

Report

NPL Report TQE 3

Investigations Using the Wireless Sensor Network Demonstrator Testbed

**M A Collett, D B Adamson and
T J Esward**

NOT RESTRICTED

August 2008

Investigations Using the Wireless Sensor Network Demonstrator Testbed

M A Collett, D B Adamson
Industry and Innovation Division
T J Esward
Quality of Life Division

August 2008

© Crown copyright 2008
Reproduced with the permission of the Controller of HMSO
and Queen's Printer for Scotland

ISSN 1754-2995

National Physical Laboratory,
Hampton Road, Teddington, Middlesex, United Kingdom TW11 0LW

NPL is not responsible for any consequences arising from the advice contained in this
report.

We gratefully acknowledge the financial support of the UK Department for Innovation,
Universities and Skills (National Measurement System Directorate)

Approved on behalf of the Managing Director, NPL
by J T Janssen, Knowledge Leader for the Time, Quantum and Electromagnetics Team

Executive Summary

Wireless Sensor Networks (WSNs) are set to become a significant enabling technology in many areas of measurement. An increasing range of applications including health monitoring, environmental monitoring and military surveillance are beginning to require WSN based solutions. Traditional metrology concentrates on the calibration of single instruments. WSNs represent a new type of measurement system that employs distributed sensors with both local and centralised processing. Calibration of such distributed measurement systems and the validation of their outputs present significant new metrology challenges.

In this report we summarise initial work carried out at NPL to develop a practical understanding and capability in WSN-based measurements and to demonstrate this expertise internally and externally to establish a basis for future work in this field.

The first stages of this exploratory project involved researching the existing work and industrial needs in the area of WSNs. Through a combination of a literature review, seminar attendance and visits to key collaborators a number of important measurement issues were identified. These included radio link quality, networking, sensors, onboard processing and high-level processing. It was also found that there were two main barriers to adoption of these technologies. Firstly, the need for expertise in software and electrical engineering to set up any network has proved to be prohibitively expensive and time-consuming for many potential users, this demonstrates that the technology is, in many ways still in its infancy. Secondly, the outputs of such systems are generally not fully understood. The accuracy and reliability of WSN measurements needs to be more fully characterised for cost-benefit or risk analysis to be carried out.

In order to investigate some of these measurement issues, as well as provide a demonstration platform, a testbed of MICAz nodes was set-up. These operate under the Zigbee protocol, and were applied to some measurement tasks.

Of the key measurement issues, radio link quality was chosen as one of the areas on which to concentrate. As this is important to all WSN applications, any experimental findings would have a broad impact. The importance of this issue is increased by the likelihood that WSNs will be deployed in harsh environments and be subjected to a range of interferences. Severe degradation of the wireless links could make any system completely inoperative. Using a metric of packet loss-rate the effects of a number of environments and interference types were investigated. It was found that highly reflective environments could cause the radio links to fail entirely, and the level of signal quality was highly position-dependent. When looking at the effects of external interference it was found that relatively low levels of white noise could jam the system and prevent the nodes from transmitting. When a simulated Bluetooth signal was used, it was found that this jamming didn't occur, but the receiver was eventually blinded. The fact that the types and magnitudes of interference effects observed was dependent on the type of signal (rather than just the power) means that new methods of measuring and predicting compatibility will be required.

Another important finding of the review of work in the area of WSNs was that a large proportion of the research carried out elsewhere concentrated on individual aspects of the systems, such as novel sensors or new networking protocols. WSNs are highly integrated systems. The components of sensors, onboard processing, power supply and radio communications are all closely linked. In order to observe all the possible effects, a 'system-based'

approach was required. For this reason it was decided to look at the performance of the testbed when performing the resource-hungry application of acoustic ranging. By measuring the time of flight of an acoustic pulse the nodes can make estimates of their distances from each-other. This requires capabilities including local data processing, local data validation and synchronisation. It was found that the standard uncertainty in a single ranging estimate was approximately 3.5 cm and that different node pairs had different offsets. The nodes were all nominally the same, so these differences in measurement offset must have resulted from manufacturing differences in the hardware. It was also found that the on-board software was very unstable, so minor differences in hardware could have resulted in large differences in ranging estimate. Despite these limitations and problems, it was possible to calibrate the system to remove the offsets. With the aid of the *Largest Consistent Sub-Set* technique developed within the National Measurement System's Software Support for Metrology programme it was possible to use the sensors for a collaborative measurement task which achieved uncertainties of 0.27 cm. This was valuable as it demonstrated that this data analysis technique could be applied to WSNs and greatly improve the system performance.

This project identified a number of important areas where further work is required. Node self-localisation is vital to a large number of WSN applications, but the current techniques of acoustic ranging are not of sufficient accuracy or robustness. Methods based upon radio interferometric and signal strength measurements show more promise and warrant further work. Interference between different radio protocols and the influence of signal digitisation is becoming an increasingly important issue as the electromagnetic spectrum becomes more crowded. The need for calibration of networks made up of usually cheap, poor-quality sensors is vital for any meaningful measurements to be made. This becomes an increasingly challenging task as networks grow in scale and deployments are made in harsh environments. Techniques for performing distributed calibrations within a network are required. Synchronisation is also an important capability which is in need of improvement for these resource-constrained systems, as it is necessary for many applications and networking protocols. The experimental work within this project to look at radio interference and the resource hungry application of acoustic ranging has not only enabled the above issues to be identified, it has also helped develop vital practical experience and expertise which will help drive continued work in the future.

Contents

1	Introduction	1
1.1	Fundamentals	2
1.2	Existing Work	3
1.3	Key Wireless Sensor Network Themes	5
1.4	Project Scope	6
1.5	Testbed Platform Choice	7
1.5.1	Hardware	8
1.5.2	Software	8
2	Experimental Work 1: Office Monitoring	10
2.1	Method	10
2.2	Results	10
3	Experimental Work 2: Radio Link Quality	13
3.1	Basic Technique and Algorithm	13
3.2	Multipath Effects	16
3.3	Interference Effects	19
4	Experimental Work 3: Acoustic Ranging	23
4.1	Context	23
4.2	Ranging Algorithm	24
4.3	Calibration Method	29
4.4	Calibration Results	31
4.5	Data Filtering and Validation	36
4.6	Outlier Rejection	38
4.6.1	Outlier Rejection Process	38
4.6.2	Outlier Rejection Findings	39
4.7	Acoustic Ranging Conclusions	40
5	Summary and Conclusions	43
6	Future Work	43
7	Acknowledgments	44
8	Appendices	48
8.1	MICAz Data Sheet	48
8.2	Interpreting nesC Component Graphs	49
8.3	Readme for TOSBase application	50
8.4	VB.net Code: Packet Loss Analyser	51
8.5	Readme File from Vanderbilt University on Their Ranging Algorithm	55
8.6	Software for Analysing Ranging Data	58

List of Figures

1	Key WSN Aspects	2
2	The components of a MICAz node	8
3	The constituent components of a nesC program to flash the LEDs on a node	9
4	Simple temperature mapping with multihop routing	11
5	Raw temperature readings	12
6	Top-level components of the numbered message application	14
7	The Reverberation Chamber Layout	17
8	The effects of reflections	18
9	Layout for Interference Investigations in the OMS Lab	20
10	Reception Success Rate when Subject to Different Types of External Interference	22
11	Schematic of the timing intervals for the acoustic sounding and listening. The length of the audio signals and delays between successive chirps are known by the listener, allowing the acoustic time of flight to be estimated [27].	25
12	Components of the adapted Vanderbilt ranging algorithm	26
13	Sounding components in detail (Within OutsideRangingActuatorC in fig 12)	26
14	Listening components in detail (Within OutsideRangingSensorC in fig 12)	26
15	Example acoustic data sampled by the MICAz microphone	27
16	Extracted data for nodes listening to node 1	28
17	Layout for Ranging Calibrations in the Hemi-Anechoic Chamber	29
18	Example of Compiled Averages and Standard Deviations for Ranging from Sounder 1 for a Single Calibration Run	30
19	A set of ranging estimates from 55 cm to 100 cm	32
20	Intermittently large estimates (red) affecting average range estimate. Note that the frequency of these varies for the different sounders, and some such as node 3 remain consistently high.	33
21	Results from ‘Good’ Sounders as Indicated by Linearity of Ranging Estimates	34
22	Corrected Ranging Estimates Between Nodes 1 and 3	35
23	The results of the line fits to all the data for the ‘good’ sounders. This shows the good consistency between the nodes as listeners, as well as the slight differences in offset and gradient from the various sounders.	35
24	Estimates of the start data point of the acoustic pulse within the data buffer, with no error checking or conversion to a range. Note all these estimates are for a single position configuration, making the large spread of values an indication of errors.	37
25	The effect of filtering data with various correction criteria	40
26	Layout for Gathering Largest Consistent Sub-Set Data in the Hemi-Anechoic Chamber	41
27	Descriptions of Different Connection Types	49
28	Software Interface for Analysis of Lost Packets	55

1 Introduction

A Wireless Sensor Network (WSN) consists of a number of distributed autonomous sensors that work co-operatively to monitor physical or environmental quantities. Each node consists of a sensor or multiple sensors, local processing, data storage components, a power supply and a radio transceiver. Networks can be homogeneous with all nodes being the same, or heterogeneous if specialist tasks and capabilities are required at certain positions. There has been increasing interest in WSNs over recent years as improvements are being made in all the constituent technologies. As the capabilities of such systems increase a wider range of applications becomes feasible. These include terrain mapping, environmental and disaster monitoring, military surveillance, personal context identification (for multimedia services), asset tracking and security monitoring. Some of these are being applied now, and others require the development of advanced capabilities such as embedded intelligence, *ad-hoc* routing and high-level decision-making.

As these systems start being used for increasingly safety- and economically-critical applications, the need for a detailed understanding of network confidence and of uncertainties in network becomes vital. These platforms differ significantly from traditional sensing systems. The opportunity to carry out spatially distributed measurements involving numerous levels of data processing and fusion means that new analysis and calibration techniques need to be developed.

This project, through a review of existing work in the area, combined with practical experimental work has achieved the following objectives:

- Developed practical expertise and capability in the area of wireless sensor networks
 - identified the key measurement issues to investigate
 - identified the sources of uncertainty
 - identified the key differences between these and traditional sensing systems
 - carried out practical experiments to begin to quantify and characterise the sources of uncertainty and failure
 - used novel data aggregation and filtering techniques to overcome some of the system limitations
- Demonstrated this interest and expertise both within and outside NPL
- Helped define future projects in the area

This report is structured as follows: Section 1.1 describes in detail the fundamental components of a wireless sensor network system. Section 1.2 contains a brief literature review of existing work in the area, as well as a description of the perceived requirements and needs of end users. Section 1.3 lists the general issues and themes involved in all WSN systems and categorizes some as key measurement issues. The choice of which of these to investigate and the justification for these choices are given within section 1.4. Choosing a testbed with suitable functionality and adaptability was also a significant part of this project and is described in section 1.5. The initial practical investigations carried out in an office environment and detailed in section 2 are followed in section 3 by a description of a study of radio link quality when the network is subjected to various interferences and environments. The bulk of the

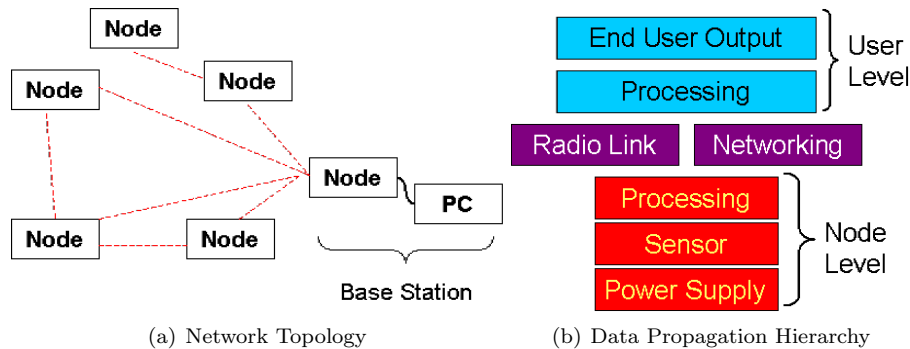


Figure 1: Key WSN Aspects

practical work involved a case-study application which looked at the functionality of the system when performing the resource-hungry application of acoustic ranging, described in section 4. The use of data validation and fusion techniques on the outputs of this application are described in section 4.5. Finally this report draws conclusions from the reported work and makes recommendations for the future.

1.1 Fundamentals

A Wireless Sensor Network (WSN) comprises a number of nodes that can all communicate wirelessly. A node comprises a power supply, on-board processing, dynamic memory storage, data storage, sensor(s) and a radio transceiver. A network could be made up of identical nodes (homogeneous) or different nodes with different capabilities (heterogeneous). This description could cover a large range of scales. Networks may comprise a few nodes, or thousands if the protocol were able to accommodate them. The functionality could also vary widely from ‘dumb’ nodes which simply stream data back to a central point, to highly intelligent systems with collaborative data analysis between nodes to exploit redundancy and provide higher level outputs. Despite this wide range of potential types of system, there remain common components and issues to be considered (see figure 1(b)).

The key aspects and their effects on measurements made by WSNs are described in more detail in section 1.3. Fundamentally all wireless sensor networks will have the components shown in figure 1(b), and depending on the application, the different components will have different levels of importance. For example having a suitable power supply may be a vital consideration when installing a long-term environmental monitoring system whereas for short-term disaster monitoring sufficient power supplies will be less of a difficulty. The user output is the overriding factor that affects the choice of the other system components. Outputs could be purely raw data, a triggered alarm or some kind of actuation for a control system. These would all require different amounts of data processing and autonomy from the system.

The wide range of potential applications which could be fulfilled by WSNs include the following:

- Environmental monitoring
- *Ad-hoc* battlefield deployment
- Short-term disaster monitoring
- Production-line monitoring
- In-home ‘well-being’ monitoring
- Body sensor network health monitoring

As this list shows, WSN systems can cover a wide range of scales and levels of autonomy. The current state of the technology means that some of these are closer to being realised than others. This is discussed further in section 1.2.

1.2 Existing Work

The early stages of this exploratory project involved the investigation of existing work being carried out on WSNs. This was carried out through a combination of a literature review and industrial and academic visits. It was found that there was a large body of work which concentrated on the software programming language considerations involved in wireless sensor network systems. One of the main languages in use is nesC [36], which is implemented with the TinyOS operating system. This language was developed at the University of California Berkeley. It tries to embody all the limitations and restrictions caused by resource constraints in wireless sensor networks, by being modular and customisable for specific applications [16] (nesC is described in more detail in section 1.5.2). This has been widely adopted as the standard operating language for WSN systems. Using this language for higher level system functionality makes up one of the most significant areas of work currently being carried out in WSN development.

Networking and messaging functionality constitute a large part of the work carried out using nesC and many researchers have devised both generic and application-specific techniques [19]. P Buonadonna et. al. [7] describe an active messaging format which can dynamically change depending on the application requirements. Such approaches aim to maximise efficiency by minimising radio and processing overheads. Closely linked to messaging is the area of networking protocols. There is a large library of work focusing on the implementation and theory of ad-hoc, dynamic multi-hop routing. One solution involves position-based routing [24], where the path of data throughout a network is chosen based on the spatial separations of the nodes, allowing the number of hops between the source and destination to be minimised. This has the advantage of being efficient in terms of the number of required transmissions, however, the ability for nodes to self-locate still needs to be developed to an adequate precision. Many other routing protocols, such as Surge developed by Crossbow [39], rely on a combination of data on received signal strength, node battery levels and reception success rates to build up a picture of the quality of links between all the node pairs. This information is then used to choose the best route for a particular message. This has the advantage of being able to spread the communications burden throughout a network (assuming that there are no spatial choke points). The downside is that there are relatively high overheads in terms of processing and the need for ‘housekeeping’ messages. The majority of work being carried out in the area of routing is through simulation. This has the

advantage that networks of thousands of nodes can be simulated without all the associated costs and difficulties of setting up a real-world system. A downside is that such simulations involve many assumptions about the propagation channels and equipment functionality.

