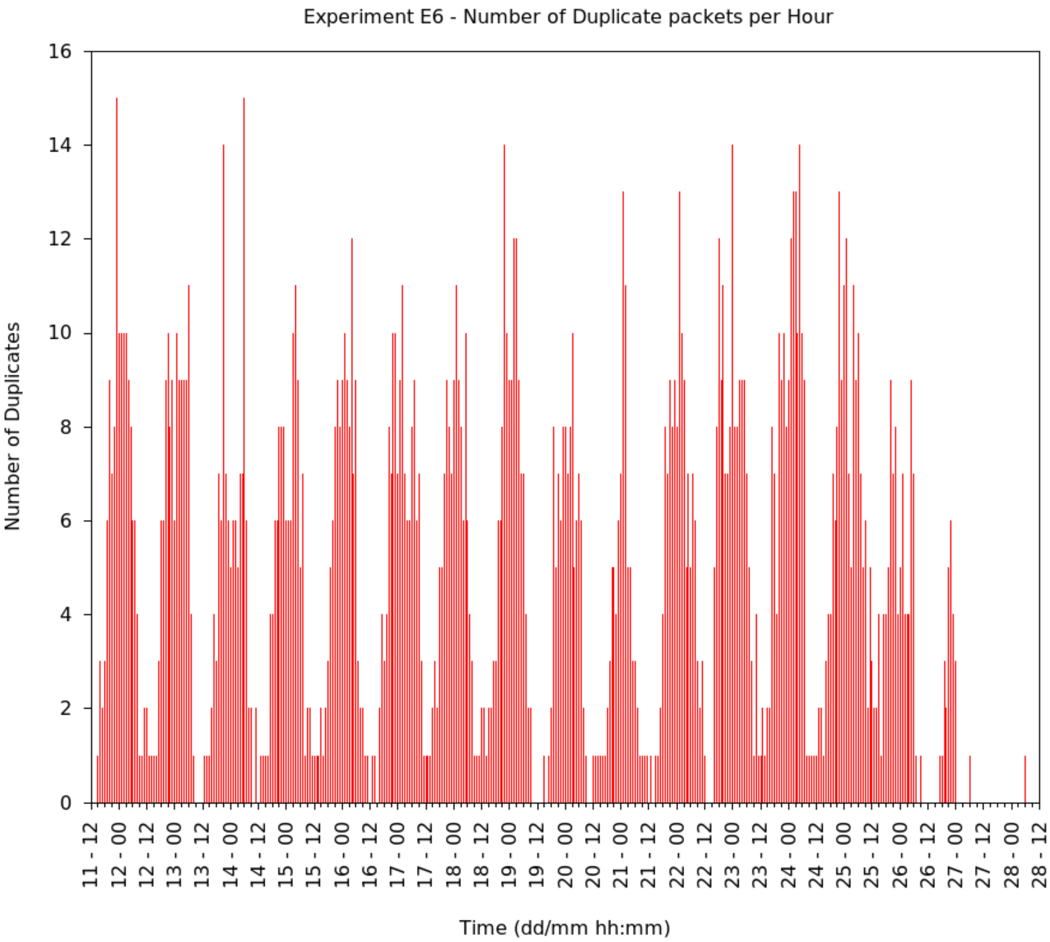

Figure 6.12: Number of sequential duplicate packets per hour during experiment E6.

would have to be justified on a sensor by sensor basis. Differentiating between genuine gradual change and noise would be difficult. Regardless of the conservative nature of the values produced by this metric, it provided a good indicator that a sensing rate is too high.

Increasing the sensing rate to a higher fixed level, as seen in experiment E6, resulted in unnecessary sensing during periods of slow or no environmental change. By modifying the sensing rate more intelligently, such as in experiments E7, E8 and E9, the amount of data could be increased while producing less duplicate data than a fixed increase in sensing rate. The centre hormone produced an increase or decrease in the amount of duplicate data by increasing or decreasing the sensing rate. By promoting the sensing rate at night more duplicate data was obtained. The trade-off between higher temporal resolution and unnecessary data may be acceptable if there was some particular event occurring. On a day to day basis, however, it may not.

6.5.4 Power Analysis of Data Experiments

The experiments detailed in this chapter were designed to explore the effects of endocrine inspired control methodologies on data quality. However, given the methods used to attempt to modify data quality, their effect on power consumption is examined. The metric used to analyse the power usage in the previous chapter fitted a line of best fit using R's *rlm()*[26] function to the amount of light received by each node integrated over a day and the change in battery voltage over the same day. The naïve model detailed in Section 5.5 suggested that an increase in steepness of the line fit indicated a better ability to adapt to the available power. Table 6.5 shows the results of running a Spearman's correlation between the integrated light and change in voltage for the three data quality experiments. Table 6.6 shows the slope of the line fit and average absolute error of residuals for each experiment. Section 5.6 describes the process for the calculating the correlation coefficient and fitting of the best fit line in more detail.

Table 6.5: Spearman's rank correlation coefficient for integrated light and change in voltage for the three data quality experiments.

	Spearmans Rho	N	Alpha	P Value	Reject/Accept H_0
E7	0.2816227	42	0.05	0.03539595	Reject
E8	0.7045392	105	0.05	<0.001	Reject
E9	0.4112632	160	0.05	<0.001	Reject

The correlations and p-values in Table 6.5 allow us to accept $H1$ that there is a correlation between integrated light and change in battery voltage for experiments E7, E8 and E9. The plots and fitted linear models for experiments E7, E8 and E9 are shown in Appendix A.

Table 6.6: Slopes of the line fits and average absolute error for the data quality experiments.

Experiment	Slope ($\times 10^5$)	Average Absolute Error of Residuals ($\times 10^5$)
E7	77.30	5.11
E8	46.02	3.75
E9	28.27	3.74

$$\text{E7} = \frac{77.30}{11.96} = 6.46 \text{ times steeper than the control}$$
$$\text{E8} = \frac{46.02}{11.96} = 3.85 \text{ times steeper than the control}$$
$$\text{E9} = \frac{28.27}{11.96} = 2.36 \text{ times steeper than the control}$$

Of all the experiments run, the light hormone experiment has the steepest slope, see Table 6.6, almost 6.5 times higher than the control experiment. This may be due to the short duration of the experiment and the associated higher error in the linear model fit. Another possibility was that the light experiments increased sensing rate, and therefore power consumption, during the day, providing the ability to consume more energy during the day when more energy was available. The centre hormone experiment provided the ability to suppress and promote the sensing rate which may have hindered the ability of selfish hormone to regulate energy consumption. The slope of the centre hormone was still steeper than that of the anger hormone with the normal sensing rate. This may have been due to the presence of the light hormone, which increased daytime power consumption. Finally, the wind hormone experiment showed a shallower slope than the even the selfish hormone experiment. The wind hormone provided the ability to promote and suppress sensing rate according to wind conditions. This, in conjunction with the light hormone and centre hormone, may have had detrimental effects on the power usage, by over suppressing the network when it was not necessary.

6.6 Summary

In this chapter three experiments using three different endocrine inspired methods of improving data quality were presented. Firstly, the light hormone was introduced which provided a mechanism for a node experiencing large changes in light level to increase the sensing rate of itself and neighbouring nodes. As the light level is prone to changing quickly and by large amounts due to cloud cover or obstruction by foliage this aimed to produce higher resolution data at these times. By analysing the number of packets produced by the network we were able to determine that the light hormone was successful in increasing the sensing rate of the network. In addition it was shown that the light hormone produced a positive correlation between the standard deviation of the light level and the amount of data produced. This suggested that the quality of data was higher as the temporal resolution increased due to the light level becoming more variable.

In the next experiment the centre hormone was introduced, which provided a user with the ability to promote or suppress the sensing rate of the whole network from one location, for a period of up to 12 hours. Clear evidence was found suggesting that the centre hormone was effective in modifying the sensing rate. However, due to the nature of broadcast transmissions and the duty cycle approach, some nodes did not receive the hormone and remained unaffected. Most nodes, however, did receive the hormone and their sensing rates were successfully suppressed or promoted for 12 hr periods. Importantly the effects of the centre hormone were combined with the effects of the other hormones used by the nodes. This allowed multiple behaviours to mix in a simple and intuitive way. Lowering the sensing rate using the centre hormone mixed with the increase in sensing rate produced by the light hormone. Thereby producing the same modulation of sensing rate as just the light hormone but at an offset. Using the packet count metric the centre hormone was shown to be effective in increasing and decreasing the sensing rate.

Lastly, a ROWind wind speed and direction sensor was added to the base station and used to produce a wind hormone. The wind direction and wind speed were combined in an antagonistic manner at the base station and the resulting wind hormone transmitted throughout the network on an hourly basis. The wind hormone had the ability to increase or decrease the sensing rate of a node for a period of up to one hour. The effects of the wind hormone combined with those of the light and centre hormones to produce complex combinations of behaviours. There was no correlation between the change in wind direction and the amount of data produced by the network. The change in wind direction threshold chosen may have been too low as the network seemed to be suppressed very regularly. However, the correlation between the standard deviation of the wind and the number of data points was good. The correlation for the light level remained good, 0.39, with the addition of the wind hormone. This suggested that the wind hormone did not negatively impact the network by being added. The wind hormone succeeded in promoting and suppressing the sensing rate of the network.

Experiments were run on a wireless sensor network consisting of 20 sensor nodes arranged in a partially connected mesh with predefined routes. Due to equipment failure and potentially a software bug nearly half of the sensor nodes ceased operating after only a few days of deployment. Metrics were chosen that analysed the experiments on an hour by hour basis so as to minimise the impact of these equipment failures.

The work in this chapter shows that endocrine inspired approaches to improving power consumption did not reduce the quality of the data. There was not a significant decrease in packet counts or duplicate data and no change in correlation between standard deviation of the environmental light and the number of packets hourly. In addition, the data quality experiments presented in this chapter did not adversely affect the power consumption and adaptation of the sensor network. There remained a good correlation between the integrated light and the change in voltage for these experiments. The slopes of the fitted linear model in experiments E7 and E8 were higher than those of experiment E5. Experiment E5 only used the selfish and anger hormones. This suggests that the power usage was, in fact, improved by the light and centre hormones. This is thought to be due to these hormones being able to increase the sensing rate during the day. This allowed the nodes to consume more power, by sensing and transmitting more frequently thus using more energy when it was available. The linear model did appear to become shallower as more data quality hormones were introduced. The short length of the light hormone experiment and associated higher error may have resulted in this.