A limited amount of practical work has been carried out on networking and routing issues. For example [17] involves testing algorithms designed to minimise transmission powers, while maintaining robust data links. This not only saves battery power, but also reduces the likelihood of adverse interference between nodes of a network. The authors found that the energy usage was successfully reduced, but that the routing protocols were not always robust. The term ‘Network Services’ is used to describe higher-level functionality of WSNs which involves a number of resources and adds capabilities which are important for a number of types of application. Synchronisation is one such network service which is required for applications such as target tracking, localisation, some networking protocols and other monitoring applications. This capability requires a trade-off to be made between the level of accuracy and the messaging and computational cost [25]. Such trade-offs are common in WSNs and it is often the application requirements that determine the balance between accuracy and use of resources. [19] describes some of the synchronisation protocols implemented for various networking and data gathering applications. Issues also arise for large networks and errors in synchronisation generally scale with the number of hops [20].

The ability of nodes to self-locate is vital for any mapping or monitoring application which uses ad-hoc deployment such as military surveillance or precision agriculture [27]. The majority of solutions to this problem involve the multi-lateration of ranging estimates [2]. These ranging estimates may be based on the received radio signal strength, which is attractive to system designers as it can be added to existing radio stack software without the need for additional hardware. Unfortunately in real environments, this ranging method is not accurate enough for satisfactory localisation [11]. More accurate methods are based upon the time of flight of acoustic or ultrasonic pulses [2] [27] [31]. There is a large amount of work in this area and development of these network services has also catalysed the development of other lower-level capabilities such as high-rate timers and on-board data analysis techniques [1].

Despite a large proportion of the existing work being relatively abstract and limited to specific components or functionalities, there is some work being done at the application level. Much of this is in the area of environmental detection and monitoring [29] [22]. Such application-based work has provided a significant catalyst to the development of WSN capabilities owing to the targeting of efforts. There may be the drawback that some of the developments are not generalisable, but the ‘whole system’ approach is providing significant developments in WSN capabilities.

Informal consultation with industrial parties including ARUP [33] and Applied Nanodetectors [34] showed that a key concern was the ease of installation and use of any WSN monitoring system. The need for knowledge of programming or electronic engineering was a significant barrier to their adoption of WSN systems. The reliability and robustness of the systems both in terms of measurement accuracy and frequency of failures was also of paramount concern. Cost-benefit and risk analysis are commonly used when choosing new asset monitoring systems, but currently WSNs are not sufficiently understood for meaningful assessments to be made.

1.3 Key Wireless Sensor Network Themes

Based upon the review described in section 1.2 a number of common, general themes were identified. These were then categorised by whether they *directly* affect the measurement process or not. Measurement issues are those which lie directly in the data path (as illustrated in figure 1(b)). Non-measurement issues are those which don't necessarily affect the data, but need to be accounted for as part of the whole system as they may lead to failure modes or spurious system outputs.

Measurement Issues

- **Sensors:** All measurements originate from some form of sensor. Their behaviour could be affected by factors such as power supply, temperature or interference. To create large networks economically, the sensors are likely to be cheap and relatively low quality. This brings in issues of data fusion in which the aim is to improve confidence, by combining measurements from many poor sensors (which could be carried out within the network, or at the base station). The potential for heterogeneous networks of different types and qualities of sensor should also be considered. Some sensor-generated artefacts in the data are observed in section 2.
- **On-board Processing:** Radio communications commonly constitutes the largest power overhead of a WSN system. This makes it prudent to send the minimum possible number of packets, with no loss of information. This can be achieved by carrying out some data processing on board the sensor, either to give a higher level output such as a triggered alarm, or simply to filter out redundant data. A number of uncertainty issues arise from the fact that in many cases, the user will only have access to the processed data.
- **Radio Link Quality:** The wireless nature of the data link gives rise to a number of effects, which can have different influences on the system. Interference from external sources could cause data packets to be lost and necessitate re-transmission. The data within the packet could also become corrupted or changed by the interference. Confirmation of reception may be required, as transmission cannot be guaranteed. A trade-off must be made between battery life and communication link quality. In the increasingly dense wireless environment, these issues become ever more important. Some of these effects are investigated in section 3.
- **High-Level Processing:** In many monitoring applications the customer is only interested in high-level information such as an alarm signalling deviation from set conditions. Users are rarely interested in the raw data so it is necessary to condense this down to the meaningful information, with known confidence levels. The difficulty of this data fusion is exacerbated by the vast amounts of data often produced by even modestly sized networks. Examples of this type of processing are discussed in section 4.5.
- **Timing and synchronisation:** For applications such as long-term status monitoring, it is important to have sufficiently accurate knowledge of the time at which measurements were taken. For real-time applications such as target tracking or location, very accurate synchronisation may be required. Some routing and networking protocols also rely on an accurate 'global' time to be maintained. This can be complicated by

the fact that the transit time for a packet of data is not always easily determined (processing, system overheads, routing and radio traffic at the base station could all incur a delay). The propagation of timing synchronisation information throughout a network is also a challenging metrological issue.

Supporting Issues

- **Routing:** In many configurations and topologies not all sensor nodes can link directly to the base station (or it is inefficient to do so). In this case data must be routed through intermediate nodes. This could cause the data to be changed and will also increase the burden on the routing nodes. The routing protocols themselves will require data gathering and processing in order to judge the quality of radio links. Some issues relating to this have already been observed and are discussed in section 2. Complications may arise if nodes join or leave the network, or if the signal propagation characteristics of the environment vary over time.
- **Scalability:** It is envisaged that future networks will be made up of thousands of nodes. Networks of this scale will produce vast amounts of data, and the routing overheads could quickly swamp the system. This issue is often investigated by simulation.
- **Power Supply:** The limitation of power supply has far-reaching connotations when it comes to WSN systems. The whole ethos is built around the fact that the resources are limited for broadcasts, actuation and processing. This limits the functionality of any system. A power supply being shared by all the operations on a node can also have an effect on sensor readings and cause erroneous results (as seen in section 2). The inclusion of sleep modes can save energy, whilst introducing potential problems of wake-up synchronisation across the network, and long latency in reacting to stimulus.

1.4 Project Scope

The technology of wireless sensor networks encompasses a large range of disciplines including software engineering, electronic engineering, wireless communications and signal processing. As described in section 1.3, the wide variety of important issues cover all of these areas. In choosing which of these important issues to investigate, the following criteria were used:

1. Is it important to a large number of applications?
2. Can NPL make improvements or add value in this area?

Based upon these criteria and the original studies of the area, it was decided that the quality of the wireless radio links would benefit from further study. This would also benefit from drawing on the experience at NPL in carrying out Electro-Magnetic Compatibility (EMC) testing. Factors which can effect wireless communications links include the reflectivity of the environment (leading to interference and multi-path effects) and the level of other electromagnetic signals within the environment from external sources. The details of the work carried out in this area are discussed in section 3.

Another important finding of the investigation of existing work and issues was the level of integration of WSN systems. For the majority of applications, every component will have

some effect on the output of the system. These components are likely to interact together in ways which are not predictable by simulation and in fact, it is virtually impossible to investigate separately the issues described in section 1.3. For these reasons it was decided to also look at the performance of a WSN system when performing a task which utilises a broad range of the available functionality. Acoustic ranging was chosen as the demonstration application. This is a resource-heavy application, and is also an important capability for many applications. The justification for this choice, and details of the work are described in section 4.

1.5 Testbed Platform Choice

As with choosing areas for investigation, choosing equipment for this project also involved a number of key criteria:

1. Is the equipment flexible and capable enough to investigate the chosen areas?
2. Is the equipment consistent with what is available to industry and end-users?
3. Is it within the resources and capabilities of the project to set up the chosen testbed equipment?

Below is a list of the key considerations involved in choosing a wireless sensor network platform. These focus on the requirements for this project, but they also apply to choosing equipment for real measurement applications.

- **Radio Protocol:** Each protocol has its technical pros and cons. Issues include data rates, power consumption, support of large numbers of nodes and types of network topology supported. Flexibility with regard to node functions, data types and level of customisation is also important. Although many networking issues could be looked at using any of the protocols, some offer more flexibility than others. It is also prudent to choose a protocol that is widely used, and has a future.
- **Internal Processing:** Although ‘dumb’ sensor nodes are available which transmit a constant stream of data, on-board microprocessors are now commonplace and offer a number of advantages. These include the ability to perform data analysis, data combining and information routing operations on the fly. Considerations include the processor speed, ram and power consumption. Often the operations such as networking and power management must be juggled with the data analysis operations.
- **Sensors:** The issues being looked at in this project are predominantly to do with the network itself. There are some very important factors in choosing the sensors, however. The physical property must be well understood, easily controllable and of use in a real world situation. Possibilities include acoustic, electric field, temperature or light sensors. Generally such WSN systems will be mounted with cheap inaccurate sensors. The output quality of such systems, therefore, is likely to be dependent on various levels of data analysis and error checking.
- **Software:** The software issues can be divided into two parts. The most fundamental part is the on-board operating system. This controls and accesses all the node

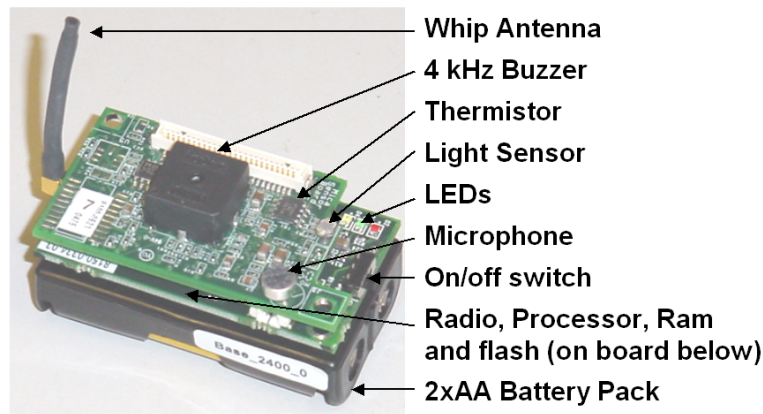


Figure 2: The components of a MICAz node

hardware. The quality and flexibility of the operating system determines what can be done with the testbed. The other software aspect is the higher level processing. This may be possible with generic data analysis tools, but the issues of connectivity, compatibility and controlling network operations from the central computer must all be accounted for.

- **Size of Supplier:** Although not a technical issue, this is important for a number of reasons. Firstly there are the indications of product quality and lifetime implied by the quality of the supplier. In addition, the number of other groups working with a given platform opens up a large database of potential applications and activities which could be drawn upon.

1.5.1 Hardware

Based on all the considerations discussed in section 1.5, Crossbow MICAz nodes were chosen. These operate on a ZigBee [41] compliant radio protocol which operates at 2.4 GHz (a consortium including Siemens, Texas Instruments and Philips hold the current specification). This is designed for large networks with low power consumption and is one of the most widely adopted communications protocols for WSNs. The nodes also come with an on-board suite of sensors to measure temperature, sound, light levels, acceleration and magnetic fields. For a full specification of the platform see section 8.1. The kit chosen includes seven nodes and a base node for connection to a personal computer (PC).

Figure 2 shows one of the nodes, the dimensions of which are approximately 60 mm × 30 mm × 25 mm. These are commonly used as a research platform and are likely to require adaptation and development in order to be used in real-world or critical applications.

1.5.2 Software

A predominant reason for choosing the MICAz system was its use of the TinyOs operating system, which uses the specialist programming language nesC. This language is specifically

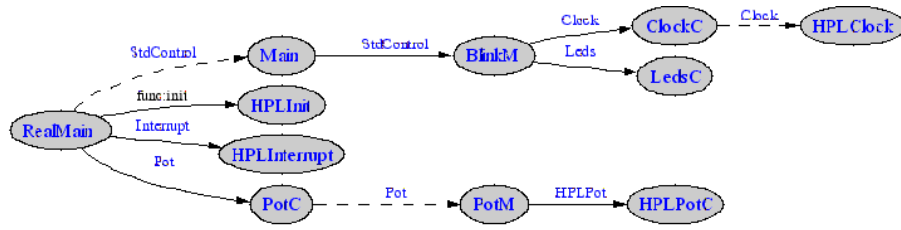


Figure 3: The constituent components of a nesC program to flash the LEDs on a node

designed to account for all the limitations of the WSN systems. It includes accessible hardware abstractions to allow for customisation, it is very efficient in terms of memory and processing requirements. Real-time radio reception and data gathering are also robustly dealt with¹. The structure of a simple program is shown in figure 3. A description of nesC component graphs can be found in section 8.2.

Figure 3 shows how the operating system only includes the components that are necessary for the application (these must be selected or created manually), maintaining efficiency. There is a relatively large number of components required for such a simple LED flashing application including components to control a running timer and the LEDs themselves. When more involved applications are considered, the situation becomes increasingly complicated. One drawback of this operating system is the time required to create programs for the nodes. Some of the low level hardware abstractions are sufficiently generic to make use of, but anything more specialist must be written from scratch. This problem could be overcome by the fact that there is such a large community of people working with this software [36]. The access to libraries of established knowledge and application software meant that the progress of this project could be significantly accelerated by building on existing work.

¹Concurrency has been subject to vast amounts of research in the computer sciences [26]

2 Experimental Work 1: Office Monitoring

The initial stage of this project employed the pre-installed software to carry out some distributed temperature sensing within an office environment. This software periodically samples the temperature sensor on the node and then transmits the data back to the central base-station. This software also includes a mesh-networking component which performs *ad-hoc* routing of the data. The aim of this part of the project was to become familiar with the equipment, assess the quality and functionality of the pre-installed software and provide recommendations for improvements and future work. It was found that, although relatively easy to set up, the software had some problems both with sensing and routing.

2.1 Method

The MICAz developer kit comes ready installed with a basic application called ‘surge 2420 hp’ [37]. This application has the functionality to stream back sensor data on the temperature, light level, accelerations and reference voltage measured by the MTS300/310 sensor boards attached to the nodes. Where this application comes into its own is the inclusion of a module called ‘surge reliable’. This component enables the nodes to carry out *ad-hoc* mesh networking. By intermittently sending out ‘housekeeping’ message packets, signal strength and packet loss rate information is used to generate a Link Quality Indicator (LQI). Based on this LQI the routing algorithm chooses the best path for messages to take. A limitation of this algorithm is that it is inefficient with regard to power and communications overhead. It can also only handle messages being sent from the network to a single base station node. It is possible to inject command messages with this routing system (for example to tell all the nodes to change their sample rate), but only broadcasts from the base station can be made. Individual nodes cannot be targeted. The algorithm itself is also prone to some inconsistencies in that it sometimes switches the routing for no apparent reason. This is likely to be caused by the LQI metric not being robust with respect to minor perturbations of the system.

Another benefit of ‘surge 2420’ is its compatibility with a program called Mote-View [38]. This pc-based software allows the data gathered from a network of ‘surge 2420’ programmed motes to be collected and easily displayed.

To study the functionality of the ‘surge 2420’ software within an office environment, the nodes were distributed as shown in figure 4(a) and set to stream temperature data back to the base station over a three-hour period. The positions were chosen to be relatively uniformly spaced, within the constraint of available desks and furniture. The nodes were placed either on desks or shelves at roughly the same height (around 1.2m off the ground). The exception was node 7, which was placed in a filing cabinet drawer. The data were displayed using Mote-View and also written to a database for more detailed analysis.

2.2 Results

Figures 4(c) and 4(d) represent the distribution of temperature as measured by the sensors, and estimated by the visualisation software (these are best viewed in colour in the electronic version of this report). By extrapolating adjacent sensor readings the distribution through-

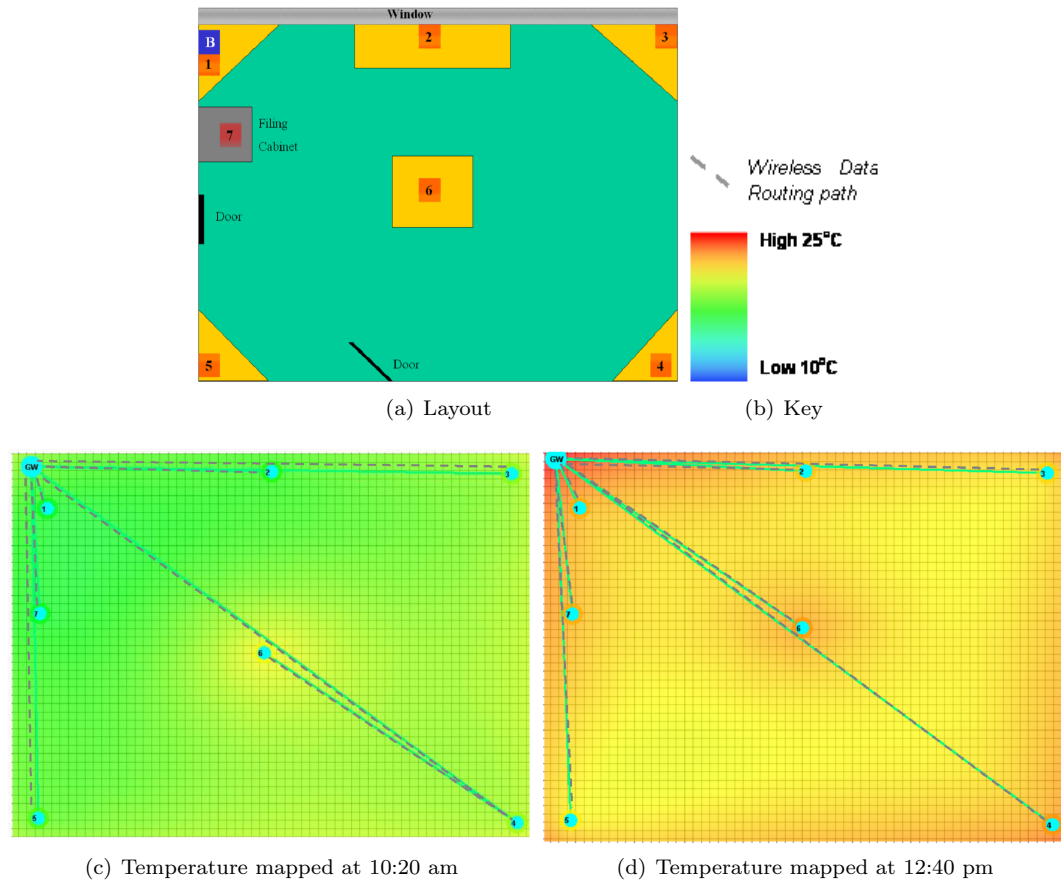


Figure 4: Simple temperature mapping with multihop routing

out the whole room was estimated (as shown by the colour gradients). It was found that the visualisation software was very limited, and often led to questionable outputs. The position of the sensors on the plots was set manually to reflect their physical positions. The base station was able to reconstruct the routing paths based on source and destination information contained within the data packets.

From figures 4(c) and 4(d) it can be seen that the room was generally warmest in the centre and cooler near the walls and windows, and that the room warmed by around 3 degrees throughout the morning. The gateway also streamed its own temperature data, although a sensor board was not attached. There is no failsafe to alert that the data being requested is an artefact from the analogue-to-digital converter rather than from a genuine sensor. This is the cause of the apparent hotspot around the gateway node in figure 4(d). A better estimate of the thermal field distribution throughout the room could be achieved by a more uniform and even dispersal of sensors. Attaching nodes to the ceiling could avoid the constraints of furniture and other obstacles.

The routing (shown by the dashed lines) also displayed a number of interesting features. At 10:20 am the data from node 6 was being routed through node 4. This may seem counter-

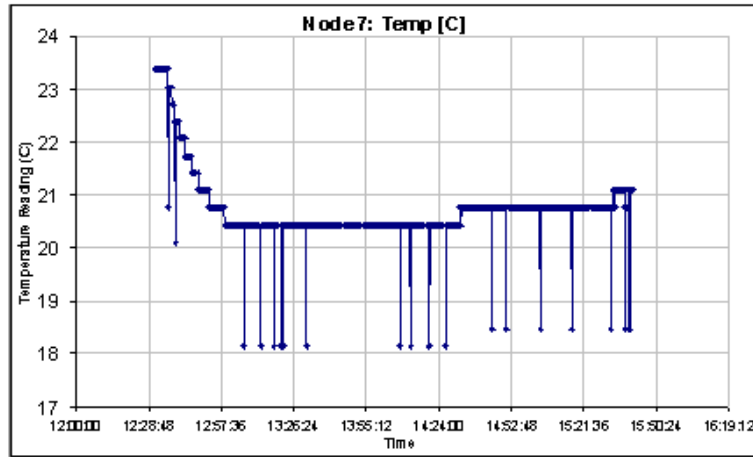


Figure 5: Raw temperature readings

intuitive as it more than doubles the effective path length for the signal. The link may have been affected by interference from electrical equipment, multi-path effects or obstacles, which would alter the link quality estimation of the routing software. Another possibility is that the routing software made an error; at times it has been unreliable in establishing and maintaining links. By 12:40 p.m. the routing had reverted to a star topology (i.e. all nodes streaming directly to the gateway).

Figure 5 displays the temperature reading from a single node over a three-hour period. The temperature is measured using a thermistor with a fixed reference voltage (see section 8.1). The power is supplied by two AA batteries, and must be shared by all the processes being carried out by the node. The effects of radio stack activity on the reference voltage (and hence the sensor output) can be seen from the periodic 2°C drops. Similar behaviour was observed on the responses of all the nodes, with a comparable number of similar 2°C drops. A likely solution would be to re-write the data-gathering program. It would be necessary to include a checking method, which would prevent data being gathered during the sending or receiving of a message². Alternatively, if this effect is anticipated, then the data could be filtered at the sensor to remove the erroneous data before it is sent to the base station. Even within this simple study the problems of data routing, node placement and resource constraints (in the form of limited power supplies) have been highlighted. These preliminary investigations also showed that the pre-installed software was very limited both in functionality and stability. This is similar to the findings of one of the companies consulted during the early stages of this project, ARUP [33].

²This problem has been overcome in subsequent work on distributed temperature measurements. The solution was explicitly to shut the radio off unless the system was between sampling periods. Ongoing work in this area is described in [9] and section 6

3 Experimental Work 2: Radio Link Quality

The radio link quality throughout a WSN has a significant effect on its performance. Data may be lost or corrupted, measurements may have to be broadcast repeatedly. This may cause problems for alarm or control applications where timeliness of the information is paramount. There are a number of effects specific to such systems resulting from the potential for mesh type configurations, and large number of transceivers likely in a relatively small area. The robustness of the MICAz radio was investigated using a method of streaming numbered message packets and measuring the reception success rate. This was carried out under a number of conditions and it was found that the platform is highly susceptible to multipath effects, and different types of external noise have varying degrees and modes of influence.

3.1 Basic Technique and Algorithm

There are a large number of potential metrics for the quality of a radio link. These include the signal strength, signal to noise ratio and packet loss rate. Factors such as signal strength may have an effect on the power efficiency of a system as well as potential interference between nodes within a network. However, when data are being gathered, the main concern is whether or not a sent message is faithfully received at the chosen destination. The radio protocol used by the MICAz nodes includes an element of error checking in the form of a Cyclic Redundancy Check (CRC) [30]. This is a form of checksum which can be used to identify whether any of the data within the packet has changed in transit. If this CRC test is failed then the packet is rejected at the receiver. Hence any level of data corruption will simply result in the packet being discarded. For this reason it was decided to use packet reception rate as a measurement of the quality of a radio link. This counts the total proportion of packets which are faithfully received at the destination node. The basic equations that define reception success rate are shown below. If all the packets are received the success rate is unity and loss rate zero, and vice versa if no packets are received.

$$\text{Reception Success Rate} = \frac{\text{Packets Received}}{\text{Packets Sent}} = \frac{\text{Packets Received}}{\text{Broadcast Rate} \times \text{Broadcast Duration}}$$

$$\text{Packet Loss Rate} = 1 - \text{Reception Success Rate}$$

In order to investigate the packet loss rate in various situations or environments, a relatively basic application was written. Once installed on the nodes, they would begin to broadcast a running count (along with their own identification number). The counts ran cyclically from 0 to 255. By looking at the message counts it was easy to see any messages that had been lost. As an extra debugging tool, they were also set to flash an LED after every successfully sent message. The constituent components for this application are shown in figure 6 (Note: for conciseness some of the unused functionality of this program is not shown).

The base-station setup for these investigations comprised a number of parts. A node was programmed with an application called TOSBase (see also section 8.3) and connected to a MIB510 interface board, which was in turn connected to the RS-232 port of a laptop computer. TOSBase is a basic forwarding program which receives messages over the radio and forwards them to the serial port. A simple PC-based program called XListen was also

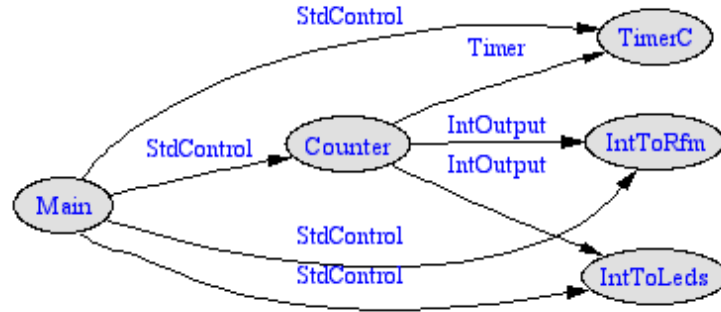


Figure 6: Top-level components of the numbered message application

used. This takes any data received over the serial port and writes it to a file. The benefit of this method (rather than trying to do any real-time loss analysis) was that it allowed the raw data packets to be saved and any subsequent analysis could be verified manually.

A Visual Basic.net program was written to analyse the stored data files to look for the number of packets lost. Firstly it would sort all the data lines by node ID and write the data for each node to a separate file. It would then go through each file looking at the running count contained within each message. If there were any missing or repeated packets (as indicated by non-consecutive or repeat counts) then these would be flagged and written to another analysis file. An example of stages that the data went through is shown below (note that the different stages are not necessarily the same data set in this example. They have been chosen to demonstrate a number of different effects). The programs themselves are shown in section 8.4 of this report.

```

7e42ffff0a7d5d1a050013010100de01e501d901db01de01e001e301e301dc01e2015716 [36]
7e42ffff0a7d5d1a060005010100f401ee01e701f301ea01ed01ef01ee01ef01f1019251 [36]
7e42ffff0a7d5d1a04003b010100ee01ed01e901e901e801ec01ed01ee01ea01e701b087 [36]
7e42ffff0a7d5d1a0600fc850100e101e701ea01ee01e301df01e401ee01ee01e401(7d5e)06 [37]
7e42ffff0a7d5d1a04003d010100e601e901ee01ed01e701e301e501ed01e601e901cb14 [36]

```

This is a stream of raw messages as recorded by XListen. Each line is a new data packet. 7e42 signifies the start of a packet. The breakdown of the packets and meanings of all the fields can be found in [30]. Note that a measurement run may be made up of thousands of packets. The different length of the third packet is due to the data payload containing a '7e'. This number is reserved to signify the start of a packet. If the payload needs to represent this number it uses '7d5e' instead, causing the packet to be slightly longer. This is highlighted within brackets.

```

251,fb,06,7e42ffff0a7d5d1a(06)00(fb)850100e801ea01ef01eb01e201eb01e101ed01e801e801c7b3 [36]
253,fd,06,7e42ffff0a7d5d1a(06)00(fd)850100d801f301de01eb01db01e901e201f001da01ec01f38c [36]
252,fc,06,7e42ffff0a7d5d1a(06)00(fc)850100e101e701ea01ee01e301df01e401ee01ee01e4017d5e06 [37]
254,fe,06,7e42ffff0a7d5d1a(06)00(fe)850100dd01ec01da01ea01e801ec01ea01e401e701ea01561d [36]
255,ff,06,7e42ffff0a7d5d1a(06)00(ff)850100ee01ea01ec01e901e001ee01de01f201e501e5013707 [36]
000,00,06,7e42ffff0a7d5d1a(06)00(00)860100ec01e701dc01e401e901e601ed01ec01e101ef015487 [36]

```

This is a set of packets from node six. The columns are: Packet Counter Number (converted to decimal), Packet Counter Number (in original hexadecimal), source node ID and the raw packet itself. The position of the counter and source node ID within the raw packet have been highlighted with brackets.

```
-4,050,32,05,7e42ffff0a7d5d1a0500322d01004b02e101cc019401df0112022102de0110020b024d4f [36]
-3,083,53,05,7e42ffff0a7d5d1a0500532d0100d501ed01f601e901d701ee010e02fa01f701ea017f56 [36]
00,125,7d,05,7e42ffff0a7d5d1a05007d5d2d0100f101f001e601ea01eb01f201db01f401ea01e901430c [37]
00,125,7d,05,7e42ffff0a7d5d1a05007d5d2d0100f101f001e601ea01eb01f201db01f401ea01e901430c [37]
-2,125,7d,05,7e42ffff0a7d5d1a05007d5e2d0100e901ec01ee01f101f801e901e401e301e001e0011403 [37]
-2,127,7f,05,7e42ffff0a7d5d1a05007f2d0100e701f301ec01ec01e201ed01dd01f901e901d20184b8 [36]
*****
---STATS---
Total Packets Received: 8321
Single Packets Lost: 43
Larger Loss Events: 12
Total packets lost (buggy): 79
Average Loss size: 1.4363
Packets Copied (buggy): 33
*****
```

This shows the output of the analysis program. Having scanned through all the received packets from a given node, the positions of any non-consecutive counts are printed to a file, along with the offending packets. The numbers at the start of each row show the size of the difference between the counter of consecutive packets. -1 represents packets that are consecutive (so are not flagged up by the algorithm), -2 represents the loss of a single packet. A count difference of zero means that the same packet has been received twice (which may occur in environments prone to multipath effects). The ‘buggy’ note is a reminder that the transition from 255 back to zero is not dealt with very robustly by this software. If a packet loss were to occur either side of this transition then the number of lost packets would be over-estimated by around 254 (depending on the size of the actual loss). This error would be easy to spot because of the inclusion of the raw packet data in the analysis output.

It should also be noted that this method makes the assumption that the counter on-board the node is robust. This isn’t always the case. It was observed that if the packet broadcast rate was set too high, then the counter became unreliable. This is likely to be due to the latency of sending a message. If the timer fires and the node tries to send a message while another is still being broadcast, then the counter may increment without a message being sent. For this reason the broadcast rate of the nodes was set to be significantly below the maximum at 307 packets per minute. This also meant that the receiver was not swamped when all seven nodes were broadcasting counters simultaneously. This rate was still sufficient to give relatively large data sets in a relatively short space of time. A number of issues relating to the timers and this sort of internal synchronisation were observed; these are discussed in more depth in section 4.2.

The above method had to be modified for situations where there were packet loss rates above 25 percent. This was because of the reliance on adjacent numbered packets to judge if any had been lost. If a large number of consecutive packets (greater than 254) were lost then the returned loss count would become ambiguous. This could have been overcome by redefining the message structure to allow for a larger counter. For simplicity, another method was chosen. As the packet broadcast rate was known for all the nodes, the total

number of packets sent for a given run could be found by measuring the duration of the run. So, by finding the difference between this and the number of received packets it was possible to calculate the number of packets lost. In fact it was this method that was used for the majority of the measurements. The above analysis software was used to look at lower loss situations and was valuable in finding any repeat reception events.

3.2 Multipath Effects

Based on previous related experience and the literature survey in section 1.2 the following effects were anticipated when our system operated in a reflective or multipath environment:

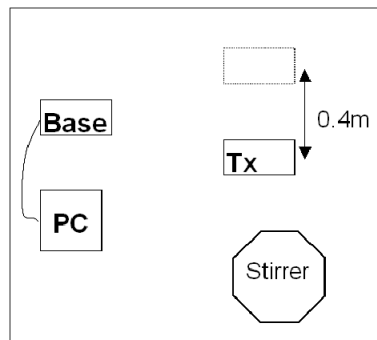
- Poor signal to noise ratio at the receiver
- Position dependent interference
- Repeat reception of packets
- Packets arriving in the wrong order

In a real environment such as an office, it would have been very difficult to have a repeatable amount of reflectivity of the radio signals owing to people moving around and other effects. Such environments are also subject to their own sources of interference such as electrical equipment and mobile phones. For these reasons it was decided to use the NPL electromagnetic reverberation chamber [3]. Not only did this provide a repeatably reflective environment, with some control offered by the movable reflective stirrer, it was also shielded from any external RF noise sources.