Analysis of the experiment with a fixed increase in sensing rate suggested that using a fixed increase in sensing rate resulted in large quantities of duplicate data. This stresses the network unnecessarily and produces data of little value. The potential for unmanageable quantities of data suggested that more intelligent modification of sensing rate would be advantageous. The hormone inspired approaches to improving data quality yielded a better correlation between number of samples and the variability of the environmental light. This produced less duplicate data than a consistently high sensing rate.

Chapter 7

Conclusions and Future Work

This chapter reviews the work presented in the preceding chapters and provides a discussion of the key contributions. Lessons learnt over the course of the work are discussed and future work outlined.

7.1 Power Management

The two endocrine inspired power management methods that were implemented resulted in improvements in power usage and node lifetime in the wireless sensor network used. The first of these approaches, the selfish hormone, allowed a node to modify how much time it spent waiting to service routing requests based on the available power stored in its battery. The second approach, the anger hormone, provided a method for nodes to modify the energy consumption of other nodes in order to improve their ability to successfully transmit data. The selfish hormone approach was tested on both a small 5 node network and a larger 20 node sensor network. An improvement in node lifetime was observed between the control and selfish hormone experiments in the smaller network. To analyse how well the available power was used in any particular experiment for each node the amount of light received in a day was plotted against the change in voltage over the same day. The naïve model developed in Section 5.5 showed that an increase in slope of a linear model fitted to this data would be indicative of a better adaptation to the available power. Both the selfish and anger hormone experiments yielded significantly steeper slopes. The Anger hormone was also successful in helping nodes with communication problems to increase the amount of data they were able to transmit to the base station. In particular three nodes with very poor connectivity showed a significant increase in the number of packets received over the first week of the experiment when compared to the control experiment.

7.2 Data Quality

To examine the effects of endocrine inspired control methodologies on the data quality from a wireless sensor network, three additional hormone systems were developed. The light hormone was generated in response to rapidly changing light levels and provided a mechanism for promoting and suppressing the sensing rate of a node and its neighbours. The centre hormone provided a hormone-like way for users to interact with the network. Through the release of the centre hormone, the sensing rate of the whole network could be promoted and suppressed. Lastly, a wind hormone generated in response to wind speed and change in wind direction enabled the suppression and promotion of the network's sensing rate in response to wind conditions. Analysis of the amount of data from each experiment showed that a flat out increase in sensing rate, such as in the high sensing rate anger hormone experiment, resulted in large quantities of redundant data being collected and transmitted by the network. The light hormone improved the amount of data received during the day and also showed that as the standard deviation of the light increased so too did the sensing rate; suggesting that the hormone based rapid light change detector was functioning correctly and that the light hormone was being distributed and used by neighbouring nodes. Analysis of the centre hormone experiment showed that it was possible to control, to some extent, the behaviour of the network in an endocrine inspired way. An increase in the amount of data was seen when the centre hormone was promoting the sensing rate and the inverse was true when the centre hormone was suppressing the sensing rate. Occasionally some nodes would not receive the broadcast containing the centre hormone and as a result did not modify their behaviour. The wind hormone experiment showed a good correlation between the standard deviation of the wind speed and the amount of data collected by the sensor network. There was no correlation between the standard deviation of the change in wind direction and the amount of data collected, however there was a noticeable suppressing effect when the standard deviation of the change in wind direction was low. The lack of correlation may imply that the change in wind direction threshold used to generate wind hormone was incorrect.

7.3 The Endocrine System and Wireless Sensor Networks

Throughout this work, hormones were added to the wireless sensor network platform one at a time. By the time the last experiment was run, there were five endocrine inspired systems all working simultaneously to affect the behaviour of individual nodes and the network as a whole. These endocrine inspired ideas and the hormones that they produce were able to be combined whilst preserving the effects of each individual. The transmission of hormones throughout the network can be likened to the transmission of hormones through the body. Nodes represent various bodily tissues and glands, receptive to hormones and producing other hormones. Together these hormones, and the nodes that produce them, come together to maintain homeostasis.

7.4 Sensor Node Platform

The sensor node platform used worked well given the rapid rate of development, testing and experimentation. Having the environmental sensors and microcontroller integrated into one circuit board was both good and bad. Due to the microcontroller used, there were almost no free I/O pins. As the sensors couldn't be removed this meant that adding or swapping sensors was not an option. The most problematic sensor was the light sensor. To measure the environmental light it needed to be exposed to light. However, the node case was not transparent. This required the use of an LED 'light pipe' to channel the light from outside to the sensor. These issues were balanced by the fact that having the sensors mounted on the microcontroller board meant there were never issues with bad connections or with reading a sensor.

The boards were remarkably robust and, with a coat of conformal coating, all but one board survived the full duration of all experiments. The boards were subjected to some truly awful weather; gale force winds, torrential rain, prolonged periods of high humidity, particle and fauna ingress and frequent temperature cycling.

The microcontroller and sensor board was very cost effective, only costing £75 per unit. A major problem, however, was availability as a year into the work presented, the boards ceased to be manufactured and are now no longer available.

The biggest source of issues with the microcontroller and sensor board was that it was primarily designed as a hobbyist device for measuring the weather. It was not really designed with wireless sensor networks in mind. As a result there were a few features that were missing such as a stable and accurate oscillator and some local storage. A better oscillator would have enabled more precise time-keeping and potentially some form of synchronisation. Some small quantity of local storage would have provided a buffer for data and the ability to store data until transmission was possible. Power consumption was also higher than was desirable as the electronics were not designed with very low power consumption in mind. While it performed admirably and survived some harsh conditions for over 6 months, a more specialised wireless sensor network platform would be preferable in the future.

The XBee wireless transceiver used in the sensor nodes worked well. The range achieved, considering the 2.4 GHz frequency used, was much greater than expected although it did not cope with obstructions very well. The modules were very flexible and there were several different firmware versions available to support different protocols and systems. However, getting the modules to behave in ways that were unanticipated by the developers was impossible. The firmware was not available and so could not be modified to add or modify the behaviour of the devices. The method of communication with the microcontroller, logic level serial, used up the microcontroller's only hardware serial port. This would normally be used for sending debug information during development and so this further complicated debugging and development. The only problem with the physical design of the connector was that the U.FL antenna connection provided poor mechanical strength and was not designed for repeated insertion. There are XBee modules with alternative antenna connectors however these were not available at the time that development was taking place.

The Arduino development environment enabled very rapid development and made finding libraries for various pieces of hardware much easier. Being lightweight, multi-platform and free meant that it could be installed on a number of devices for making changes in the field. The Arduino environment provides a large number of libraries and utility functions. One issue with this was that it took a significant amount of time to fully understand how this code worked and the implications of using it.

7.4.1 Mechanical Hardware

The node enclosures worked well, protecting the node from wind and rain whilst allowing air to flow across the sensors. The cases sustained only very slight damage such as scratches in the paint and a few small dents. The solar panel mounts were strong enough to withstand strong wind conditions and there was almost no water ingress into the inner protective box. The size and awkward shape of the enclosures made carrying more than 4 nodes very difficult. This greatly increased the amount of time required for deployment and retrieval of the nodes, as multiple trips were needed. While the lead acid batteries used were very robust to a wide range of temperatures and charging conditions their weight made carrying nodes hard work. The louvred vents used to provide ventilation and protection from the elements worked well. The addition of some mesh behind the vents would have perhaps stopped the ingress of some of the larger fauna that tried to take up residence inside the enclosures.

Whilst not a problem in a longer deployment, multiple shorter deployments required repeated disassembly of each node. Each node required 8 screws to be undone to access the electronics inside which took significant amounts of time across 20 nodes. Finally the mounting system worked very well. The clamping screws could be tightened by hand and would still provide a solid connection to the scaffold tube a node was mounted to. Attaching or detaching nodes to their mounting posts was very quick and nodes could be oriented to a particular direction easily.

7.5 Key Contributions

The novel contributions that have been made by this work are as follows:

- Multiple endocrine inspired components can be combined while retaining the behaviour of each individual component.(Chapters 5 and 6).
- The ability of a wireless sensor network to adapt power consumption can be improved through the use of endocrine inspired systems when compared to a simple non adaptive control system. (Chapter 5).
- The quality of data returned from a wireless sensor network can be improved through the use of endocrine inspired systems. (Chapter 6).

- The power adaptation and data quality can be improved simultaneously through the use of endocrine inspired systems. (Chapter 4 and 6).

- Deployment of several endocrine inspired control methodologies on a real world sensor network. (Chapters 3, 4, 5 and 6).

- Development of a metric to analyse how well the energy available to a wireless sensor network was used. (Chapter 5).

7.6 Limitations and Lessons

The endocrine inspired methods used in the work presented in this thesis rely on the hardware, software and design of the sensor network platform providing control over the power consumption and sensing rate.

Although significant effort was put in to ensuring each experiment was run in the same manner, there is no way to control the environment or weather. As a result, any replicate experiments would have to be run under different conditions.

Over the course of this work several important lessons were learnt. The importance of a reliable base station was under estimated. As there was no other mechanism for recording what individual nodes and the network as a whole were doing a failure in the base station resulted in the complete loss of data. The base station software and hardware was made more robust over time to prevent the loss of data.

Another important lesson was that deployments in real world are time consuming, expensive and difficult. The amount of time required to halt an experiment and set up another was around 8-10 hours. This necessitated waiting for days with good weather to change experiments. Cost of getting to and from the deployment site was also non-trivial despite being relatively close. Deployments further afield with a greater number of nodes would pose a significant challenge and take several days to complete.

Development and debugging of the node platform itself was also very challenging. The hardware used makes the use of many existing debuggers and analysers impossible and once multiple nodes become involved, tracking down issues and bugs becomes very time consuming. When changing experiments in the field, it was essential to bring a large quantity of tools, laptops and materials for dealing with any issues that arose. The importance of remote management was also made clear when the base station program crashed. Being able to log in remotely to fix issues would have saved time and data.