The process described in section 3.1 was initially used to investigate the behaviour of the radio within the reverberation chamber. Preliminary investigations showed that at a node-base separation of 1 m there was no successful radio communications whatsoever. It was also noticed that the communications improved significantly while there was someone in the chamber. This showed that the chamber introduced so many multipath effects that the receiver was swamped by all the asynchronous receptions of the messages. It was also possible that the transmitting node may have got ‘locked’ in receive mode because it was seeing reflections of its own message. By having someone in the chamber, a proportion of the radio transmission was absorbed, reducing the level of ambient reflected signal and hence improving communication levels between nodes. In order to recreate this repeatably, a box of Radio Absorbing Material (RAM) of approximately 1 m^3 was placed in the chamber (shown in figure 7(b)).

Using the methods described in section 3.1 the packet loss rate was measured at a number of positions in the chamber. Node 4 was placed at approximately 2 m from the base station and it was then moved at 2 cm increments along a line between 2.4 m and 2.6 m from one of the chamber walls, as indicated in figure 7(a). Although a relatively arbitrary choice of position and area to map out, this still provided information on the spatial variability of the multipath interference effects. This was subsequently repeated with the stirrer operating in continuous rotation of 1.5 rpm.

As can be seen in figure 8, with no stirring the packet reception rate was found to be highly spatially dependent. For the majority of positions the reception rate appeared minimal.



(a) Sketched layout showing relative positions of base node and transmitting node within the chamber.



(b) Chamber with RAM and roughly placed node/base station

Figure 7: The Reverberation Chamber Layout

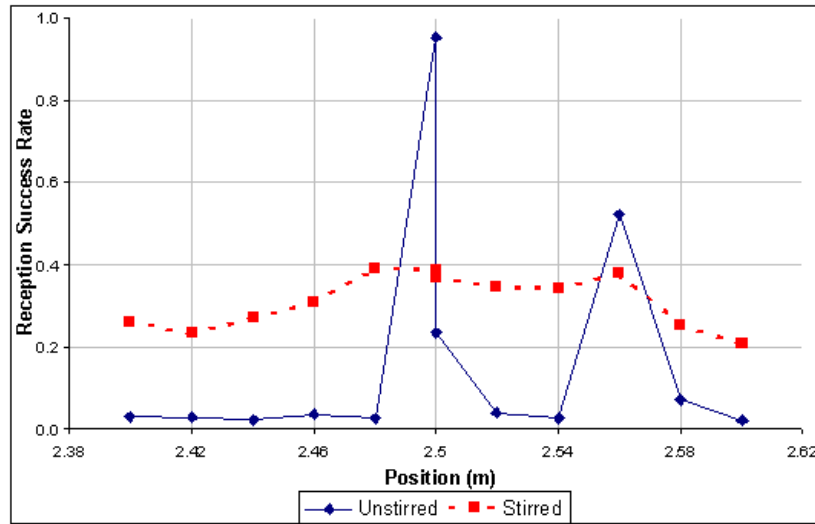


Figure 8: The effects of reflections

However, at 2.50 m and 2.56 m from the wall communications hotspots were observed. This spatial dependence implies that there was destructive interference of the radio signals at the majority of positions investigated. It is possible that smaller-grained position increments may have exposed more hotspots, but this wouldn't have been practical for this preliminary experiment.

The stirrer in the chamber is made up of a number of reflective panels at various angles attached to a spindle. During operation the spindle and panels rotate. This acts to uniformly smear any electromagnetic field within the chamber giving a homogeneous (time-averaged) field throughout the chamber. It can be seen that the stirrer also acts to average out the reception success rate with respect to position. The fact that this average is around 25 percent of the broadcast rate implies that there may be more communication hotspots which were not observed during the unstirred measurements. It is unlikely that the two peaks observed would have been sufficient to account for the level of the smeared average. It also shows that the observed position of the hotspots was due to a combination of reflections from the walls and panels on the (stationary) stirrer.

It would have been possible to investigate these multipath effects in much more detail, had this been within the scope of the project. Possible investigations could have included the modelling of propagation and reflections of the radio output of the MICAz motes to make predictions about the positions of constructive and destructive interference. There could also have been value in measuring the RF signal at various positions around a transmitter to study the potential anisotropy generated by the whip antennas. Mapping out the field generated within a reflective environment could also have proved interesting. There were a number of reasons these were not carried out, not least the issue of time constraint of this one-year project. Also, the aim of this project was to identify and investigate issues which are general to a large number of WSN platforms, hence this was the depth to which these multipath investigations were carried out. In-depth characterisation of the MICAz platform would have been of limited general value. These more in-depth investigations are left for future work and may be more appropriate if specific platforms start being used for safety

critical measurements.

Another key output from this part of the work was that it verified reception success rate as a viable metric of our communication system, and that it could be used to look at the influence of various factors.

3.3 Interference Effects

Another important factor which affects radio communication is the level of compatibility with other signals within the operating environment. This is becoming increasingly important in the dense wireless environment in which many WSN systems will have to operate. Potential interference effects which would be detrimental to communications within a network are listed below:

- Swamping the receiver
- Affecting the electronics (such as inducing currents in the components possibly causing damage or malfunction)
- Affecting transmission (such as making a channel appear busy, causing a node to hold off its transmission)
- Asymmetry could result in which node A could hear node B, but not vice versa

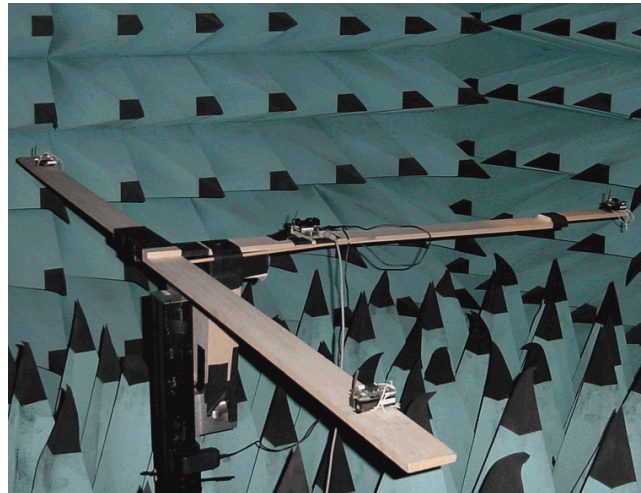
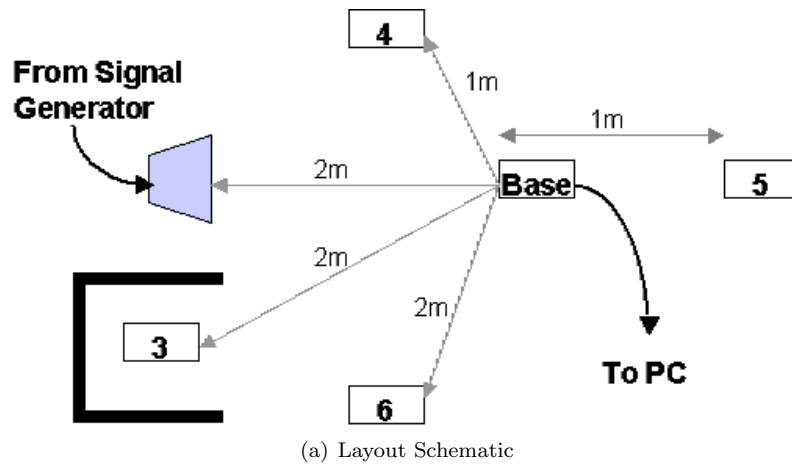
To be able to have a controlled level of externally generated interference and eliminate the multipath effects observed in section 3.2, the NPL OMS³ Lab was used. This is a shielded room with non reflective materials on the walls. External noise sources were attenuated by shielding, and any input noise were relatively easy to characterise or repeat, owing to the removal of reflections from the walls. An HP E4433B ESG-D signal generator was connected via two 3 dB attenuators and 15 m of RF cable to a horn antenna. This was used to input various levels of different types of noise into the chamber. The nodes were arranged as shown in figure 9(a). Although this appears to be an arbitrary layout, it allowed the investigation of effects such as transmission blocking, receiver swamping and antenna anisotropy. Again, the method of counting lost packets described in section 3.1 was used as a way of measuring the quality of the radio link when subjected to a variety of noise sources.

Additive White Gaussian Noise (AWGN) was chosen as the first type of interference to introduce. This is commonly used to simulate background noise of a channel under study, in addition to multipath effects, terrain blocking, interference and self interference that radio systems are subjected to in real environments. Preliminary tests were carried out to find the input noise power levels which resulted in packet losses.

The second type of noise input to the system was a simulated Bluetooth signal. This is one of the most popular protocols currently in use and it operates in the same band as the Zigbee radios used by the MICAzs. This means that cross interference is quite likely to occur.

Note that the following discussion is qualitative and serves only to highlight some of the potential interference effects observed. Giving quantitative results would require the inclusion

³Optically Modulated Scatterer - a technique to measure EM fields whilst causing minimal perturbations.



(b) Layout Picture

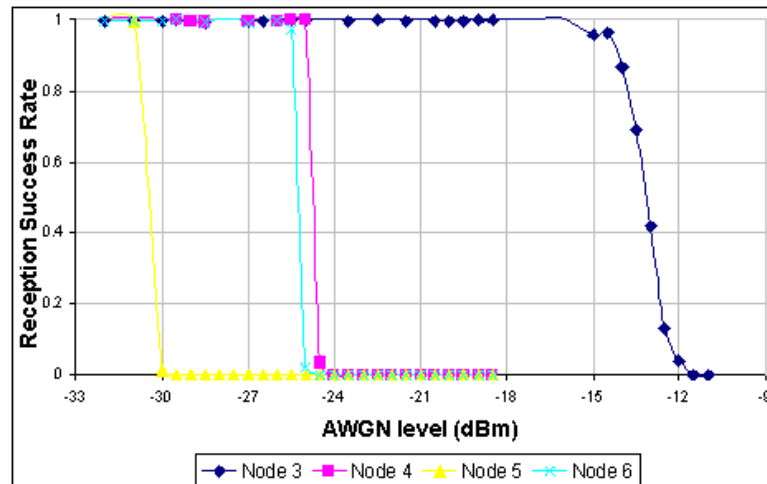
Figure 9: Layout for Interference Investigations in the OMS Lab

of the horn gain profile (for different signal types) as well as the antenna gain for the nodes themselves.

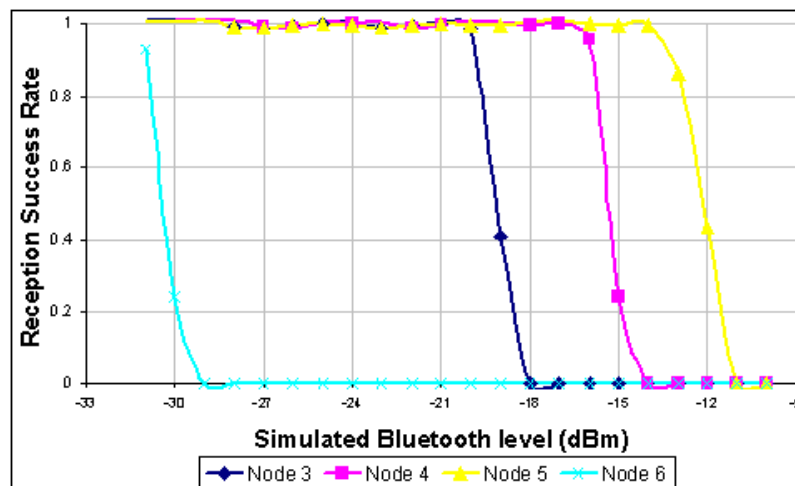
It can be seen from figure 10(a) that the transmissions from unshielded nodes were all blocked by relatively low levels of AWGN. The debug LED which was included in the program in section 3.1 showed that the unshielded nodes were actually being prevented from transmitting (by the lack of a blinking LED to indicate successful transmission). This was possibly caused by the node antennas picking up the noise and anticipating reception of a message. If the nodes switched themselves into receive mode, then they would be unable to transmit a message. Another possibility is that the electronics were being interfered with, affecting the node operation, although this is less likely as the receiver did not appear to be affected at these lower noise levels. The shielded node could be heard despite higher noise levels, until the receiver was eventually swamped.

Figure 10(b) shows that in a Bluetooth environment, generally the unshielded nodes' transmissions could be received at higher levels than in an AWGN environment. The reason for node 6 being blocked at such a low level may have been owing to the anisotropy of its antenna, combined with the choice of orientation. None of the nodes were prevented from transmitting by the simulated Bluetooth signal. This was observed by the LEDs continuing to flash signifying successful transmission. This indicates that this type of noise acts to swamp the receiver rather than having other effects on the electronics of the nodes.

This relatively basic set of investigations has shown that in certain situations this specific platform is susceptible to low levels of ambient RF noise. The effects of the interference have been found to depend on a number of factors such as noise type, node positioning, node orientation and signal strength. Despite having been observed in this artificial situation, these issues are also likely to occur for many real applications. Once platforms start being used more widely in noisy environments, then this will require more quantitative investigation, especially as they start to become relied upon for safety- and asset-critical monitoring applications.



(a) AWGN



(b) Bluetooth

Figure 10: Reception Success Rate when Subject to Different Types of External Interference

4 Experimental Work 3: Acoustic Ranging

4.1 Context

One of the main advantages of wireless sensor networks (WSN) is the fact that, unlike wired sensors, they do not rely on pre-installed infrastructure. Communication links and power supplies are included on each node. This affords the possibility of having a network deployed in an *ad-hoc* manner, for example, battlefield monitoring sensors being deployed from moving aircraft [18]. One of the challenges introduced by this *ad-hoc* deployment is that of self-localisation. For the majority of applications, it is necessary to know the spatial origins of any given sensor reading with some degree of confidence. Traditional GPS type systems are rarely suited to resource constrained WSN systems [27]. Novel techniques for individual nodes to locate each other are required in order to tackle this issue.

It is also important to make the distinction between types of localisation. There is passive localisation of a target. An example of this would be the localisation of a sniper based on analysis of the measured acoustic signal at a number of nodes [5]. There would need to be some method that ensures that the nodes themselves can know their location (either by GPS or manual placement etc.). A similar localisation application is the use of seismic monitoring to locate a moving vehicle. Based upon signal analysis of the distributed seismic data and known node locations, it is possible to locate a target vehicle. These passive target location problems still require the nodes to know their own positions and it is this node self-localisation application which will be investigated further here.

The majority of self-localisation methods rely on the multi-lateration of multiple ranging estimates made between a number of sensors⁴ [2] [6]. It has been shown that position estimation is highly unstable with respect to errors in ranging [13] [23]. So, for such lateration-based location methods, accurate ranging techniques are required. There are a number of techniques based upon Radio Signal Strength Indications (RSSI) [4], time delay of arrival of audible acoustic signals [27] and time delay of arrival of ultrasonic acoustic signals [40].

It was decided to carry out acoustic ranging investigations. There were a number of reasons for this decision. Firstly, the availability of equipment meant that there were platforms incorporating the necessary sounders and microphones, as well as other sensors which would enable other future investigations. Also from reviewing the literature it was found that RSSI ranging is subject to a number of fundamental problems resulting from antenna anisotropy and multipath effects. Finally (and possibly most importantly), the aim of this project was not to find the best ranging or location method. The aim was to investigate fully the functionality and operation of a resource-constrained wireless sensor network. Acoustic ranging utilises a broad range of functionality of the system and encompasses issues such as synchronisation, radio communications, high sample rate sensing and actuation, on-board processing and data analysis. Hence, as well as being a highly useful application in its own right, it also exposes a large number of the general issues relating to WSN measurements (as listed in section 1.3).

⁴There is some work being done on localisation using measured angles between nodes, but it is at an early stage and is predominantly investigated by simulation [8]

4.2 Ranging Algorithm

It would have been possible to write a new ranging algorithm and install it on the testbed nodes. A large number of software components would have been required for such a program including bespoke high-rate timers and custom radio message types as well as all the modules for controlling the actuation and on board data processing. A much more efficient use of time was to choose an existing acoustic ranging algorithm to use as a demonstration application.

There were a number of potential algorithms available which had been developed for the Mica2 platform. ‘Thunder’, developed at University of Virginia uses a central node with GPS and a loudspeaker to propagate position information to the rest of the network [32]. The solution proposed by Vanderbilt University [27] appeared to be best suited to the needs of this project. The quoted ranging error (of around 10 cm) was smaller than for the majority of other methods and there was also literature detailing that some other groups who had successfully implemented and investigated this algorithm. It was also being investigated on the Mica2 platform by collaborators at Coventry University [2], giving the opportunity to compare our two platforms. There were some modifications required to port the application to the MICAz platform, however. The main issue was an incompatibility with a specialist timer interface used on the Mica2 version of the application. The registry used for this timer is used for some of the radio functionality on the MICAz. The reason that this timer was developed was that the default operating system only incorporates one high rate timer component, which is used by the radio stack. The other timers which are available by default only have rates up to approximately 3 kHz. This is not sufficient for accurately synchronising the two nodes and would fundamentally limit the ranging performance to, at very best ± 0.5 m (assuming the speed of sound is 340 ms^{-1} and sampling uncertainty of half the sample window: ± 1.5 ms).