One component that was particularly prone to damage was the radio antennae. Heat cycling of the antennae caused fatigue in certain metal parts which led to breakages and bad connections. In addition the antennae proved to be tempting perches for the local Red Kites and a few were damaged by sheep and horses.

7.7 Future Work

There are a number of potential avenues that would be worth investigating in the future.

Investigating the use of power hormones that work over longer time scales than the selfish hormone presented in this work. These hormone(s) could promote and suppress the power consumption of individual nodes or the network as a whole and would be generate in response to some long term average. This may enable nodes to only moderately increase their power consumption on an uncharacteristically energy abundant day during a period of low power availability. It may also prevent unnecessary low power consumption on a particularly poor day during periods of high energy availability. Some degree of hysteresis could provide the network with time to recover from periods of low energy availability.

Other aspects of the endocrine system, such as negative feedback, hormone cascades and hormone sensitivities would be an interesting avenue for further research.

Better management of power would also be worth investigating. The charging system used in this work directly charged the batteries from a solar panel. Such frequent charging and the potential for over-charging is likely to reduce the batteries useful lifespan. By using some more intelligent charging system, such as a purpose built solar charging IC would help reduce the wear on batteries.

Lowering the power consumption of nodes to levels similar to other sensor network platforms could provide interesting data on how endocrine inspired systems can benefit very low power systems. In a similar vein, moving to a hardware platform better suited to very low power consumption could be beneficial. Bringing the endocrine inspired systems into more aspects of network and individual nodes behaviour has the potential to improve performance in these areas. The development of a wireless sensor network where control of all aspects of operation was given over to endocrine inspired systems would provide a fascinating platform to study.

Performing a comparative study, comparing the methods described and analysed in this work to other biologically inspired wireless sensor work would be well worth pursuing although performing a valid comparison using real world deployments may be difficult.

The work presented in this thesis has attempted to stay as close to a real world wireless sensor network scenario as possible. Taking these endocrine approaches and using them to develop wireless sensor network solutions to real world problems would provide numerous fields with valuable data.

Bibliography

[1] K. Akkaya and M. Younis. A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, 3(3):325–349, 2005.

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. *Computer Networks*, 38:393–422, 2002.

[3] Arduino. Arduino Homepage. `https://www.arduino.cc/`, 2015. Accessed: 2016-04-15.

[4] R. C. Arkin. Homeostatic control for a mobile robot: Dynamic replanning in hazardous environments. *Journal of Robotic Systems*, 9(2):197–214, 1992.

[5] R. C. Arkin. Survivable robotic systems: Reactive and homeostatic control. In *Robotics and remote systems for hazardous environments*, pages 135–154. Prentice-Hall, Inc., 1993.

[6] O. Avila-García and L. Cañamero. Using hormonal feedback to modulate action selection in a competitive scenario. In *From Animals to Animats: Proceedings of the 8th International Conference of Adaptive Behavior*, pages 243–252, 2004.

[7] S. Bapat, V. Kulathumani, and A. Arora. Analyzing the yield of ExScal, a large-scale wireless sensor network experiment. *Proceedings of the International Conference on Network Protocols, ICNP*, 2005:53–62, 2005.

[8] G. Barrenetxea, F. Ingelrest, G. Schaefer, and M. Vetterli. The hitchhiker's guide to successful wireless sensor network deployments. In *Proceedings of the 6th ACM conference on Embedded network sensor systems - SenSys '08*, pages 43–56, 2008.

[9] G. Barrenetxea, F. Ingelrest, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange. SensorScope: Out-of-the-box environmental monitoring. *Proceedings of the 2008 International Conference on Information Processing in Sensor Networks, IPSN 2008*, pages 332–343, 2008.

[10] E. a. Basha, S. Ravela, and D. Rus. Model-based monitoring for early warning flood detection. In *Proceedings of the 6th ACM conference on Embedded network sensor systems - SenSys '08*, pages 295–308, 2008.

[11] J. Beauquier, P. Blanchard, J. Burman, and S. Delaët. Tight complexity analysis of population protocols with cover times - The ZebraNet example. *Theoretical Computer Science*, 512:15–27, 2013.

[12] C. Bernard. Lessons on the phenomena of life common to animals and vegetables. *Bailliere et Fils, Paris*, 1878.

[13] S. Bhatti, J. Carlson, H. Dai, J. Deng, J. Rose, A. Sheth, B. Shucker, C. Gruenwald, A. Torgerson, and R. Han. MANTIS OS: An embedded multithreaded operating system for wireless micro sensor platforms. *Mobile Networks and Applications*, 10(4):563–579, 2005.

[14] P. Boonma, P. Champrasert, and J. Suzuki. BiSNET: A Biologically-Inspired Architecture for Wireless Sensor Networks. In *Proceedings of the 2006 International Conference on Autonomic and Autonomous Systems*, pages 54–54, 2006.

[15] P. Boonma and J. Suzuki. BiSNET: A biologically-inspired middleware architecture for self-managing wireless sensor networks. *Computer Networks*, 51(16):4599–4616, 2007.

[16] U. Brinkschulte, M. Pacher, and A. von Renteln. Towards an Artificial Hormone System for Self-organizing Real-Time Task Allocation. In *Software Technologies for Embedded and Ubiquitous Systems*, pages 339–347. 2007.

[17] U. Brinkschulte and A. von Renteln. Analyzing the Behavior of an Artificial Hormone System for Task Allocation. In Juan González Nieto, Wolfgang Reif, Guojun Wang, and Jadwiga Indulska, editors, *Proceedings of the 6th International Conference*

on Autonomic and Trusted Computing*, volume 5586 of *Lecture Notes in Computer Science*, pages 47–61, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[18] R. Brooks and P. A. Viola. Network based autonomous robot motor control: From hormones to learning. *Advanced Neural Computers*, pages 341–348, 1990.

[19] R. a. Brooks. Integrated systems based on behaviors. *ACM SIGART Bulletin*, 2(4):46–50, 1991.

[20] M. Buettner, G. V. Yee, E. Anderson, and R. Han. X-MAC. In *Proceedings of the 4th international conference on Embedded networked sensor systems - SenSys '06*, pages 307–320, New York, New York, USA, 2006. ACM Press.

[21] W. B. Cannon. Organization for Physiological Homeostasis. *Physiological reviews*, 9(3), 1929.

[22] Q. Cao and T. Abdelzaher. liteOS: A Lightweight Operating System for C++ Software Development in Sensor Networks. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, SenSys '06, pages 361–362, New York, NY, USA, 2006. ACM.

[23] Q. Cao, T. Abdelzaher, J. Stankovic, and T. He. The LiteOS operating system: Towards Unix-like abstractions for wireless sensor networks. *Proceedings of the 2008 International Conference on Information Processing in Sensor Networks, IPSN 2008*, pages 233–244, 2008.

[24] T. M. Cao, B. Bellata, and M. Oliver. Design of a generic management system for wireless sensor networks. *Ad Hoc Networks*, 20:16–35, 2014.

[25] R. C. Carrano, D. Passos, L. C. S. Magalhaes, and C. V. N. Albuquerque. Survey and taxonomy of duty cycling mechanisms in wireless sensor networks. *IEEE Communications Surveys and Tutorials*, 16(1):181–194, 2014.

[26] Austria Computing, R Foundation for Statistical Vienna. *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria, 2008.

[27] L. Crapo. *Hormones the Messengers of Life.* W.H. Freeman and Company, 1985.

[28] Crossbow. Mica 2 Datasheet. `http://www.eol.ucar.edu/isf/facilities/isa/internal/CrossBow/DataSheets/mica2.pdf`, 2005. Accessed: 2016-04-15.

[29] Crossbow Technology. Mica2dot Datasheet. `http://www.willow.co.uk/html/mpr5x0-_mica2dot_series.php`.

[30] Crossbow Technology. TelosB Datasheet. `www.willow.co.uk/TelosB_Datasheet.pdf`.

[31] Crossbow Technology. iMote 2 datasheet. `http://bullseye.xbow.com:81/Products/Product_pdf_files/Wireless_pdf/Imote2_Datasheet.pdf`, 2012. Accessed: 2016-04-15.

[32] M. F. De Castro, L. B. Ribeiro, and C. H. S. Oliveira. An autonomic bio-inspired algorithm for wireless sensor network self-organization and efficient routing. *Journal of Network and Computer Applications*, 35(6):2003–2015, 2012.

[33] J. Decotignie and J. Hernandez. Low Power MAC Protocols for Infrastructure Wireless Sensor Networks. *Mobile Networks and Application*, 10(5):675–690, 2005.

[34] G. Di Caro and M. Dorigo. AntNet: Distributed stigmergetic control for communications networks. *Journal of Artificial Intelligence Research*, 9:317–365, 1998.

[35] Digi. XBee/XBee-PRO DigiMesh 2.4 User Guide. `http://ftp1.digi.com/support/documentation/90000903_G.pdf`, 2014. Accessed: 2016-04-15.

[36] Y. Ding, Y. Jin, L. Ren, and K. Hao. An Intelligent Self-Organization Scheme for the Internet of Things. *IEEE Computational Intelligence Magazine*, 8(3):41–53, August 2013.

[37] D. Dong, H. You, Y. Zhang, and X. Wang. A hormone-based clustering algorithm in wireless sensor networks. In *Proceedings of the 2nd International Conference on Computer Engineering and Technology*, volume 3, pages 555–559, 2010.

[38] M. Dorigo and C. Blum. Ant colony optimization theory: A survey. *Theoretical Computer Science*, 344(2-3):243–278, 2005.

[39] F. Dressler, I. Dietrich, R. German, and B. Krüger. Efficient operation in sensor and actor networks inspired by cellular signaling cascades. In *Proceedings of the 1st international conference on Autonomic computing and communication systems*, volume 1, page 7, 2007.

[40] F. Dressler, I. Dietrich, R. German, and B. Krüger. A rule-based system for programming self-organized sensor and actor networks. *Computer Networks*, 53(10):1737–1750, 2009.

[41] A. Dunkels, B. Grönvall, and T. Voigt. Contiki - A lightweight and flexible operating system for tiny networked sensors. *Proceedings of the Conference on Local Computer Networks, LCN*, pages 455–462, 2004.

[42] A. Dunkels, O. Schmidt, N. Finne, J. Eriksson, F. Österlind, N. Tsiftes, and M. Durvy. The Contiki OS: The Operating System for the Internet of Things. `www.contiki-os.org`, 2011. Accessed: 2016-04-15.

[43] P. Dutta and a. Dunkels. Operating systems and network protocols for wireless sensor networks. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 370(1958):68–84, 2012.

[44] P. Dutta, J. Hui, J. Jeong, S. Kim, C. Sharp, J. Taneja, G. Tolle, K. Whitehouse, and D. Culler. Trio: enabling sustainable and scalable outdoor wireless sensor network deployments. In *2006 5th International Conference on Information Processing in Sensor Networks*, pages 407–415. IEEE, 2006.

[45] V. Dyo, K. Yousef, S. A. Ellwood, D. W. Macdonald, A. Markham, N. Trigoni, R. Wohlers, C. Mascolo, B. Pásztor, and S. Scellato. WILDSENSING. *ACM Transactions on Sensor Networks*, 8(4):1–33, September 2012.

[46] J. Eliasson, P. Lindgren, J. Delsing, S. J. Thompson, and Y. B. Cheng. A power management architecture for sensor nodes. *IEEE Wireless Communications and Networking Conference, WCNC*, (APRIL):3010–3015, 2007.

[47] J. Elson, L. Girod, and D. Estrin. Fine-grained network time synchronization using reference broadcasts. *ACM SIGOPS Operating Systems Review*, 36(SI):147–163, December 2002.

[48] A. Eswaran, A. Rowe, and R. Rajkumar. Nano-RK: An Energy-Aware Resource-Centric RTOS for Sensor Networks. In *26th IEEE International Real-Time Systems Symposium (RTSS'05)*, pages 256–265. IEEE, 2005.

[49] M. O. Farooq and T. Kunz. Operating systems for wireless sensor networks: A survey. *Sensors*, 11(6):5900–5930, 2011.

[50] P. A. Frangoudis, G. C. Polyzos, and V. P. Kemerlis. Wireless community networks: An alternative approach for nomadic broadband network access. *IEEE Communications Magazine*, 49(5):206–213, 2011.

[51] S. C. Gadanho and J. Hallam. Emotion-Triggered Learning in Autonomous Robot Control. *Cybernetics and Systems*, 32(5):531–559, 2001.

[52] S. Ganeriwal, R. Kumar, and M. B. Srivastava. Timing-sync protocol for sensor networks. In *Proceedings of the first international conference on Embedded networked sensor systems - SenSys '03*, page 138, New York, New York, USA, 2003. ACM Press.

[53] L. Gao, M. Zhang, and G. Chen. An Intelligent irrigation system based on wireless sensor network and fuzzy control. *Journal of Networks*, 8(5):1080–1087, 2013.

[54] D. Gay, P. Levis, R. von Behren, E. Welsh, M.and Brewer, and D. Culler. The nesC language. In *Proceedings of the ACM SIGPLAN 2003 conference on Programming language design and implementation - PLDI '03*, page 1, New York, New York, USA, 2003. ACM Press.

[55] A. Giagkos and M. S. Wilson. BeeIP - A Swarm Intelligence based routing for wireless ad hoc networks. *Information Sciences*, 265:23–35, 2014.

[56] C. Gomez and J. Paradells. Wireless Home Automation Networks- A Survey of Architectures and Technologies. *IEEE Communications Magazine*, 48(June):92–101, 2010.

[57] R. Guy, B. Greenstein, J. Hicks, R. Kapur, N. Ramanathan, T. Schoellhammer, T. Stathopoulos, K. Weeks, K. Chang, L. Girod, and D. Estrin. Experiences with the Extensible Sensing System ESS. *Center for Embedded Network Sensing*, page 61, 2006.

[58] H. Hamann, T. Schmickl, and K. Crailsheim. A hormone-based controller for evolutionary multi-modular robotics: From single modules to gait learning. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–8, 2010.

[59] H. Hamann, T. Schmickl, and K. Crailsheim. A hormone-based controller for evaluation-minimal evolution in decentrally controlled systems. *Artificial life*, 18(2):165–98, 2012.

[60] H. Hamann, J. Stradner, T. Schmickl, and K. Crailsheim. Artificial Hormone Reaction Networks: Towards Higher Evolvability in Evolutionary Multi-Modular Robotics. *Artificial Life*, 12:773–780, nov 2010.

[61] M. Handy, M. Haase, and D. Timmermann. Low energy adaptive clustering hierarchy with deterministic cluster-head selection. In *4th International Workshop on Mobile and Wireless Communications Network*, pages 368–372, 2002.

[62] P. Haneveld. Evading murphy: A sensor network deployment in precision agriculture. `http://www.st.ewi.tudelft.nl/~koen/papers/LOFAR-agro-take2.pdf`, 2007. Accessed: 2016-04-15.

[63] D. G. Hardie. *Biochemical Messengers: Hormones, Neurotransmitters an Growth Factors*. Chapman and Hall, 1991.

[64] J. K. Hart and K. Martinez. Environmental Sensor Networks: A revolution in the earth system science? *Earth-Science Reviews*, 78(3-4):177–191, 2006.

[65] A. Hasler, I. Talzi, and J. Beutel. Wireless sensor networks in permafrost research-concept, requirements, implementation and challenges. *Proceedings of the Ninth international conference on permafrost*, pages 669–674, 2008.

[66] T. He, S. Krishnamurthy, L. Luo, T. Yan, L. Gu, R. Stoleru, G. Zhou, Q. Cao, P. Vicaire, J. Stankovic, T. F. Abdelzaher, J. Hui, and B. Krogh. VigilNet: An integrated sensor network system for energy-efficient surveillance. *ACM Trans. Sen. Netw.*, 2(1):1–38, 2006.

[67] W. R. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *Proceedings of the 5th annual*

ACM/IEEE international conference on Mobile computing and networking - MobiCom '99, pages 174–185, New York, New York, USA, 1999. ACM Press.

[68] W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, volume vol.1, page 10. IEEE Comput. Soc, 2000.

[69] S. Henna. RI-MAC: Sender-assisted receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks. *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering*, 81 LNICST:120–135, 2012.

[70] J. Hill, M. Horton, and R. Kling. The Platforms Enabling Wireless Sensor Networks. *Communications of the ACM*, 47(6):41–46, 2004.

[71] J. L. Hill. *System Architecture for Wireless Sensor Networks.* PhD thesis, University of Berkeley, 2003.

[72] J. L. Hill and D. E. Culler. Mica: A wireless platform for deeply embedded networks. *IEEE Micro*, 22(6):12–24, 2002.

[73] J. Hinson, P. Raven, and S. Chew. *The Endocrine System.* Elsevier, 2nd edition, 2007.

[74] J. Hsu, S. Zahedi, A. Kansal, M. Srivastava, and V. Raghunathan. Adaptive Duty Cycling for Energy Harvesting Systems. In *Proceedings of the 2006 International Symposium on Low Power Electronics and Design*, pages 180–185. IEEE, October 2006.

[75] L. Huang, P.and Xiao, S. Soltani, M. W. Mutka, and N. Xi. The Evolution of MAC Protocols in Wireless Sensor Networks: A Survey. *IEEE Communications Surveys & Tutorials*, 15(1):101–120, January 2013.

[76] R. Huang, W. Z. Song, M. Xu, N. Peterson, B. Shirazi, and R. Lahusen. Real-world sensor network for long-term volcano monitoring: Design and findings. *IEEE Transactions on Parallel and Distributed Systems*, 23(2):321–329, 2012.

[77] S. W. Hussain, T. Khan, and S. M. H. Zaidi. Latency and Energy Efficient MAC (LEEMAC) Protocol for Event Critical Applications in WSNs. In *International Symposium on Collaborative Technologies and Systems (CTS'06)*, pages 370–378. IEEE, 2006.

[78] A. Ibrahim Abdu. An Adaptive Energy-Aware Transmission Scheme for Wireless Sensor Networks. *International Journal of Wireless Communications and Mobile Computing*, 1(1):14, 2013.

[79] F. Ingelrest, G. Barrenetxea, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange. SensorScope. *ACM Transactions on Sensor Networks*, 6(2):1–32, 2010.

[80] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed Diffusion : A Scalable and Robust Communication. *Proceedings of the 6th annual international conference on Mobile computing and networking (MobiCom '00)*, pages 56–67, 2000.

[81] Intel. Intel Mote 2 Engineering Platform. `http://wsn.cse.wustl.edu/images/c/cb/Imote2-ds-rev2_2.pdf`. Accessed: 2016-04-15.

[82] P. Jacquet, P. Mühlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized link state routing protocol for ad hoc networks. In *Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International*, pages 62–68. IEEE, 2001.

[83] X. Jiang, J. Polastre, and D. Culler. Perpetual environmentally powered sensor networks. *4th International Symposium on Information Processing in Sensor Networks*, 2005:463–468, 2005.

[84] X. Jiang, J. Taneja, J. Ortiz, A. Tavakoli, P. Dutta, J. Jeong, D. Culler, P. Levis, and S. Shenker. An architecture for energy management in wireless sensor networks. *ACM SIGBED Review*, 4(3):31–36, 2007.

[85] Y. Jin, Y. Ding, and K. Hao. Using an improved hormone secreting mechanism to realize persistent object tracking in WSN. In *Proceedings of the 3rd International Conference on Digital Manufacturing and Automation, ICDMA 2012*, pages 1009–1013, 2012.

[86] Y. Jin, Y. Ding, K. Hao, and Y. Jin. An endocrine-based intelligent distributed cooperative algorithm for target tracking in wireless sensor networks. *Soft Computing*, 19(5):1427–1441, 2015.

[87] D. Johnson and C. Aichele. A simple pragmatic approach to mesh routing using. In *2nd IFIP International Symposium on Wireless Communications and Information Technology in Developing Countries*, page 12, 2008.

[88] D. B. Johnson. Routing in Ad Hoc Networks of Mobile Hosts. *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications*, 1(December):158–163, 1994.

[89] 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. *Proceedings of IEEE Sensors*, (NOVEMBER):1439–1442, 2009.