A significant amount of time was spent attempting to develop an equivalent high sample rate clock for the MICAz. Although ultimately unsuccessful, this work gave the opportunity to investigate the hardware abstractions of the timers in more detail. Instead of using a new timer, it was realised that the ranging algorithm had the property that the radio was always suspended during sampling and actuation. This can be seen from the inclusion of the `RadioSuspendC` component in figures 13 and 14. This was done to maintain a steady bias voltage for the microphone sensor ADC, as well as preventing any hardware interrupts from the radio during acoustic sensing. As observed in figure 5, this can significantly affect any sensor readings. The benefit of this exclusivity of sampling vs. radio activity within the algorithm meant that in fact the program could operate on one high rate timer alone. Thus with some adjustment to the Mica2 based code, it was possible to remove the timer interface which caused the registry clash. Following this adjustment, the algorithm was successfully ported onto the MICAz platform. It should be noted that this adaptation may have had some effect on the operation of the algorithm and hence the results from the system. These possible effects are discussed further in section 4.4.

The basic operation of the application shown in figure 12 can be described as follows (more detailed description can be found in the Vanderbilt *readme* file in appendix 8.5).

- The sounding node sends out a triggering radio message.
- 0.5 seconds later the sounding nodes emits 16 buzzes separated by variable silence lengths (as shown in figure 11).

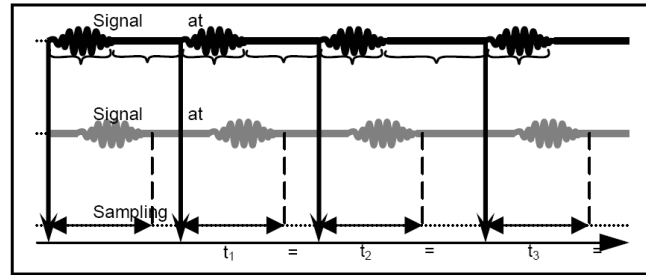


Figure 11: Schematic of the timing intervals for the acoustic sounding and listening. The length of the audio signals and delays between successive chirps are known by the listener, allowing the acoustic time of flight to be estimated [27].

- The listening node samples the microphone over intervals equal to the sounding windows and superposes each of these buzzes on top of each other.
- The buffered data is then analysed to find the start of the combined acoustic pulses, and the delay between this and the radio trigger is found.
- A message is broadcast by the listening node containing the identity of the sounder and the estimate of the range from it.

Every node can act both as a sounder and as a listener. When a node is first powered on, it sends out a ranging trigger and acoustic buzz, as described above. It will then wait approximately 40 seconds and buzz again. While it is waiting it is able to listen for other nodes sounding. By turning the nodes on at 5 second time intervals, it was possible to ensure that no two nodes tried to sound at the same time. There was some more sophisticated software available which included some synchronisation and networking, but this would have required a different approach to correcting the timer clash. It was also preferable to strip out as much of the functionality as possible so that the data could be analysed at the earliest possible stage. A common issue within wireless sensor networks is the difficulty of getting at the raw data from a system. Usually any sensor data will have gone through a number of stages before reaching the user (such as on board processing and radio networking as listed in section 1.3).

One of the most challenging problems for the on-board software to handle was judging the time of the start of the acoustic pulse. This proved to be the predominant source of uncertainty within the on-board ranging algorithm. The task of judging the start of a pulse when there is a noisy signal is very challenging, even in non resource-constrained situations. An example of the raw microphone data is shown in figure 15. Note this figure is simply an indication and has some features such as clipping, which would not be present in the buffered data during the ranging process.

As the sample rate is limited to 14 kHz, even one data point either side of the true start of the pulse results in a 5 cm variation in the range estimate (assuming the speed of sound is 340 ms^{-1}), so it is vital to find a way of consistently judging the start of the received acoustic pulse. As part of the software to assess the start of the acoustic pulse, there is some error-checking built in: if the maximum of the pulse is less than twice the background level, then the system returns an error flag. An error flag is also returned if the measured

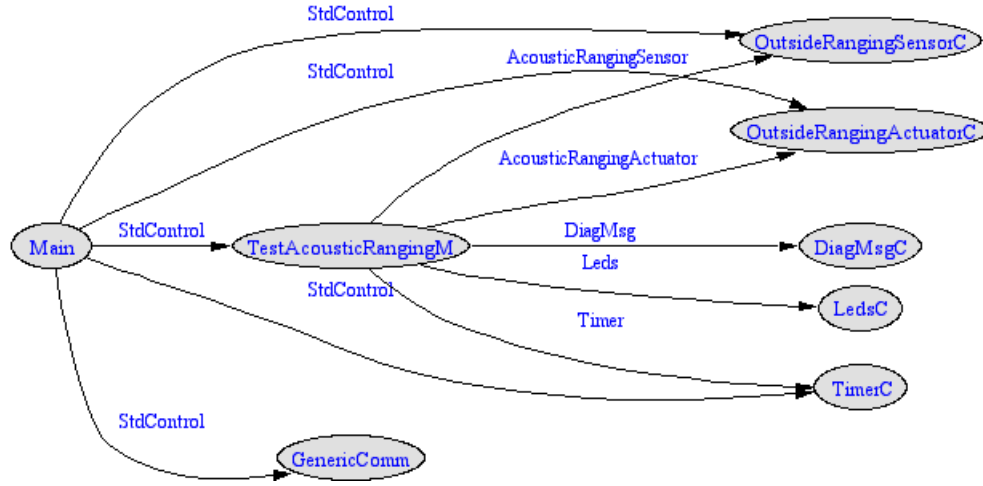


Figure 12: Components of the adapted Vanderbilt ranging algorithm

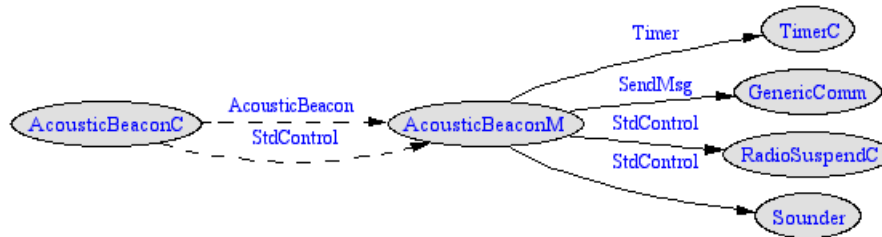


Figure 13: Sounding components in detail (Within OutsideRangingActuatorC in fig 12)

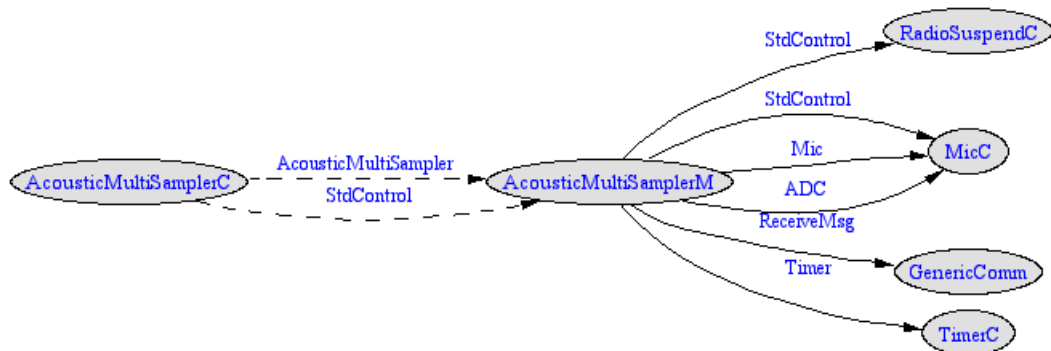


Figure 14: Listening components in detail (Within OutsideRangingSensorC in fig 12)

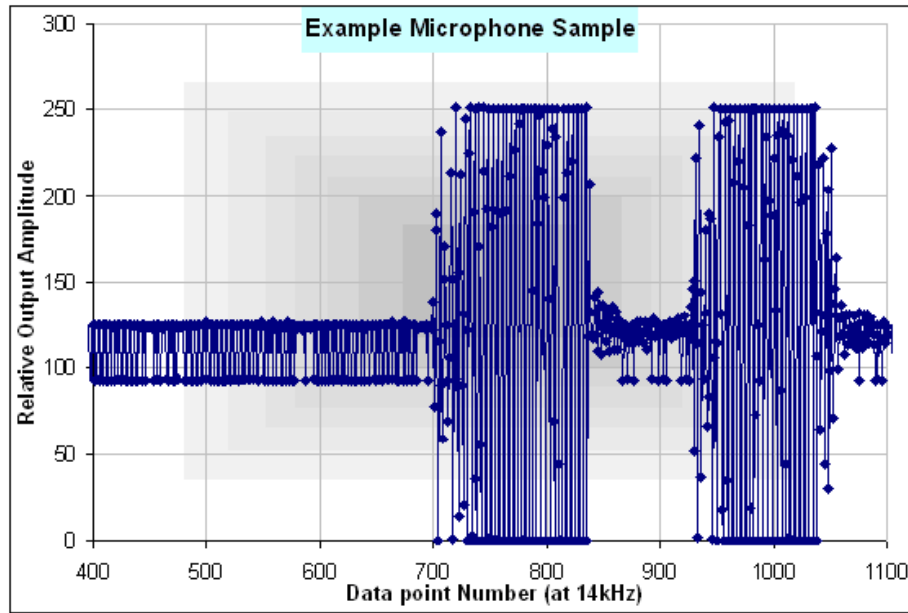


Figure 15: Example acoustic data sampled by the MICAz microphone

width of the pulse is significantly greater than the broadcast buzz. Such broadening of the pulse could be caused by poor synchronisation leading to the superposed signals being out of sync, or another cause could be an acoustically reflective environment. If the acoustic pulses were reflected from surfaces, then many echoes could arrive at the microphone causing the measured pulse to be smeared. As all these effects would lead to an erroneous (and meaningless) ranging estimate, the software is designed to flag them as errors. Full analysis and description of the algorithm are contained in [27] and is not discussed further in detail here.

As described in section 8.5 Vanderbilt University also provided PC-based software for receiving and interpreting the ranging packets. This was useful at early stages in the work and allowed the investigations to progress more rapidly. One drawback of this software was that it masked the number of error flags coming from the system. In addition the data could not be displayed or output in the most useful way. For these reasons the data extraction and analysis software described in section 3.1 was adapted to take data from the packets broadcast by the ranging nodes. The details of this ranging data analyser can be found in section 8.6. The different stages which the data went through are described below.

```
7e42ffffa37d5d06060000000000003f0 [16]
7e42ffffb17d5d105fb772616e67696e6706004505001700ccd5 [26]
7e42ffffb17d5d105fb772616e67696e6706004507001700a438 [26]
7e42ffffb17d5d105fb772616e67696e670600450200ffff0a03 [26]
7e42ffffb17d5d105fb772616e67696e670600450300ffffbe75 [26]
7e42ffffb17d5d105fb772616e67696e6706004504001a0024d5 [26]
7e42ffffa37d5d0605000000000000e33e [16]
7e42ffffb17d5d105fb772616e67696e6705004502004300189e [26]
7e42ffffb17d5d105fb772616e67696e670500450700ffffcd67 [26]
```

Node Pair	1from1	1from2	1from3	1from4	1from5	1from6	1from7
Separation							
Mean Estimate	0	14.5	18.33333	15.4	0	15.28571	0
s.d.	0	2.783882	0.942809	2.939388	0	3.452299	0
Data Points	0	8	6	5	0	7	0
Errors	0	0	0	0	8	0	7
		17	17	13	-1	17	-1
		17	17	19	-1	17	-1
		13	19	19	-1	19	-1
		17	19	13	-1	19	-1
		9	19	13	-1	13	-1
		13	19		-1	13	-1
		17			-1	9	-1
		13			-1		

Figure 16: Extracted data for nodes listening to node 1

```

7e42ffffa37d5d060100000000004238 [16]
7e42ffffb17d5d105fb772616e67696e6701004506001100f2fd [26]
7e42ffffb17d5d105fb772616e67696e67010045020011000337 [26]
7e42ffffb17d5d105fb772616e67696e670100450500ffff634b [26]
7e42ffffb17d5d105fb772616e67696e670100450700ffff0ba6 [26]
7e42ffffb17d5d105fb772616e67696e6701004503001100b741 [26]

```

This is an example of some of the raw ranging packets as received by XListen (similar to the method described in section 3.1). The shorter packets are the triggers sent out by the nodes as they are about to sound. The longer packets are the ranging estimates sent by the other listening nodes following the trigger. It was seen that for a single sounding event, estimates were not always received from all the other nodes. This is indicated above by the varying number of ranging estimate packets after each trigger. It is likely that the receiver was being swamped by all the nodes trying to broadcast their ranging estimates simultaneously. Some time was spent adding a varying delay to the broadcasts from the nodes in order to remove this clashing problem. However it was found that this led to synchronisation problems, so it was removed. Instead, data were gathered over extended intervals for each ranging position. This meant that, although some of the estimates were being lost due to radio crashing, sufficient estimates could be gathered for all the relevant node pairs. Once the raw packets had been recorded to a file, the data were extracted to Excel and sorted by node pairing as shown below.

As can be seen from figure 16, the analysing software also calculated the mean and standard deviation of the readings for a given run. The errors signified by -1 ranging estimates were those generated by the on-board data processing to indicate low signal to noise ratios or an excessively broad peak in the microphone data. The variable number of estimates resulting from message crashing can also be seen here. Similar tables were generated for all the node pairs. It should also be noted that the ranging estimates appear to be quantised. This is due to a combination of the microphone sampling resolution and the on board algorithm which analyses the microphone data. This is discussed in some more detail in section 4.4.

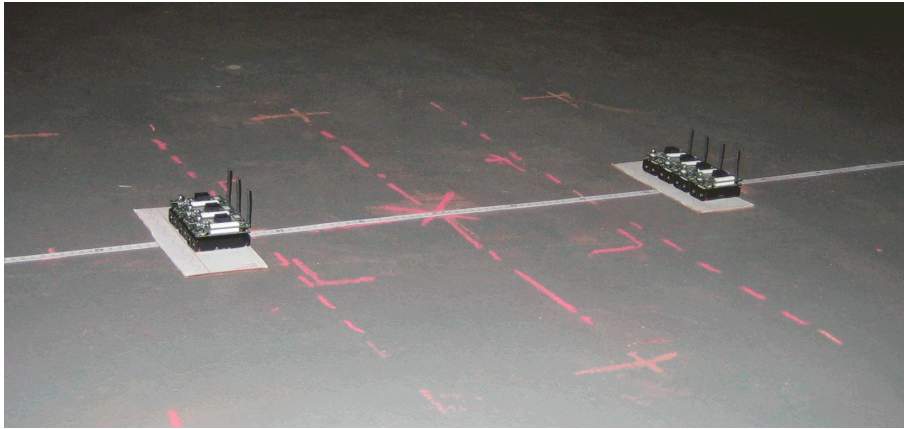


Figure 17: Layout for Ranging Calibrations in the Hemi-Anechoic Chamber

4.3 Calibration Method

Once the system and process of ranging was finalised, it would have been possible to analyse each source of uncertainty in detail and draw up an uncertainty budget. This would have had some value if this specific system were to be used for a safety critical or traceable measurement. However, in order to understand the quality of the system's results and provide general practical information on resource-constrained measurements, a 'black-box' calibration method was sufficient.

The basic aim of this calibration was to compare the ranging estimates from the node pairs to a reference distance (in this case a tape-measure). A simple process was set up where node pairs were placed at distances between 55 cm and 100 cm (as shown in figure 17) and range estimates made at 5 cm increments.

At each of the positions, ranging estimates were made for 5-7 minutes, giving around 9 ranging attempts per node pair. The estimates were then extracted as shown in figure 16. The arithmetic mean of the estimates for each given node pair at each given separation were then plotted against the nominally 'true' separation as indicated by the tape-measure. The standard deviation in each of these means was also recorded for later analysis. Examples of these compiled averages with their standard deviations are shown in figure 18.