[90] D. H. Jones and G. H. Gudmundsson. Instruments and methods: Aircraft-deployable ice observation system (ADIOS) for instrumenting inaccessible glaciers. *Journal of Glaciology*, 59(218):1129–1134, 2013.

[91] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking. *ACM SIGOPS Operating Systems Review*, 36(5):96–107, December 2002.

[92] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Next century challenges: mobile networking for Smart Dust. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking - MobiCom '99*, pages 271–278, New York, New York, USA, 1999. ACM Press.

[93] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Emerging Challenges: Mobile Networking for Smart Dust. *Journal Of Communications And Networks*, 2(3):188–196, 2000.

[94] A. Kansal, J. Hsu, M. Srivastava, and V. Raqhunathan. Harvesting aware power management for sensor networks. In *Proceedings of the 43rd ACM/IEEE Design Automation Conference*, pages 651–656. IEEE, 2006.

[95] K. K. Khedo, M. R. Hosseny, and M. Z. Toonah. PotatoSense: A wireless sensor network system for precision agriculture. In *Proceedings of the 2014 IST-Africa Conference*, pages 1–11. IEEE, May 2014.

[96] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon. Health Monitoring of Civil Infrastructures Using Wireless Sensor Networks. *2007 6th International Symposium on Information Processing in Sensor Networks*, pages 254–263, 2007.

[97] K. Klues, G. Hackmann, O. Chipara, and C. Lu. A component-based architecture for power-efficient media access control in wireless sensor networks. In *Proceedings of the 5th international conference on Embedded networked sensor systems - SenSys '07*, page 59, New York, New York, USA, 2007. ACM Press.

[98] J. Ko, Q. Wang, T. Schmid, W. Hofer, P. Dutta, and A. Terzis. Egs: A Cortex M3-based mote platform. *Proceedings of the 7th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pages 5–7, 2010.

[99] A. N. Kolmogorov. On tables of random numbers. *Theoretical Computer Science*, 207(2):387–395, 1998.

[100] E. A. Kravitz. Hormonal control of behavior: amines and the biasing of behavioral output in lobsters. *Science (New York, N.Y.)*, 241(4874):1775–1781, 1988.

[101] D. Krüger, D. Pfisterer, and S. Fischer. CUPID - Communication pattern informed duty cycling in sensor networks. *Proceedings of the 5th International Conference on Systems and Networks Communications, ICSNC 2010*, pages 70–75, 2010.

[102] R. V. Kulkarni, S. Member, A. Förster, and G. K. Venayagamoorthy. Computational Intelligence in Wireless Sensor Networks : A Survey. *Communications Surveys & Tutorials, IEEE*, 13(1):68–96, 2011.

[103] R. V. Kulkarni and G. K. Venayagamoorthy. Particle swarm optimization in wireless-sensor networks: A brief survey. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 41(2):262–267, 2011.

[104] K. Langendoen, A. Baggio, and O. Visser. Murphy loves potatoes: experiences from a pilot sensor network deployment in precision agriculture. In *Proceedings of the 20th IEEE International Parallel & Distributed Processing Symposium*, pages 174–182. IEEE, 2006.

[105] L. Li and J.Y. Halpern. Minimum-energy mobile wireless networks revisited. In *Proceedings of the IEEE International Conference on Communications*, volume 1, pages 278–283. IEEE, 2002.

[106] Libelium. Waspmote Datasheet. `http://www.libelium.com/downloads/documentation/waspmote_datasheet.pdf`. Accessed: 2016-04-15.

[107] T. H. Lim, I. Bate, and J. Timmis. Validation of performance data using experimental verification process in wireless sensor network. In *Proceedings of the 2012 IEEE 17th International Conference on Emerging Technologies & Factory Automation (ETFA 2012)*, pages 1–8. IEEE, September 2012.

[108] T. H. Lim, I. Bate, and J. Timmis. A self-adaptive fault-tolerant systems for a dependable Wireless Sensor Networks. *Design Automation for Embedded Systems*, pages 1–28, 2014.

[109] S. Lindsey and C.S. Raghavendra. PEGASIS: Power-efficient gathering in sensor information systems. In *Proceedings of the IEEE Aerospace Conference*, volume 3, pages 3–1125–3–1130. IEEE, 2002.

[110] T. Liu, C. M. Sadler, P. Zhang, and M. Martonosi. Implementing Software on Resource-Constrained Mobile Sensors : Experiences with Impala and ZebraNet. *System*, pages 256–269, 2004.

[111] Y. Liu, Y. He, M. Li, J. Wang, K. Liu, and X. Li. Does Wireless Sensor Network Scale? A Measurement Study on GreenOrbs. *IEEE Transactions on Parallel and Distributed Systems*, 24(10):1983–1993, October 2013.

[112] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications - WSNA '02*, page 88, New York, New York, USA, 2002. ACM Press.

[113] A. Manjeshwar and D. P. Agrawal. TEEN: a routing protocol for enhanced efficiency in wireless sensor networks. In *Proceedings of the 15th International Parallel and Distributed Processing Symposium. IPDPS 2001*, number JANUARY 2014, pages 2009–2015. IEEE Comput. Soc, 2001.

[114] a. Manjeshwar and D. P. Agrawal. APTEEN: a hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks. *Proceedings 16th International Parallel and Distributed Processing Symposium*, 2002.

[115] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi. The Flooding Time Synchronization Protocol. *Proceedings of the ACM International Conference on Embedded Networked Sensor Systems*, pages 39–49, 2004.

[116] D. Marsh, R. Tynan, D. O'Kane, and G. M. P. O'Hare. Autonomic wireless sensor networks. *Engineering Applications of Artificial Intelligence*, 17(7):741–748, 2004.

[117] I. Martin, T. O Farrell, R. Aspey, S. Edwards, T. James, P. Loskot, T. Murray, I. Rutt, N. Selmes, and T. Baugé. A High-Resolution Sensor Network for Monitoring Glacier Dynamics. 14(11):3926–3931, 2014.

[118] K. Martinez and P. Basford. Robust wireless sensor network performance analysis. *Proceedings of the IEEE SENSORS Conference*, (3):203–206, 2011.

[119] K. Martinez, P. Basford, J. Ellul, and R. Spanton. Gumsense-a high power low power sensor node. In *6th European Conference on Wireless Sensor Networks.*, pages 4–5, 2009.

[120] K. Martinez, P. J. Basford, D. D. Jager, and J. K. Hart. A wireless sensor network system deployment for detecting stick slip motion in glaciers. In *Proceedings of the IET Conference on Wireless Sensor Systems (WSS 2012)*, pages 1A2–1A2. IET, 2012.

[121] K. Martinez, J. Hart, and R. Ong. Environmental sensor networks. *IEEE Computer*, 37(8):50–56, 2004.

[122] K. Martinez, J. K. Hart, and R. Ong. Deploying a wireless sensor network in iceland. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5659 LNCS:131–137, 2009.

[123] K. Martinez, P. Padhy, A. Riddoch, R. Ong, and J. Hart. Glacial environment monitoring using sensor networks. In *Proceedings of the 1st Workshop on Real-World Wireless Sensor Networks.*, 2005.

[124] MEMSIC. LOTUS Datasheet. `http://www.memsic.com/userfiles/files/Datasheets/WSN/6020-0705-01_A_LOTUS.pdf`. Accessed: 2016-04-15.

[125] MEMSIC. MicaZ Datasheet. `www.openautomation.net/uploadsproductos/micaz_datasheet.pdf`. Accessed: 2016-04-15.

[126] MEMSIC. IRIS Sensor Node Module. `http://www.memsic.com/userfiles/files/Datasheets/WSN/6020-0124-01_B_IRIS.pdf`, 2011. Accessed: 2016-04-15.

[127] M. Mendao. Hormonally moderated neural control. In *AAAI Spring Symposium.* AAAI Press, Menlo Park, CA. 2004, 2004.

[128] G. V. Merrett, N. R. Harris, B. M. Al-Hashimi, and N. M. White. Energy managed reporting for wireless sensor networks. *Sensors and Actuators, A: Physical*, 142(1):379–389, 2008.

[129] R. Mugwanya and G. Marsden. E-Infrastructures and E-Services on Developing Countries. *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering*, 38(April):20–31, 2010.

[130] T. Murray, N. Selmes, T. D. James, S. Edwards, I. Martin, T. O'Farrell, R. Aspey, I. Rutt, M. Nettles, and T. Baugé. Dynamics of glacier calving at the ungrounded margin of Helheim Glacier, southeast Greenland. *Journal of Geophysical Research: Earth Surface*, 120(6):964–982, June 2015.

[131] S. Murthy and J. Garcia-Luna-Aceves. An efficient routing protocol for wireless networks. *Mobile Networks and Applications*, pages 183–197, 1996.

[132] L. Nachman, J. Huang, J. Shahabdeen, R. Adler, and R. Kling. IMOTE2: Serious computation at the edge. In *Proceedings of the International Wireless Communications and Mobile Computing Conference*, pages 1118–1123, 2008.

[133] T. Naumowicz, R. Freeman, A. Heil, M. Calsyn, E. Hellmich, A. Brändle, T. Guilford, and J. Schiller. Autonomous monitoring of vulnerable habitats using a wireless sensor

network. In *Proceedings of the workshop on Real-world wireless sensor networks - REALWSN '08*, page 51, New York, New York, USA, 2008. ACM Press.

[134] M. Neal and J. Timmis. Timidity: A useful emotional mechanism for robot control? *Informatica*, 4(27):197–204, 2003.

[135] M. Neal and J. Timmis. Once More Unto the Breach: Towards Artificial Homeostasis. In *Recent Developments in Biologically Inspired Computing*, pages 340–365. Igi Global, 2005.

[136] G. Oliver and E. A. Schäfer. The physiological effects of extracts of the suprarenal capsules. *The Journal of physiology*, 18(3):230–276, 1895.

[137] OLPC Organization. OLPC. `http://one.laptop.org`. Accessed: 2016-04-15.

[138] P. Padhy, R. K. Dash, K. Martinez, and N. R. Jennings. A utility-based sensing and communication model for a glacial sensor network. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 1353–1360. ACM, 2006.

[139] N. a Pantazis, S. Nikolidakis, and D. D. Vergados. Energy-Efficient Routing Protocols in Wireless Sensor Networks : A Survey. *IEEE Communications Surveys & Tutorials*, 15(2):551–591, 2013.

[140] C. Park and P. H. Chou. AmbiMax: Autonomous energy harvesting platform for multi-supply wireless sensor nodes. *2006 3rd Annual IEEE Communications Society on Sensor and Adhoc Communications and Networks, Secon 2006*, 1(c):168–177, 2007.

[141] V. D. Park and M. S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. *Infocom*, pages 1405–1413, 1997.

[142] S. N. Parmar, S. Nandi, and A. R. Chowdhury. Power Efficient and Low Latency MAC for Wireless Sensor Networks. *2006 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks*, 3(C):940–944, 2006.

[143] V. Paruchuri, S. Basavaraju, A. Durresi, R. Kannan, and S. S. Iyengar. Random asynchronous wakeup protocol for sensor networks. *Proceedings - First International Conference on Broadband Networks, BroadNets 2004*, pages 710–717, 2004.

[144] C. E. Perkins, M. Park, and E. M. Royer. Ad-hoc On-Demand Distance Vector Routing. *In Proceedings of Second IEEE Workshop on Mobile Computing Systems and Applications (WMCSA), 1999*, pages 90–100, 1999.

[145] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems - SenSys '04*, page 95, New York, New York, USA, 2004. ACM Press.

[146] J. Polastre, R. Szewczyk, A. Mainwaring, D. Culler, and J. Anderson. Analysis of Wireless Sensor Networks for Habitat Monitoring. In *Wireless sensor networks*, chapter 18, pages 399–423. 2004.

[147] J. M. T. Portocarrero, F. C. Delicato, P. F. Pires, N. Gámez, L. Fuentes, D. Ludovino, and P. Ferreira. Autonomic Wireless Sensor Networks: A Systematic Literature Review. *Journal of Sensors*, 2014:1–13, 2014.

[148] S. Rajasegarar, C. Leckie, and M. Palaniswami. Anomaly detection in wireless sensor networks. *IEEE Wireless Communications*, 15(4):34–40, August 2008.

[149] V. Rajendran, K. Obraczka, and J.J. Garcia-Luna-Aceves. Energy-Efficient , Collision-Free Medium Access Control for Wireless Sensor Networks. *Proceedings of International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 1–34, 2003.

[150] V. Rodoplu and T. H. Meng. Minimum energy mobile wireless networks. *IEEE Journal on Selected Areas in Communications*, 3(8):1333–1344, 1999.

[151] D. Roemmich, G. Johnson, S. Riser, R. Davis, J. Gilson, W. B. Owens, S. Garzoli, C. Schmid, and M. Ignaszewski. The Argo Program: Observing the Global Oceans with Profiling Floats. *Oceanography*, 22(2):34–43, 2009.

[152] G. E. Rolader, J. Rogers, and J. Batteh. Self-healing minefield. In Raja Suresh, editor, *Battlespace Digitization and Network-Centric Systems IV*, volume 5441 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 13–24, July 2004.

[153] A. Rowe, R. Mangharam, and R. Rajkumar. RT-Link: A Time-Synchronized Link Protocol for Energy- Constrained Multi-hop Wireless Networks. In *2006 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks*, volume 2, pages 402–411. IEEE, 2006.

[154] M. Saleem, G. a. Di Caro, and M. Farooq. Swarm intelligence based routing protocol for wireless sensor networks: Survey and future directions. *Information Sciences*, 181(20):4597–4624, 2011.

[155] M. Saleem, I. Ullah, and M. Farooq. BeeSensor: An energy-efficient and scalable routing protocol for wireless sensor networks. *Information Sciences*, 200:38–56, 2012.

[156] C. Sauze. *A Neuro-Endocrine Inspired Approach to Power Management in Sailing Robots*. PhD thesis, Department of Computer Science, Aberystwyth University, 2011.

[157] E. A. Schafer. On Internal Secretions. *Address in Physiology*, 1895.

[158] T. Schmid, H. Dubois-Ferrière, and M. Vetterli. SensorScope: Experiences with a Wireless Building Monitoring Sensor Network. *Proc. Workshop on Real-World Wireless Sensor Networks (REALWSN '05)*, (5005), 2005.

[159] T. Schmid, P. Dutta, and M. B. Srivastava. High-resolution, low-power time synchronization an oxymoron no more. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks - IPSN '10*, page 151, New York, New York, USA, 2010. ACM Press.

[160] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava. Optimizing sensor networks in the energy-latency-density design space. *IEEE Transactions on Mobile Computing*, 1(1):70–80, 2002.

[161] S. Selvakennedy, S. Sinnappan, and Y. Shang. A biologically-inspired clustering protocol for wireless sensor networks. *Computer Communications*, 30(14-15):2786–2801, 2007.

[162] C. E. Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3, January 2001.

[163] W. Shen, Y. Lu, and P. Will. Hormone-based control for self-reconfigurable robots. In *Proceedings of the fourth international conference on Autonomous agents - AGENTS '00*, pages 1–8, New York, New York, USA, 2000. ACM Press.

[164] W. Shen, B. Salemi, and P. Will. Hormone-inspired adaptive communication and distributed control for CONRO self-reconfigurable robots. *IEEE Transactions on Robotics and Automation*, 18(5):700–712, 2002.

[165] W. Shen, P. Will, A. Galstyan, and C. Chuong. Hormone-Inspired Self-Organization and Distributed Control of Robotic Swarms. *Autonomous Robots*, 17(1):93–105, July 2004.

[166] Shimmer. Shimmer Mote. `http://www.shimmersensing.com`, 2015. Accessed: 2016-04-15.

[167] R. B. Smith. SPOTWorld and the Sun SPOT. *2007 6th International Symposium on Information Processing in Sensor Networks*, pages 565–566, 2007.

[168] P. Sommer and R. Wattenhofer. Gradient Clock Synchronization in Wireless Sensor Networks. *Proceedings of the International Conference on Information Processing in Sensor Networks, 2009*, pages 37 – 48, 2009.

[169] E. Starling. The Croonian Lectures on the Chemical Correlation of the Functions of the Body. *The Lancet*, 166(4277):501–503, August 1905.

[170] J. Stradner, H. Hamann, T. Schmickl, and K. Crailsheim. Analysis and implementation of an artificial homeostatic hormone system: A first case study in robotic hardware. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*, pages 595–600, 2009.

[171] S. Sudevakatam and P. Kulkarni. Energy Harvesting Sensor Nodes: Survez and Implications. *IEEE Communications Surveys and Tutorials*, 13(3):443–461, 2011.

[172] E. W. Sutherland. Studies on the mechanism of hormone action. *Science (New York, N.Y.)*, 177(47):401–408, 1972.

[173] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler. An analysis of a large scale habitat monitoring application. *Proceedings of the 2nd international conference on Embedded networked sensor systems SenSys 04*, 2:214–226, 2004.

[174] R. Szewczyk and J. Polastre. Telos: Enabling ultra-low power wireless research. *IPSN/SPOTS*, 00(C), 2005.

[175] R. Szewczyk, J. Polastre, A. Mainwaring, and D. Culler. Lessons from a Sensor Network Expedition. *Wireless Sensor Networks*, pages 307–322, 2004.

[176] I. Talzi, A. Hasler, S. Gruber, and C. Tschudin. PermaSense: Investigating permafrost with a WSN in the Swiss Alps. *Proceedings of the Fourth Workshop on Embedded Networked Sensors, 25-26 June 2007*, pages 8–12, 2007.

[177] L. Tang, Y. Sun, O. Gurewitz, and D. B. Johnson. PW-MAC: An energy-efficient predictive-wakeup MAC protocol for wireless sensor networks. In *Proceedings of the 2011 IEEE INFOCOM Conference*, pages 1305–1313. IEEE, April 2011.

[178] J. Tateson, C. Roadknight, A. Gonzalez, T. Khan, S. Fitz, I. Henning, N. Boyd, C. Vincent, and I. Marshall. Real World Issues in Deploying a Wireless Sensor Network for Oceanography. In *Workshop on Real-World Wireless Sensor Networks*, 2005.

[179] G. Tolle, D. Gay, W. Hong, J. Polastre, R. Szewczyk, D. Culler, N. Turner, K. Tu, S. Burgess, T. Dawson, and P. Buonadonna. A macroscope in the redwoods. In *Proceedings of the 3rd international conference on Embedded networked sensor systems - SenSys '05*, page 51, New York, New York, USA, 2005. ACM Press.

[180] W. Trumler, A. Pietzowski, B. Satzger, and T. Ungerer. Adaptive self-optimization in distributed dynamic environments. In *Proceedings of the First International Conference on Self-Adaptive and Self-Organizing Systems, SASO 2007*, pages 320–323, 2007.

[181] W. Trumler, T. Thiemann, and T. Ungerer. An Artificial Hormone System for Self-organization of Networked Nodes. In *Biologically Inspired Cooperative Computing*, volume 216, pages 85–94. Springer US, 2006.

[182] T. van Dam and K. Langendoen. An adaptive energy-efficient MAC protocol for wireless sensor networks. *Proc. 1st international conference on Embedded networked sensor systems (SenSys '03)*, pages 171–180, 2003.

[183] P. Vargas, R. Moioli, L. N. de Castro, J. Timmis, M. Neal, and F. J. Von Zuben. Artificial Homeostatic System: A Novel Approach. In *Proceedings of the 8th European conference on Advances in Artificial Life*, pages 754–764, 2005.

[184] P. Vicaire, T. F. Abdelzaher, T. He, Q. Cao, T. Yan, G. Zhou, L. Gu, L. Luo, R. Stoleru, and J. a Stankovic. Achieving long-term surveillance in VigilNet. *ACM Transactions on Sensor Networks*, 5(1):1–39, February 2009.

[185] L. J. G. Villalba, A. L. S. Orozco, A. T. Cabrera, and C. J. B. Abbas. Routing protocols in wireless sensor networks. *Sensors (Basel, Switzerland)*, 9(11):8399–421, 2009.

[186] J. H. Walker and M. S. Wilson. Task allocation for robots using inspiration from hormones. *Adaptive Behavior*, 19(3):208–224, 2011.