The geometry of the layout meant that the actual node separations were subject to a systematic offset from the nominal separation, depending on node positions in the layout. The maximum size of this discrepancy due to geometry occurs at the closest distance, meaning that for some of the pairs in this configuration the actual separation would be 55.5 cm instead of 55 cm. From the fact that the diameter of the microphone is 5 mm and the ranging estimates returned by the nodes are in whole cm, it was decided that this geometric effect was negligible. The layout also meant that a given node only ranged from up to four others in any given calibration run. Efforts were made to rotate the layout to ensure that all the permutations were investigated.

The finite size of the nodes makes the issue of defining the position of the nodes challenging. The actual ranging estimates are made between the buzzer on the sounding node, and the

Ranges							
Node Pair	1from1	1from2	1from3	1from4	1from5	1from6	1from7
55		10.5	11.5	12.6		14.33333	
60		15.4	11.85714	13		12.5	
65		19	22.77778	21.2		21.44444	
70		28.33333	26.75	26.22222		24.71429	
75		27	32.66667	28.4		28.57143	
80		36.2	36	37.55556		34.75	
85		38.4	36	41.33333		38.5	
90		45.33333	44.625	48.2		44.57143	
95		45.71429	49.75	47.42857		46	
100		57.28571	55.28571	55.85714		56.14286	
S.d.							
Node Pair	1from1	1from2	1from3	1from4	1from5	1from6	1from7
55	0	3.840573	3.122499	0.8	0	0.942809	0
60	0	1.959592	2.799417	0	0	4.213075	0
65	0	0	1.930905	3.655133	0	2.753225	0
70	0	1.795055	1.984313	0.628539	0	3.194383	0
75	0	4.873397	1.490712	3.2	0	5.802885	0
80	0	2.6	0	2.454525	0	3.596874	0
85	0	3.2	3.399346	5.249339	0	5.17204	0
90	0	4.268749	4.846068	0.979796	0	2.25877	0
95	0	3.194383	2.817357	3.886134	0	4.335897	0
100	0	1.665986	0.699854	0.989743	0	3.522522	0

Figure 18: Example of Compiled Averages and Standard Deviations for Ranging from Sounder 1 for a Single Calibration Run

microphone on the listening node. As seen in figure 2 these are separated by approximately 2.5 cm, and the apparent measured separation of the two becomes dependent upon the orientation of the nodes. This issue of defining the position of a node of finite size affects both the ranging data in section 4.4 and the measurements made for the data fusion investigations in section 4.5. Depending on the application and required accuracy, this offset may need to be corrected for. This sort of effect could cause a problem for real-world networks which are randomly deployed, as the orientation information as well as the spatial coordinates would be required to describe fully a node's position.

Preliminary readings showed that the different nodes (there were seven in total) behaved differently. In this set-up each node could act as a listener and as a sounder. Based upon the early findings, it was found to be necessary to treat each of the forty-two node pair systems independently, accounting for the directionality of the ranging estimates (i.e. the difference between node A listening to node B compared to node B listening to node A).

It was also found early on that the environment had a strong effect on the measurements made. A moderate level of ambient background noise (for example from air conditioning) was seen to increase the number of error flags returned by the nodes (as described in section 4.2). The presence of acoustically reflecting surfaces was also seen to have a detrimental effect on the quality of the ranging estimates. Echoes could either result in an excessively large ranging estimate (due to the reflected signal taking a longer path) or the acoustic samples could be smeared out (due to many echoes arriving at slightly different times). To control these factors, the ranging calibrations were carried out in NPL's acoustically hemi-anechoic chamber⁵. The walls of this chamber are covered in acoustically absorbent material, and the whole room is isolated from external sound and vibration sources. The room also has good environmental control, which helps remove the issue of the temperature and humidity dependence of the speed of sound (approximated by $V_{sound} = 331 + 0.6T$ where V_{sound} is in ms^{-1} and T is in $^{\circ}\text{C}$). The ranging algorithm assumes a constant speed of sound of 340 ms^{-1} , whereas in reality, it varies by around 0.6 ms^{-1} per degree of temperature change, which could cause varying offsets over time in ranging estimates made in real environments.

As with the majority of ranging techniques, the results from this process are subject to a number of significant sources of uncertainty. This is consistent with much of the data which will come from the low-cost, mass produced WSN systems which are likely to be deployed in the future. On board self-evaluation techniques could greatly benefit the performance of this system, and the development of generalised techniques will be of value to a broad range of WSN applications. The early stages of work on self-evaluation are in section 4.6.1.

4.4 Calibration Results

Following the basic method described in section 4.3 yielded some interesting findings.

Figure 19 uses the convention XfromY where X is the sounder and Y the listener. The ranging estimates have a significant offset from the nominal true separation (shown by the dotted line). There are a number of potential reasons for these offsets. When porting the algorithm onto the MICAz platform, the scaling factor to convert to a range in *cm* was not

⁵The predominant use of this chamber is for measuring acoustic emissions from machinery and appliances. It is also used as an environment for calibration of instruments which are taken out into the field for measuring emissions for example around airports.

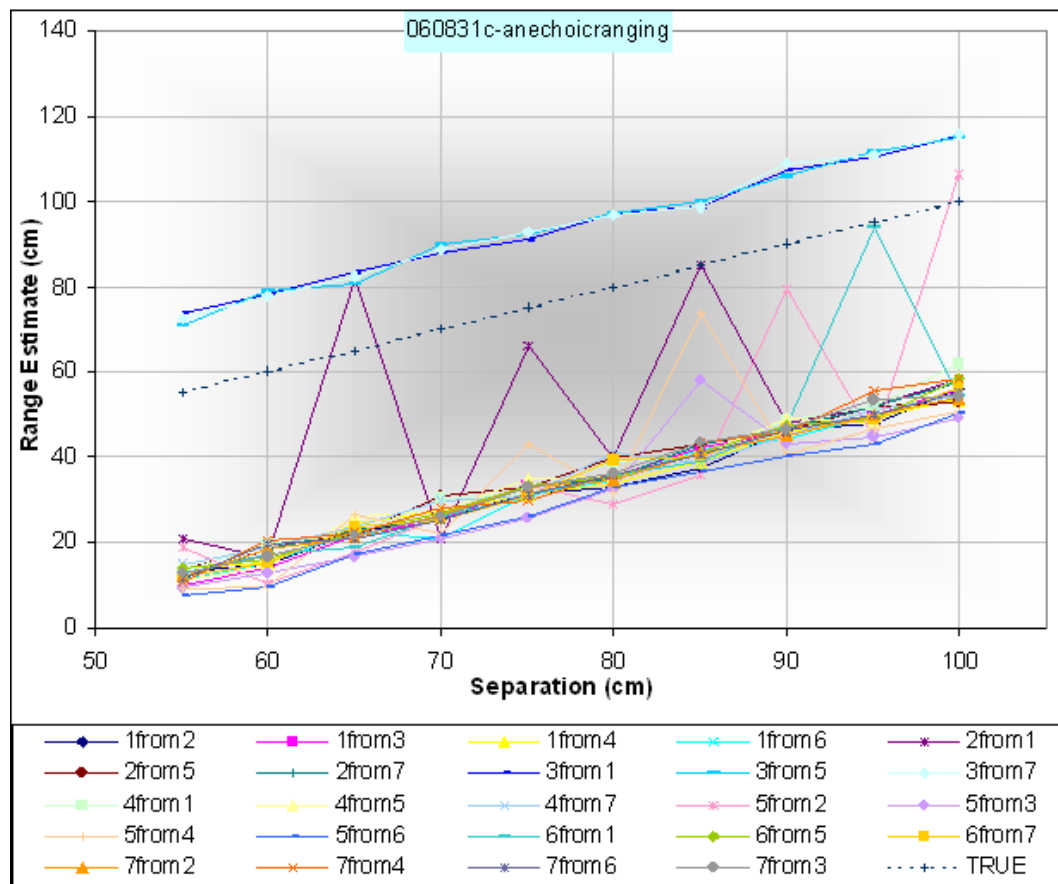


Figure 19: A set of ranging estimates from 55 cm to 100 cm

5from1	5from2	5from3	5from4	5from5	5from6	5from7
25.14286	35.75	58	74	0	36.57143	15.8
1.807016	3.929058	26.43861	24.08319	0	3.658928	3.709447
7	8	8	6	0	7	5
0	0	0	0	0	0	2
24	40	40	92		34	23
24	36	92	88		40	15
24	40	34	92		40	13
24	30	34	40		38	15
28	40	40	92		30	-1
28	30	40	40		34	13
24	34	92			40	-1
	36	92				

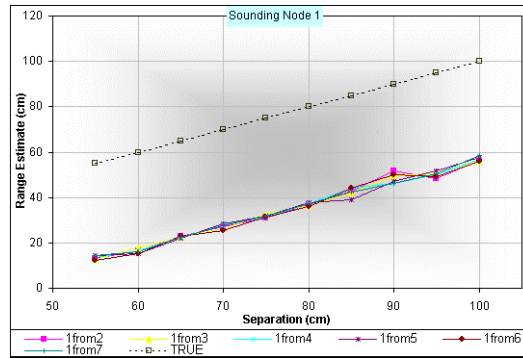
Figure 20: Intermittently large estimates (red) affecting average range estimate. Note that the frequency of these varies for the different sounders, and some such as node 3 remain consistently high.

adjusted. It is possible that the timer rate was not consistent across the platforms, which would introduce a scaling factor. If the wake-up latency of the buzzer (the time taken for it to start buzzing after a command from the CPU) was different for the MICAzs, then this would introduce an offset. The assumption of a speed of sound of 340 ms^{-1} would introduce a scaling factor if the actual speed were different. A combination of some, or all of these factors could have led to the observed offsets.

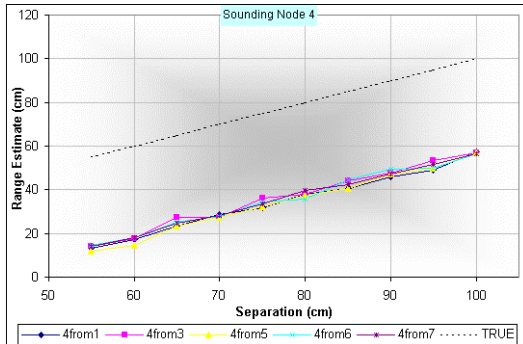
It can also be seen that all the estimates made when node 3 was sounding are significantly different from the majority of estimates (around 65 cm discrepancy). Some of the other pairings also make intermittently high estimates relative to the rest of the nodes. By looking back at the raw data in figure 20 that go to make up the plotted means, it can be seen that a large proportion of the jumps in range estimate (for example between nodes 5 and 4 at 85 cm) result from intermittent estimates which are significantly larger than the rest. The cause for this is likely to be a combination of hardware effects and software instability⁶. It was found that the sounders lacked robustness. Not only did they vary between nodes, but they could become adversely affected by any physical impact. It is therefore likely that the audio output was not particularly robust either. If the sounder frequency, output power or wake-up latency were to vary, then the resulting range estimate would also change. Some efforts were made to combat this using data analysis methods, as described in section 4.6.

Using these calibration processes for all the permutations, it was found that the largest variations in ranging estimate were caused by the sounders. For a given node sounding, the listeners generally gave consistent estimates, but these were not the same if another node was sounding. Based on this three ‘good’ sounders were identified. These were nodes which were found to produce the most consistent linear results across all the tests. The combined results from each of the good sounders are shown in figures 21(a) to 21(c). It can be seen that there is relatively good linearity of the ranging estimates (compared to figure 19). However, there are still differences between the estimates depending upon which node is listening, and

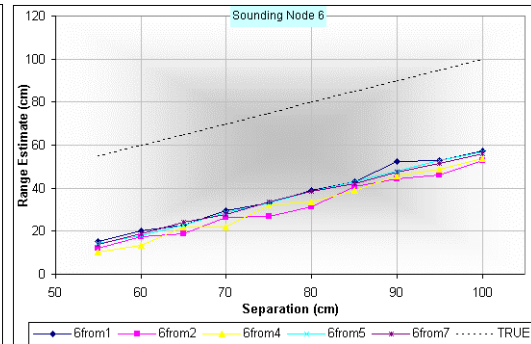
⁶One other potential cause was that the data were being incorrectly extracted from the packets. Preliminary data streaming investigations showed that extracting the wrong part of the packet string could result in aliasing giving apparently correct, yet offset results. This was ruled out in this case, however as the same offsets were observed using either the custom VB.net program or the Vanderbilt visualisation software described in section 4.2.



(a) Sounder 1



(b) Sounder 4



(c) Sounder 6

Figure 21: Results from 'Good' Sounders as Indicated by Linearity of Ranging Estimates

figure 21(c) has slightly worse linearity.

Now that systems with sufficiently good linearity have been identified, it is possible to apply straight-line fits to their data. The equations of these lines can then be used to calculate correction factors which can be applied to subsequent measurements data. The parameters of these fits are shown in figure 23.

It can be seen from figure 22 that the application of the correction factors has not only removed the offset in the readings, but also improved the gradient of the data. The error bars show two standard deviations. These are calculated from the individual readings such as those in figure 20. It can be seen that there is a large statistical uncertainty around the reading for 90cm separation. The data at this point is also slightly above the linear trend shown by the surrounding data. This is caused by a single outlier in the data. These occur intermittently and relatively rarely, although some nodes are more prone to them than others (see sounders 2 and 5 in figure 19). A combination of hardware discrepancies (such as changes in the sounder volume or frequency, or microphone response) combined with instabilities in the software algorithm are the likely causes. This kind of effect is likely to occur for many cheap systems where processing is limited.

Despite these issues of failure modes leading to discrepant data, sensor inhomogeneity and resource limitations, there are still some encouraging findings from this part of the project.

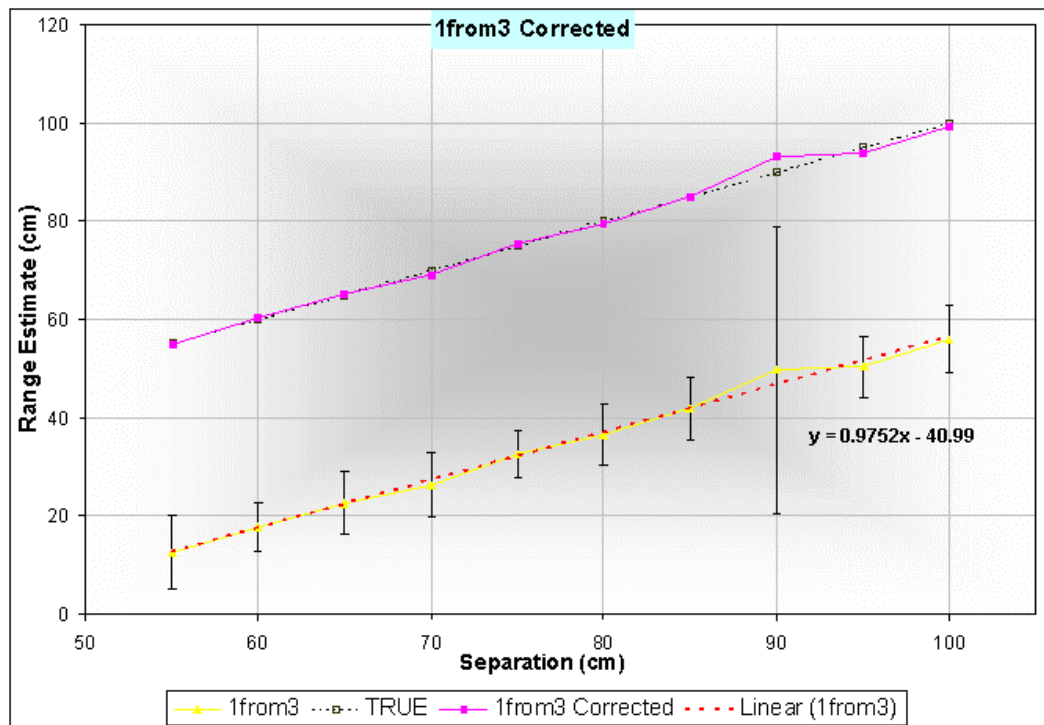


Figure 22: Corrected Ranging Estimates Between Nodes 1 and 3

Curve Fits Sounder 1			Curve Fits Sounder 4		
Listener	Gradient	Intercept	Listener	Gradient	Intercept
1			1	0.935	-37.969
2	0.983	-41.444	2	0.988	-42.070
3	0.975	-40.990	3	0.954	-37.651
4	0.962	-39.933	4		
5	0.979	-41.317	5	0.988	-42.322
6	0.999	-43.071	6	0.933	-36.768
7	0.980	-41.900	7	0.938	-37.156
Mean	0.980	-41.443	Mean	0.956	-38.989
S.D.	0.012	1.035	S.D.	0.026	2.519

Curve Fits Sounder 6		
Listener	Gradient	Intercept
1	0.972	-38.817
2	0.900	-37.981
3		
4	0.983	-44.089
5	0.972	-39.756
6		
7	0.939	-37.390
Mean	0.953	-39.607
S.D.	0.034	2.660

Figure 23: The results of the line fits to all the data for the 'good' sounders. This shows the good consistency between the nodes as listeners, as well as the slight differences in offset and gradient from the various sounders.