[187] C. Wallenta, J. Kim, P. J. Bentley, and S. Hailes. Detecting interest cache poisoning in sensor networks using an artificial immune algorithm. *Applied Intelligence*, 32(1):1–26, 2010.

[188] J. Wang, Z. Zhang, F. Xia, W. Yuan, and S. Lee. An energy efficient stable election-based routing algorithm for wireless sensor networks. *Sensors (Basel, Switzerland)*, 13(11):14301–20, 2013.

[189] B. Warneke, M. Last, B. Liebowitz, and K. S. J. Pister. Smart dust: communicating with a cubic-millimeter computer. *Computer*, 34(1):44–51, 2001.

[190] A. S. Weddell, M. Magno, G. V. Merrett, D. Brunelli, B. M. Al-Hashimi, and L. Benini. A Survy of Multi-Source Energy Harvesting Systems. In *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE), 2013*, pages 905–908, New Jersey, 2013. IEEE Conference Publications.

[191] G. Werner-Allen, J. Johnson, M. Ruiz, J. Lees, and M. Welsh. Monitoring volcanic eruptions with a wireless sensor network. In *Proceeedings of the Second European Workshop on Wireless Sensor Networks, 2005.*, pages 108–120. IEEE, 2005.

[192] G. Werner-allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh. Fidelity and Yield in a Volcano Monitoring Sensor Network. In *OSDI . USENIX, 2006.*, 2006.

[193] G. Werner-Allen, K. Lorincz, M. Welsh, O. Marcillo, J. Johnson, M. Ruiz, and J. Lees. Deploying a wireless sensor network on an active volcano. *IEEE Internet Computing*, 1:18–24, 2006.

[194] L. Xia, W Xufa, L. Yang, and Y. Haifeng. A self-organized algorithm based on digital hormone. *Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering*, 2:398–402, 2010.

[195] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin. A wireless sensor network For structural monitoring. In *Proceedings of the 2nd international conference on Embedded networked sensor systems - SenSys '04*, page 13, New York, New York, USA, 2004. ACM Press.

[196] Q. Xu and L. Wang. Recent advances in the artificial endocrine system. *Journal of Zhejiang University SCIENCE C*, 12(3):171–183, 2011.

[197] Y. Xu, J. Heidemann, and D. Estrin. Geography-informed energy conservation for Ad Hoc routing. In *Proceedings of the 7th annual international conference on Mobile computing and networking - MobiCom '01*, pages 70–84, New York, New York, USA, 2001. ACM Press.

[198] M. Yamamoto. Sozzy: A hormone-driven autonomous vacuum cleaner. *Optical Tools for Manufacturing and Advanced Automation*, pages 211–223, 1994.

[199] Y. Yao and J. Gehrke. The cougar approach to in-network query processing in sensor networks. *ACM SIGMOD Record*, 31(3):9, 2002.

[200] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 3, pages 1567–1576. IEEE, 2002.

[201] D. Young, L. Chapman, C. Muller, X. Cai, and C. S. B. Grimmond. A Low-Cost Wireless Temperature Sensor: Evaluation for Use in Environmental Monitoring Applications. *Journal of Atmospheric and Oceanic Technology*, 31:938–944, April 2014.

[202] O. Younis and S. Fahmy. Distributed Clustering in Ad Hoc Sensor Networks: A Hybrid, Energy-Efficient Approach. *Proceedings of the 23rd Annual Joint Conference*

of the IEEE Computer and Communications Societies, INFOCOM 2004, Hong Kong, 3(4):629–640, 2004.

[203] Y. Yu, R. Govindan, and D. Estrin. Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks. Technical report, UCLA Computer Science Department, 2001.

[204] M. Zamani, M. Movahedi, M. Ebadzadeh, and H. Pedram. A DDoS-aware IDS model based on danger theory and mobile agents. *Proceedings of the 2009 International Conference on Computational Intelligence and Security*, 1:516–520, 2009.

[205] P. Zhang, C. M. Sadler, S. a. Lyon, and M. Martonosi. Hardware design experiences in ZebraNet. In *Proceedings of the 2nd international conference on Embedded networked sensor systems - SenSys '04*, page 227, New York, New York, USA, 2004. ACM Press.

[206] Y. Zhang, L. D. Kuhn, and M. P. J. Fromherz. Improvements on Ant Routing for Sensor Networks. *Networks*, pages 154–165, 2004.

[207] Y. Zhang, H. You, and X. Wang. A hormone based tracking strategy for wireless sensor network. In *Proceedings of the 2009 IEEE International Conference on Intelligent Computing and Intelligent Systems*, pages 104–108. IEEE, November 2009.

[208] J. Zhou and D. De Roure. FloodNet: Coupling adaptive sampling with energy aware routing in a flood warning system. *Journal of Computer Science and Technology*, 22(1):121–130, 2007.

[209] Y. Zhu, J. Song, and F. Dong. Applications of Wireless Sensor Network in the agriculture environment monitoring. *Procedia Engineering*, 16:608–614, 2011.

[210] Zigbee Alliance. Zigbee Specification. `http://www.zigbee.org/zigbee-for-developers/network-specifications/`, 2006. Accessed: 2016-04-15.

[211] A. M. Zungeru, L. Ang, and K. Seng. Classical and swarm intelligence based routing protocols for wireless sensor networks: A survey and comparison. *Journal of Network and Computer Applications*, 35(5):1508–1536, 2012.

Appendix A

Integrated Light vs Voltage Change Plots

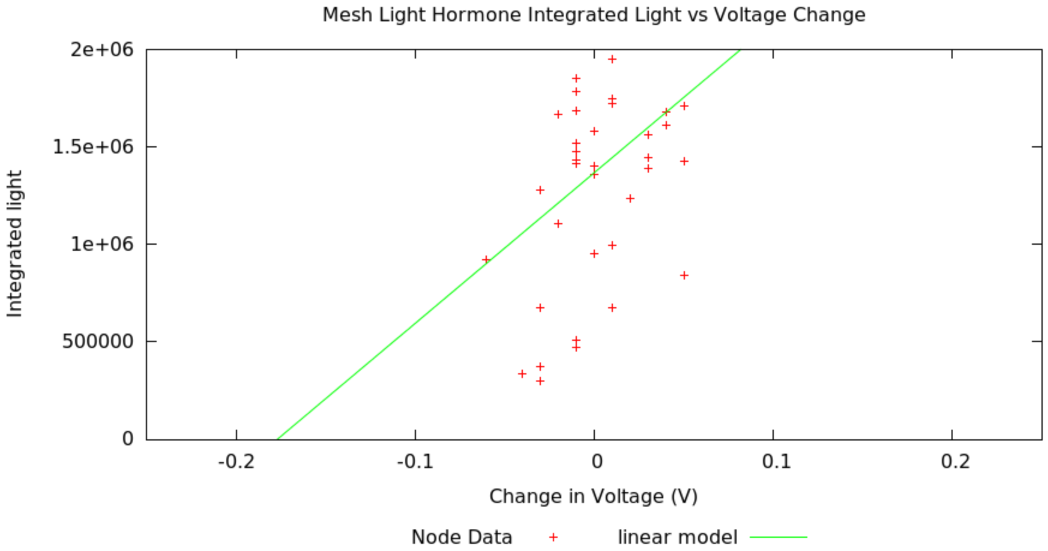

Figure A.1: Integrated Light over Voltage Change for the Mesh Light Hormone Experiment.

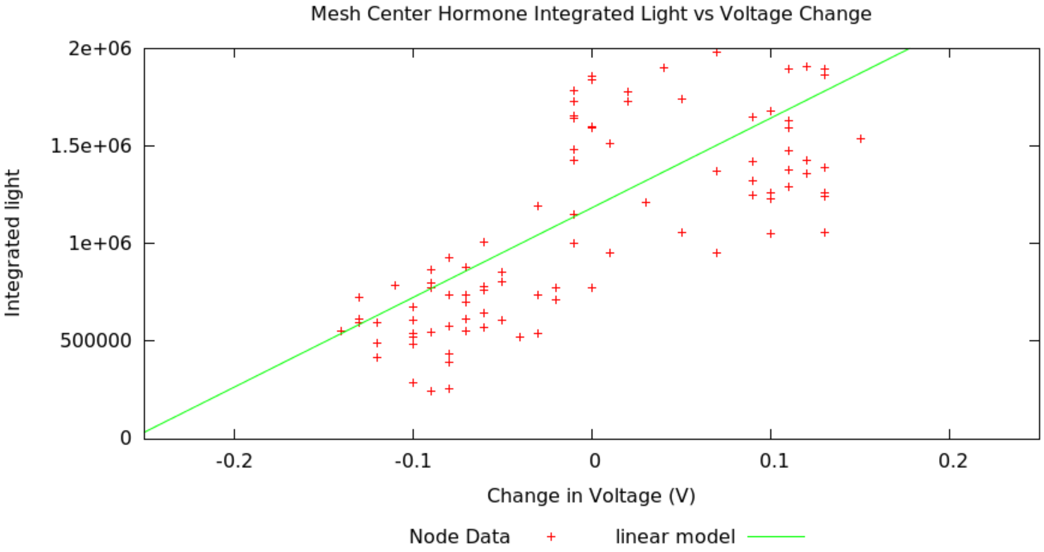

Figure A.2: Integrated Light over Voltage Change for the Mesh Center Hormone Experiment.

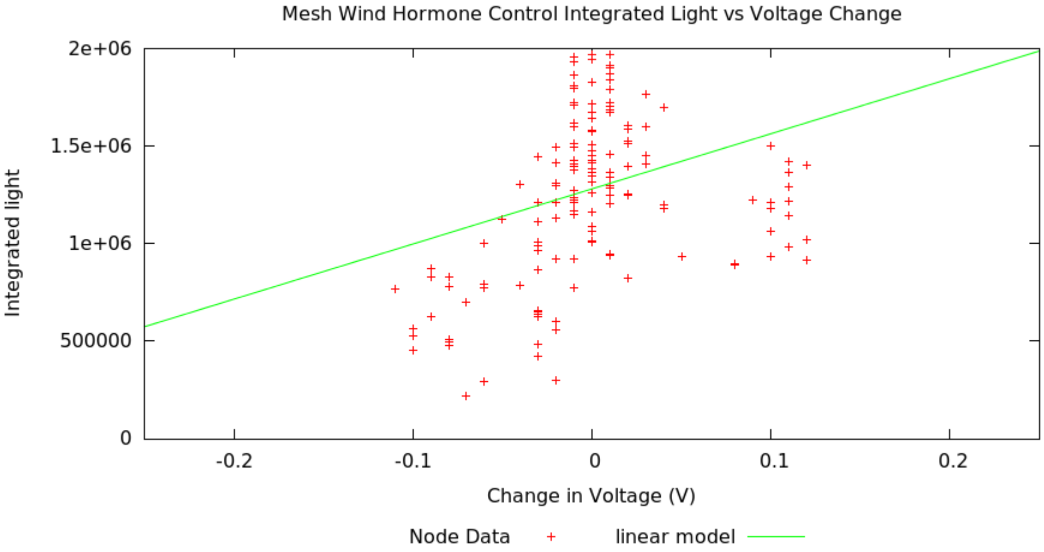

Figure A.3: Integrated Light over Voltage Change for the Mesh Light and Wind Experiment.

Appendix B

Light Detector Event Plots

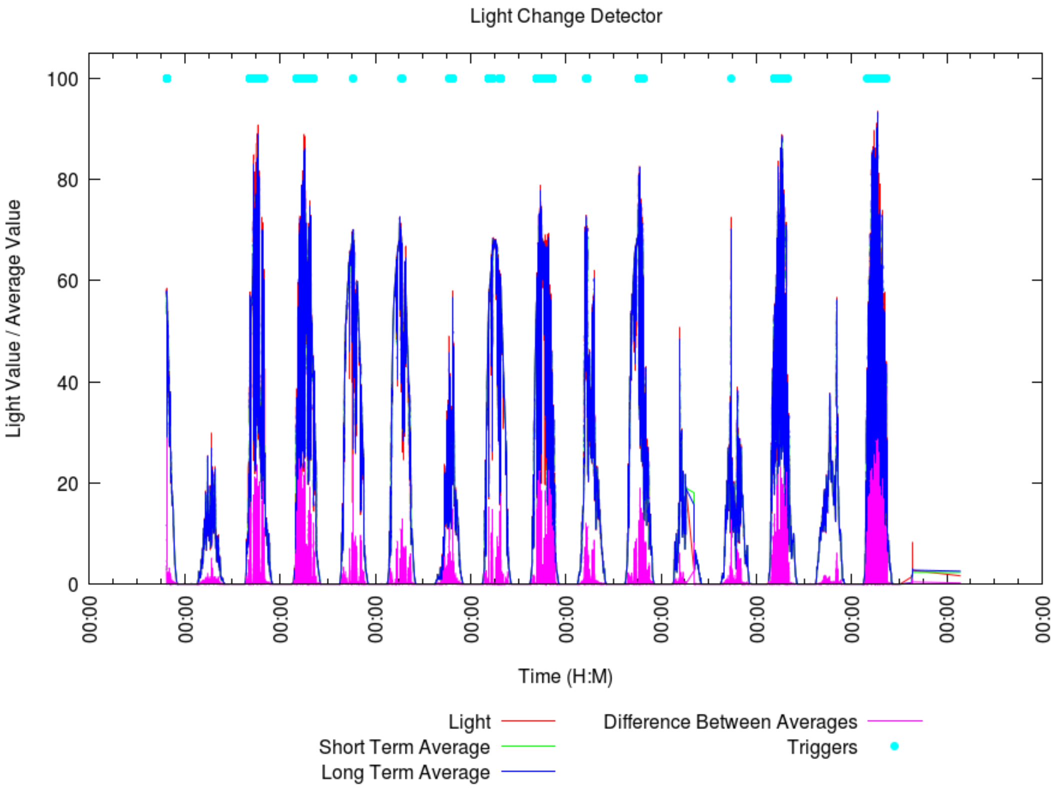

Figure B.1: Light detection events over a two week period on a single node. 1890 events were detected over this two week period.

Appendix C

Periodicity Correlations

Table C.1: Correlation between different environmental factors and packet count. Each variable was averaged on an hour by hour basis to match the packet count. Values have been rounded to 3 decimal places. N is 370 for all correlations and all correlations can be accepted as significant.

Node	Temp	p value	Pressure	p value	Humid	p value	Light	p value	Battery	p value
G	-0.66	1.482×10^{-47}	-0.129	0.01269	0.466	2.501×10^{-21}	-0.703	1.731×10^{-56}	-0.569	3.788×10^{-33}
O	-0.585	2.485×10^{-35}	-0.08	0.1268	0.384	1.96×10^{-14}	-0.563	2.352×10^{-32}	-0.458	1.255×10^{-20}
S	-0.595	9.115×10^{-37}	-0.152	0.003395	0.374	1.067×10^{-13}	-0.69	1.134×10^{-53}	-0.516	1.578×10^{-26}
F	-0.589	6.306×10^{-36}	-0.087	0.09391	0.395	3.078×10^{-15}	-0.764	6.186×10^{-72}	-0.624	2.303×10^{-41}
P	-0.756	1.128×10^{-69}	-0.127	0.01425	0.571	1.958×10^{-33}	-0.728	2.089×10^{-62}	-0.716	2.27×10^{-59}
H	-0.52	5.111×10^{-27}	-0.108	0.03804	0.319	3.536×10^{-10}	-0.651	6.685×10^{-46}	-0.472	5.914×10^{-22}
B	-0.485	3.395×10^{-23}	-0.022	0.6706	0.304	2.273×10^{-09}	-0.608	8.007×10^{-39}	-0.53	3.306×10^{-28}
M	-0.67	1.766×10^{-49}	-0.138	0.007895	0.464	4.169×10^{-21}	-0.754	3.139×10^{-69}	-0.589	6.227×10^{-36}
K	-0.741	1.487×10^{-65}	-0.102	0.05101	0.599	2.203×10^{-37}	-0.711	4.004×10^{-58}	-0.821	1.824×10^{-91}
Average	-0.622		-0.105		0.4317		-0.686		-0.588	

Appendix D

Duplicate Data Histograms

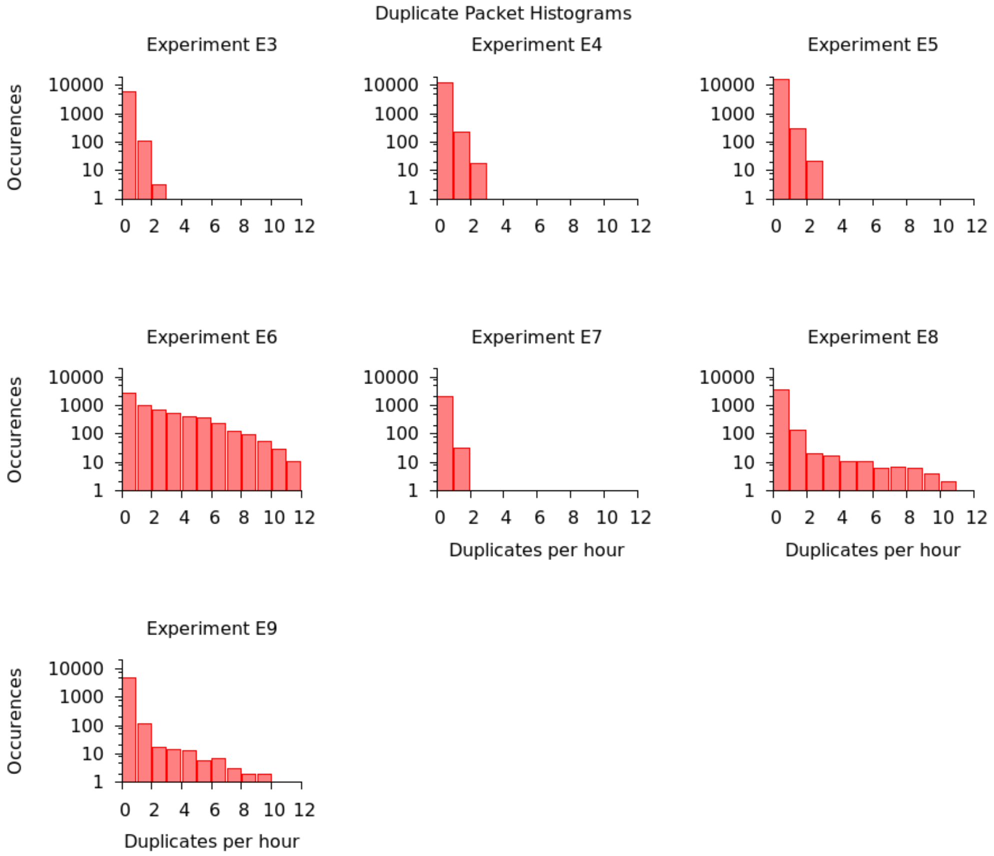

Figure D.1: Histograms showing the frequency of duplicate data per hour for all experiments. The most frequent number of duplicates is 0 in every experiment.

Appendix E

Experiment Weather Conditions

Experiment	Clear	PC	SC	MC	OC	Rain	Fog
E1	0	5	1	5	2	6	0
E2	0	36	0	2	0	6	1
E3	2	8	2	2	1	3	1
E4	0	18	0	3	2	6	0
E5	0	21	1	2	0	9	0
E6	1	15	1	0	0	1	0
E7	0	6	0	0	0	0	0
E8	0	13	0	0	0	0	0
E9	0	19	2	0	0	3	0

Table E.1: The weather experienced by each of the experiment deployments. Each value is the number of days of the experiment that fit into a certain weather category; clear, partially cloudy (PC), scattered clouds (SC), mostly cloudy (MC), overcast (OC), rain and fog.

Appendix F

Node Address Mapping

Node Address	Letter	Node Address	Letter
409a818b	A	409a9301	B
40a1f762	C	40a1f839	D
40aac452	E	409a92ed	F
409a9307	G	40a1f78f	H
40a1f84c	I	40aac497	J
409a92f0	K	409aa1d2	L
40a1f790	M	40a1f856	N
40aac4de	O	409a92fb	P
409aa1d7	Q	40a1f802	R
40aac42a	S	40aac4e1	T

Table F.1: Table showing the mapping of node addresses to letters. The shorter letter names are used in results Chapters 6 and 5 to make discussions clearer.