These highly constrained systems are capable of performing a complicated task involving a large range of functionality. The linearity and repeatability of the ranging estimates was sufficient for the offsets to be measured and corrected. By doing this the performance of the system was significantly improved. Further improvements could be made by performing outlier rejection on the data as it is gathered, and removing the data points that appear to be the result of some sort of system failure. Some preliminary work on this topic is discussed in section 4.6.

4.5 Data Filtering and Validation

Self validation is defined as the ability for a sensor to assess the level of uncertainty associated with a given measurement [14], or to flag certain data as resulting from some form of sensor failure. The approach can be sensor-specific; knowledge of the failure modes of a given sensor is exploited in order to assess if, and what type, of failure has occurred [15]. This can involve the inclusion of additional ‘complementary’ sensors to monitor the operating environment of the data gathering sensor (for example a thermometer to check that the node is within operating limits). A more generalised approach can be based upon looking at the data and using some physical knowledge about the measurand. Building up a historical picture of the data from a given sensor can allow intermittent discrepancies to be identified and possibly discarded or corrected. Comparing data from neighbouring sensors can also be used to assess the quality of an individual sensor and its readings, provided there is sufficient spatial resolution relative to the rate of change of the measurand.

In the context of acoustic ranging measurements, there are a number of failure modes which could manifest themselves in erroneous data, where self-validation could prove beneficial. In section 4.2 the following potential failures and physical constraints were identified:

- At the point of the raw data
 - If an individual microphone reading falls outside of a given range
- During On-board processing
 - If the repeat overlaid samples are not consistent with each other
 - If the shape of the data means the start of the pulse cannot be judged (e.g. smearing owing to echoes of the acoustic signal)
 - If the range estimate produced is not physically possible (e.g. negative)
 - If the range estimates change rapidly over time
- During aggregation at the base station
 - If the range estimates change rapidly over time
 - If the range estimates for nearby sensors differ widely (subject to the physical layout)

As described previously, there is already some on-board error-checking built into the ranging algorithm. An excessively poor signal to noise ratio indicates that the start point of the acoustic pulse cannot be found very easily or repeatably, so the algorithm looks out for this

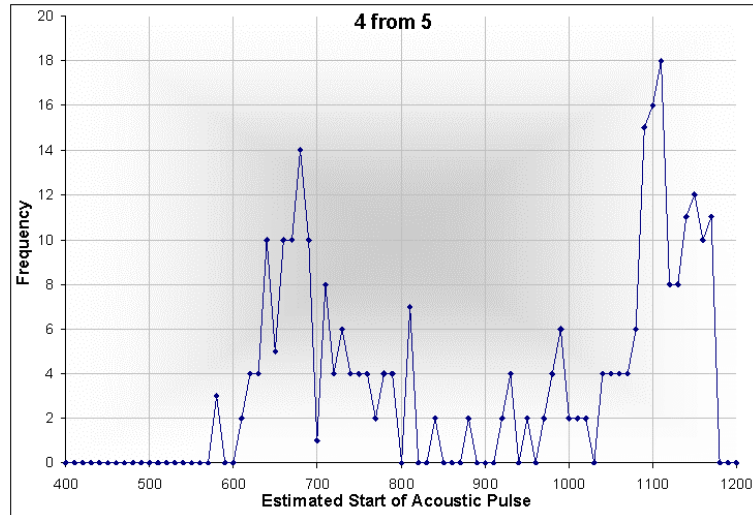


Figure 24: Estimates of the start data point of the acoustic pulse within the data buffer, with no error checking or conversion to a range. Note all these estimates are for a single position configuration, making the large spread of values an indication of errors.

and rejects estimates made under such conditions. A second on-board error-checking process looks for smearing of the pulse owing to echoes or poor synchronisation.

At the early stage of porting the algorithm the results with no error checking were investigated in an office environment. Some results can be seen in figure 24. The graph shows the position within the data buffer where the algorithm estimates the start of the received acoustic pulse. This is directly proportional to the subsequent ranging estimate which would be returned. For the purposes of this graph the main point of interest is the repeatability and stability of this estimation (hence the lack of scaling to a range). It shows that if the flagged values are not rejected, then there is a very large spread in the estimated time of the start of the acoustic pulse. Two peaks can be observed. The large number of readings around 1100 are likely to be caused by multipath acoustic reflections increasing the measured time of flight. Such data points would usually be rejected by the error checking owing to the fact that such pulses would also be broadened and more noisy, so would fail the onboard tests.

When carrying out investigations in the acoustically anechoic chamber, it was found that the number of error flags returned was relatively small. This indicates that in ideal conditions, the system may be able to operate without any checking required (at least for the two failures discussed here). The main benefit of the error checking here, is therefore to extend the range of potential operating situations in which the system works. This has shown that onboard data validation can not only improve system outputs, but for this application it has been vital to maintain functional stability in the range estimates generated.

The more general approach of performing statistical analysis on multiple sensor readings can also be used to identify and reject erroneous data. This is discussed in the following section.

4.6 Outlier Rejection

To be able to look at genuine data and identify outliers which are caused by some kind of error or failure mode, some sort of model needs to be applied. This will contain a number of inherent assumptions about the system, such as:

- The measurand is static or slowly changing.
- it is possible to make a statistical estimate of the value measurand (e.g. mean, median or mode).
- Consistency of a data point with the historical data implies its correctness.
- Outliers are rare and generally adjacent to good readings.

These methods rely on repeat measurements of nominally the same quantity. These can either be produced by multiple sensors, as long as it is ensured they are definitely measuring the same thing, or can be obtained from a single sensor taking repeated measurements of the same measurand.

The constraints imposed on an outlier rejection method which would function in resource-constrained wireless sensor network situation are as follows:

- It must be possible to operate the technique in, or near to, real time.
- Simple numerical methods must be used due to the processing constraints (complicated data fitting would be beyond the capabilities of the system).
- Only a small number of data values shall be held in memory at any one time (owing to resource limitations).

4.6.1 Outlier Rejection Process

At the time of writing, this work was at an early stage and the work below constitutes preliminary investigations into the topic.

In order to test a number of potential data filtering methods, a simple process of running a pseudo real-time filter was set up. This used existing archives of ranging data, but all the constraints and limitations listed above were adhered to. The data set initially chosen for these filtering simulations comes from some preliminary investigations into the effects of fully draining the batteries of the nodes⁷. Leaving the system running for a full day allowed this to be investigated, and also generated very large data sets (of around seven hundred repeat readings for a given node pair separation). Using the largest available set of data for these simulations had a number of advantages. Any slow changing spurious trends introduced would become evident. Also, the mean of such a large data set should be a good estimate of the measurand, giving a reference point to check any filtered data set against.

⁷It was found that there was little variation over time when the batteries were steadily drained from a moderately full starting point. When brand new batteries were first used, it was found that the ranging estimates were slightly more erratic. This is probably due to this being the period of fastest change for the battery voltage.

The filtering criteria are described below and the following variables are used: x_i is the value of the current data point to be checked for consistency, $\bar{X}_{4K}$ is the mean of the four most recent data points, $S_{X_{4K}}$ is the standard deviation of this mean, R is the chosen rejection criterion and $\tilde{x}_i$ is the data value returned by the system. Using the equation below, a very coarse outlier rejection method was implemented.

$$\tilde{x}_i = \begin{cases} \bar{X}_{4(K-1)} & \text{if } 2 \left(\frac{S_{X_{4K}}}{\bar{X}_{4K}} \right) > R \\ x_i & \text{if } 2 \left(\frac{S_{X_{4K}}}{\bar{X}_{4K}} \right) \leq R \end{cases}$$

There is a trade off to be made between the number of data points in the rolling average and the resource constraints of the system. The memory available to hold all the data values is limited. The sample rate would also have to be considered, and very low sample rates may mean that it would be impractical to perform averaging of large sets of data points.

4.6.2 Outlier Rejection Findings

The results of applying this filtering to a large set of ranging data can be seen in figure 25. Here it can be seen that the original data has one estimate of 53 cm, which constitutes a significant outlier (marked by the blue arrow). Taking the average of all the raw data gives a range estimate of 84.3 cm, which is the best guess of the measurand from this data set. Using a rejection criterion of 0.2 removes this outlier, and begins to narrow the shape of the distribution. A criterion of 0.1 further narrows the distribution, but also introduces some spurious rejections. These can be seen by the gaps starting to appear around 75 cm and 90 cm. The reason for this is that the rejection is highly dependent upon the order and shape of the data. It acts as a low pass filter and rejects any large, rapid changes. This means that values close to the overall mean (and hence best estimate) may be rejected if they happen to lie adjacent to three similar, but outlying data points. This is evidence of the failure of the above assumption that outliers are generally adjacent to good readings.

There are a number of clear limitations to this method, which need to be accounted for and recognised:

- The original data is not replaced with a corrected value. This means a potential outlier would be included in four of the rolling averages ($\bar{X}_{4K}$). This could be easily corrected in the algorithm.
- The criterion for rejection needs to be chosen arbitrarily based upon analysis of large data sets (and the distribution of returned values checked against the nominal overall mean).
- A more sophisticated check for ‘outlier status’ may be required.
- The choice of a four point running average is relatively arbitrary - more would be better, but the constraints of the hardware resources and sampling time need to be considered.

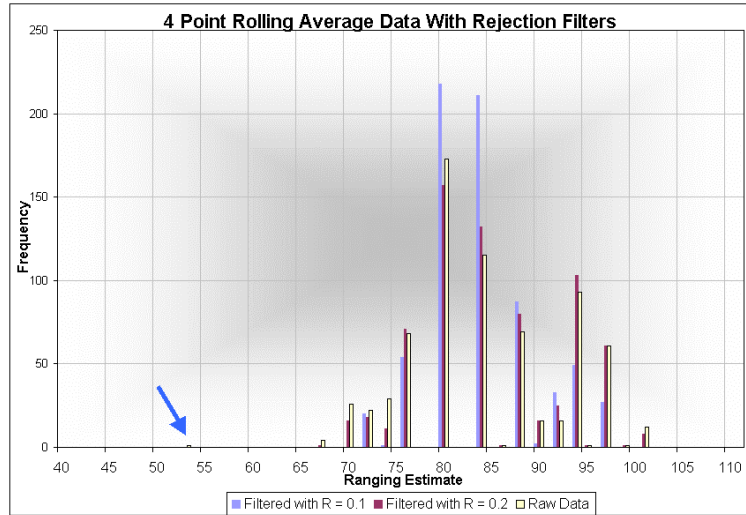


Figure 25: The effect of filtering data with various correction criteria

Despite these serious limitations, such a simple method of rejecting outliers would be perfectly acceptable for those data points which are significantly offset and sufficiently rare. It has been shown that this applies to some of the data which is generated by repeated ranging estimates. For more rigorous and robust onboard data analysis techniques, a more detailed understanding of the different failure modes, specific to the sensing task, would be required.

A more rigorous approach to analysing the output data was carried out within [10]. A measurement task was set up to use all the nodes to measure the radius of a circle (as shown in figure 26). This allowed their outputs to be directly compared with one another. The results found that by using a technique of defining the largest consistent sub-set (LCS) of the data (see [10]), it could be aggregated in such a way as to achieve an uncertainty of ± 0.27 cm, a vast improvement from the ± 3.5 cm uncertainty found in a single measurement.

4.7 Acoustic Ranging Conclusions

The main aim of this acoustic ranging case-study was to investigate the performance of a wireless sensor network system when performing a resource-hungry task. Such systems have limited capabilities in terms of data processing, sensor quality and communications capacity. A high rate of failures, which resulted in intermittent offsets in ranging estimates were observed. Significant constant offsets in the ranging estimates were also seen. These resulted from a combination of hardware frailty and software instabilities.

Despite all of these limitations, it was possible to calibrate the ranging of the sensors and correct for the fixed offsets. With statistical analysis and data filtering it was also possible to combat some of the intermittent errors and failures which were observed. These techniques both significantly improved the quality of outputs from this system.

As a method of node self-localisation, acoustic ranging suffers many significant and insur-

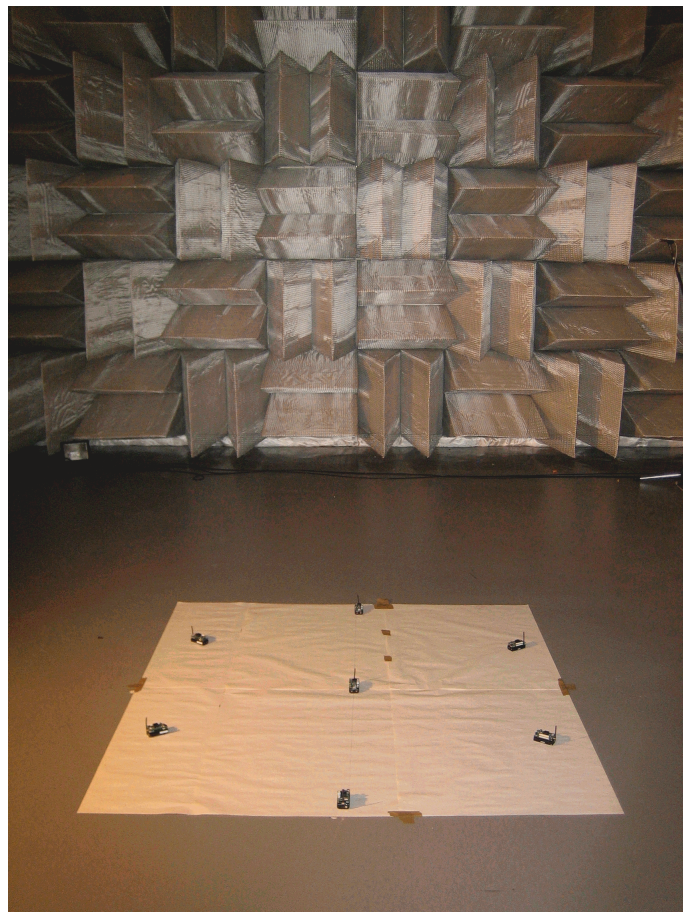


Figure 26: Layout for Gathering Largest Consistent Sub-Set Data in the Hemi-Anechoic Chamber

mountable limitations. Any acoustically noisy or reflective environment would make such systems virtually un-usable. Other potential problems include the potential for air movements and temperature gradients. Methods which show more promise involve co-operative use of interference patterns of radio signals [21]. These have the benefit of not requiring additional hardware, and are not subject to the same physical constraints as acoustic ranging.

5 Summary and Conclusions

The main aim of this project was to develop practical expertise and understanding in the area of wireless sensor networks. Based on an initial investigation into the current research in the area a number of key measurement issues were identified. These included radio link quality and data processing at various levels.

These issues were investigated using a testbed of MICAz motes. Radio link investigations found that reflective and multipath environments could be highly detrimental to the quality of radio communications within such networks. It was also found that external interference could cause a number of problems resulting in data loss as well as impaired functionality of the nodes. The type and magnitude of these effects varied depending on the interference type. Bluetooth signals had a different effect than Gaussian noise. This is an important finding as many environments contain networks operating using a variety of wireless protocols. Current electromagnetic compatibility (EMC) testing does not account for the variation in effects caused by different radio protocols. If systems are to be fully characterised to assess whether they can co-exist, then more work is required to understand the interactions between such digital systems.

The initial research into the area also showed that a large amount of the existing work being carried out was concentrating on specific components or functionalities of WSN systems. As these are highly integrated systems, with close interactions between the different components, it was important to gain an understanding of the systems as a whole. In order to achieve this, and look at the effect which could only be observed with a ‘whole system’ approach, a case study into acoustic ranging was carried out. This application utilised a significant proportion of the available functionality available to the system. It was found that a combination of hardware fragility and software instability led to a significant number of failures and erroneous readings. Despite these failings a combination of data validation and data filtering significantly improved the outputs of the individual nodes. When the network was used collectively to make a single measurement, the resulting output was approximately forty times more accurate than a single reading.

6 Future Work

Within the context of projects at NPL, there are a number of strands of work which are a continuation of some of the investigations carried out here. An Innovation Research and Development programme project (Signal Propagation and Data Transfer in Wireless Networks) is making use of the reception success rate measurement techniques developed as part of this project. The aim is to characterise more fully the interaction between different wireless communications systems operating under different protocols. As the type of digitisation has been shown to affect the nature and magnitude of inter-system interference, the aim is to augment the existing electromagnetic compatibility testing guidelines to account for signal digitisation.

The data aggregation and data fusion investigations were carried out jointly with the MET 5.1 project: *Effective Communication in Wireless Sensor Networks* and NPL’s Mathematics and Scientific Computing Group. The intention is to make the methods more generalisable, and also to port some of the filtering and data validation onto the nodes themselves. A dis-

tributed temperature measurement application using the MICAZs is providing some sample data for this work, and there is a collaborative comparison of methods with The Department of Engineering Science at Oxford University.

The Acoustics group at NPL are working on using a wireless platform to use for their novel MEMs microphones. They are receiving advice and input on the networking issues, and they are benefiting from the experience gained during the acoustic ranging investigations.

More generally, this project has identified many areas within wireless sensor networks which require further work. Self-localisation remains an important capability for many WSN applications. Methods involving acoustic time of flight have been shown to be insufficient in terms of accuracy and stability for the majority of real-world applications. Further work on using smart antennas for bearings information, RSSI or other radio-based localisation techniques are in need of further study.

The difficulty of programming in nesC caused some delay to the progress of the project, and this has been echoed by others working in the area. The development of a more intuitive, user-friendly programming environment for WSN systems would greatly speed up developments of novel applications as well as allowing more users from a non-software background to be able to concentrate on the sensing applications.

The differences in offset and behaviour of the sensors included on our MICAZ platform meant that any measurements were spurious without a thorough preceding calibration process. This would become impractical for large dispersed networks, so some methods of remote calibration need to be developed (some possible solutions to this are being looked at within the Innovation Research and Development programme). Possible solutions are novel sensor designs that incorporate their own artefact, self-validating sensors which can measure the same quantity in different ways (and hence register the occurrence of drift or analytical, model-base approaches). This is an area where NPL can make a significant contribution and enable WSN systems to reach their full potential.

7 Acknowledgments

NPL's Strategic Research funding supported much of this work. Collaborators from the Engineering Department at Oxford University provided input into the work relating to data aggregation and processing. The Mathematics and Scientific Computing Group at NPL provided data processing expertise. NPL's Acoustics group provided access to the Hemi-Anechoic chamber, which allowed repeatable acoustic ranging measurements to be made. The choice of Acoustic Ranging as a case-study application was helped in part by the Cogent Computing Group at Coventry University. They also provided invaluable tuition in nesC programming. Vanderbilt University provided the original application code for the Acoustic Ranging case-study, which was then ported to our system. Finally, the members of the TinyOS mailing list were a constant source of advice on a wide range of issues relating to programming in nesC, as well as making the most of our hardware platform.

References

- [1] Ajdler T, Kozintsev T, Lienhart R and Vetterli M, 'Acoustic Source Localization In Distributed Sensor Networks' Intel Labs, Intel Corporation, Santa Clara, USA, Signals, Systems and Computers, 2004. Conference Record of the Thirty-Eighth Asilomar Conference Volume 2, 7-10 Nov
- [2] Allen M, Gaura E, Newman R and Mount S, 'Experimental Localization with MICA2 Motes' *Technical Proceedings of the 2006 NSTI Nanotechnology Conference and Trade Show, Volume 3 Vol. 3, p 435-440*
- [3] Arnaut L, 'Operation of Electromagnetic Reverberation Chambers With Wave Diffractors at Relatively Low Frequencies' IEEE Transactions on Electromagnetic Compatibility, vol. 43, no. 4, November 2001
- [4] Bahl P and Padmanabhan V, 'RADAR: An In-Building RF-based User Location and Tracking System' Microsoft Research, Infocom 2000
- [5] Bokareva T, Hu W, Kanhere S, Ristic B, Gordon N, Bessell T, Rutten M and Jha S, 'Wireless Sensor Networks for Battlefield Surveillance' School of Computer Science and Engineering, The University of New South Wales, Land Warfare Conference 2006
- [6] Boltryk P J, Haris C J and White N M, 'Intelligent sensors - a generic software approach' *Journal of Physics: Conference Series* Vol 15 (2005) 155-160
- [7] Buonadonna P, Hill J, Culler D, 'Active Message Communication for Tiny Networked Sensors' Submitted to IEEE INFOCOM 2001
- [8] Clark j, Vinter B and Yaqoob M, 'The Shifted Rayleigh Filter: A New Algorithm for Bearings Only Tracking' Department of Electrical and Electronic Engineering, Imperial College London
- [9] Collett M, Esward T, Adamson D, Smith A, 'Wireless Sensor Network Demonstrator Testbed 109681 9SRPE210' Report to the Strategic Research Working Group, January 2007
- [10] Collett M A, Cox M, Esward T, Harris P, Sousa J, 'Aggregating Measurement Data Subject to Common Effects' *Metrologia*
- [11] Elnahrawy E, Li X and Martin R, 'The Limits of Localization Using Signal Strength: A Comparative Study' Department of Computer Science, Rutgers University
- [12] Farina A, 'Target tracking with bearings-only measurements' *Signal Processing, vol. 78, no. 1, pp. 6178, Oct. 1999.*
- [13] Forbes A, 'Multistation Coordinate Metrology' *Multilateration Presentation to SIDAnet November 2002*, Viewed online on 17th October 2006 <http://www.npl.co.uk/ssfm/download/>
- [14] Frolik J and Kandasamy P, 'A confidence-based approach to the self-validation, fusion and reconstruction of quasi-redundant sensor data' *IEEE Transactions on Instrumentation and Measurement* Vol 50, no. 6 (2001) 1761-1769

- [15] Gaura E and Newman R, 'Intelligent diagnostic for sensor arrays' IEEE Mediterranean Conference 2003 T7140, <http://med.ee.nd.edu/MED11/pdf/papers/t7-140.pdf> Viewed online on 17th October 2006
- [16] Gay D, Levis P, Von-Behren R, Welsh M, Brewer E, Culler D, 'The nesC Language: A Holistic Approach to Networked Embedded Systems' University of California at Berkeley, Intel Research, Berkeley, <http://nescs.sourceforge.net/>
- [17] Jeong J, Culler D, Jae-Hyuk Oh, 'Empirical Analysis of Transmission Power Control Algorithms for Wireless Sensor Networks' Electrical Engineering and Computer Sciences University of California at Berkeley, <http://www.eecs.berkeley.edu/Pubs/TechRpts/2005/EECS-2005-16.html>
- [18] Kushwaha M, Molnar K, Sallai J, Volgyesi P, Maroti M and Ledeczi A, 'Sensor Node Localization Using Mobile Acoustic Beacons' *2nd IEEE Conference on Mobile and Ad-hoc Sensor Systems November 2005*
- [19] Levis P, Madden S, Gay D, Polastre J, Szewczyk R, Woo A, Brewer A and Culler D, 'The Emergence of Networking Abstractions and Techniques in TinyOS' Networked Systems Design and Implementation 2004
- [20] Maroti M, Kusy B, Simon G and Ledeczi A, 'The Flooding Time Synchronization Protocol' The Institute for Software Integrated Systems, Vanderbilt University, <http://www.isis.vanderbilt.edu/publications/archive/Maroti.M.2.12.2004.The.Floodi.pdf>
- [21] Maroti M, Volgyesi P, Dora S, Kusy B, Nadas A, Ledeczi A, Balogh G and Molnar K, 'Radio Interferometric Geolocation' Institute for Software Integrated Systems Vanderbilt University, Technical Report: ISIS-05-602
- [22] Martinez K, Padhy P and Riddoch A, 'School of Electronics and Computer Science University of Southampton United Kingdom' Workshop on Real-World Wireless Sensor Networks 2005, *REALWSN'05*
- [23] Moore D, Leonard J, Rus D and Seller S, 'Robust Distributed Network Localization with Noisy Ranging measurements' *SenSys '04, November 3-5, 2004, Baltimore, Maryland, USA*
- [24] Niculescu D, Nath B, 'Forwarding and Positioning Problems in *Ad-hoc* Networks' Graduate School—New Brunswick Rutgers, The State University of New Jersey, May 2004
- [25] Ping S, 'Delay Measurement Time Synchronization for Wireless Sensor Networks' IRB-TR-03-013, June 2003
- [26] Roscoe A W, 'The Theory and Practice of Concurrency' Published 1997, revised to 2000 and lightly revised to 2005, The original version is in print in April 2005 with Prentice-Hall (Pearson).
- [27] Sallai J, Balogh G, Maroti M and Ledeczi A, 'Acoustic Ranging in Resource Constrained Sensor Networks' *Technical Report, ISIS, Vanderbilt University 2003*
- [28] Santi P and Simon J, 'Silence is golden with high probability: maintaining a connected backbone in wireless sensor networks' *Proceedings of 1st European Workshop EWSN Berlin January 19-21 (2004) 106–121*

- [29] Steere D, Baptista A, McNamee D, Pu C, and Walpole J, ‘Research Challenges in Environmental Observation and Forecasting Systems’ Department of Computer Science and Engineering Oregon Graduate Institute, The Sixth Annual International Conference on Mobile Computing and Networking, *MobiCom 2000*
- [30] Thorn J, ‘Deciphering TinyOS Serial Packets’ Director of Product Development, Octave Technologies Technical Briefing
- [31] Whitehouse C, Culler D ‘The Design of Calamari: an *Ad-hoc* Localization System for Sensor Networks’ Department of Electrical Engineering and Computer Sciences, University of California at Berkeley
- [32] Zhang J, Yan T, Stankovic J and Son S, ‘Thunder: Towards Practical, Zero Cost Acoustic Localization for Outdoor Wireless Sensor Networks’ Department of Computer Science University of Virginia, Charlottesville 22903, *Mobile Computing and Communications Review*
- [33] <http://www.arup.com/index.cfm> Viewed on December 2nd 2006
- [34] <http://www.applied-nanodetectors.com/> Viewed on December 2nd 2006
- [35] <http://nms.lcs.mit.edu/papers/assisted-localization.pdf> Viewed on December 2nd 2006
- [36] <http://www.tinyos.net/>, Viewed on December 2nd 2006
- [37] http://www.xbow.com/Products/Product_pdf_files/Wireless.pdf/Smart_Dust_AppNote.pdf Viewed on December 2nd 2006
- [38] <http://www.xbow.com/Products/productsdetails.aspx?sid=88> Viewed on December 2nd 2006
- [39] <http://www.xbow.com/Technology/MeshNetworking.aspx> Viewed on December 2nd 2006
- [40] <http://www.xbow.com/Products/> Viewed on December 2nd 2006
- [41] <http://www.stg.com/wireless/ZigBee.comp.html> Viewed on December 2nd 2006

8 Appendices

8.1 MICAz Data Sheet

MICAz

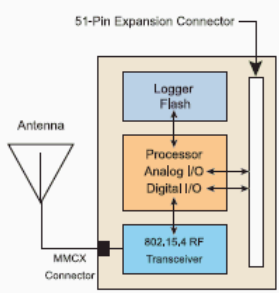
WIRELESS MEASUREMENT SYSTEM

- IEEE 802.15.4, Tiny, Wireless Measurement System
- Designed Specifically for Deeply Embedded Sensor Networks
- 250 kbps, High Data Rate Radio
- Wireless Communications with Every Node as Router Capability
- Expansion Connector for Light, Temperature, RH, Barometric Pressure, Acceleration/Seismic, Acoustic, Magnetic and other Crossbow Sensor Boards

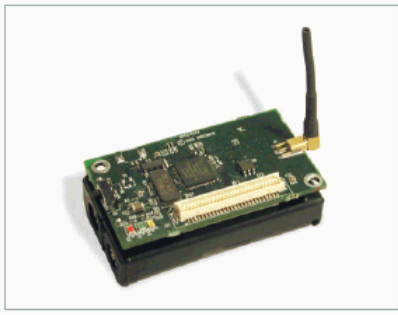
Applications

- Indoor Building Monitoring and Security
- Acoustic, Video, Vibration and Other High Speed Sensor Data
- Large Scale Sensor Networks (1000+ Points)
- ZigBee Compliant Systems and Sensors


ZigBee™ Alliance
 A member of the ZigBee Alliance



MPR2400CA Block Diagram



MICAz

The MICAz is a 2.4 GHz, IEEE 802.15.4 compliant, Mote module used for enabling low-power, wireless, sensor networks. The MICAz Mote features several new capabilities that enhance the overall functionality of Crossbow's MICA family of wireless sensor networking products. These features include:

- IEEE 802.15.4/ZigBee compliant RF transceiver
- 2.4 to 2.4835 GHz, a globally compatible ISM band
- Direct sequence spread spectrum radio which is resistant to RF interference and provides inherent data security
- 250 kbps data rate
- Runs TinyOS 1.1.7 and higher, including Crossbow's reliable mesh networking stack software modules
- Plug and play with all of Crossbow's sensor boards, data acquisition boards, gateways, and software

TinyOS is a small, open-source, energy-efficient, software operating system developed by UC Berkeley which supports large scale, self-configuring sensor networks. The source code software development tools are publicly available at: <http://webs.cs.berkeley.edu/tos>

Processor & Radio Platform (MPR2400CA)

Using TinyOS, a single processor board can be configured to run your sensor application/processing and the mesh networking radio stack simultaneously. The MICAz (MPR2400CA) IEEE 802.15.4 radio offers both high speed (250 kbps) and hardware security (AES-128). The MICAz 51-pin expansion connector supports Analog Inputs, Digital I/O, I2C, SPI and UART interfaces. These interfaces make it easy to connect to a wide variety of external peripherals.

Sensor Boards

Crossbow offers a variety of sensor and data acquisition boards for the MICAz Mote. All of these boards connect to the MICAz via the standard 51-pin expansion connector. Custom sensor and data acquisition boards are also available. Please contact Crossbow for additional information.

Base Stations

A base station allows the aggregation of sensor network data onto a PC or other computer platform. Any MICAz Mote can function as a base station by

8.2 Interpreting nesC Component Graphs

The details of nesC will not be covered here, but the component graphs provide a valuable way of viewing all the different parts included in an application. All of the stages between the hardware and the application layer can be exposed to get a better understanding of the operation of a system. It should be noted that the majority of component graphs within this document only show the upper layers of an application. The lower level components are usually more generic and perform non-specialist tasks such as controlling the radio or a timer.

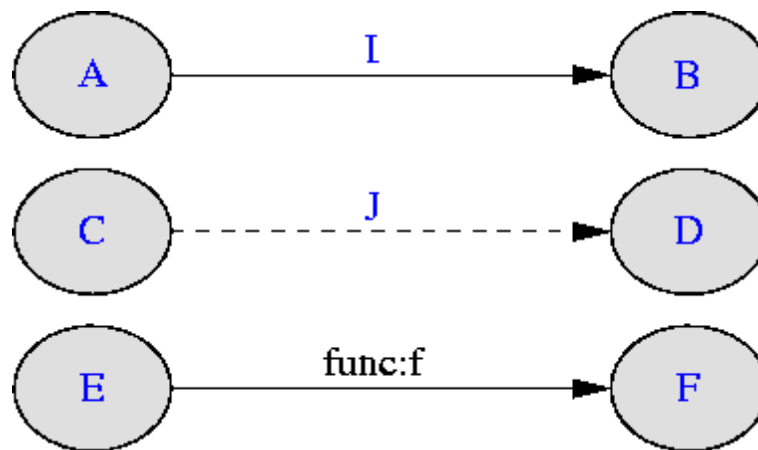


Figure 27: Descriptions of Different Connection Types

- A requires interface I, B provides I, and A and B are wired together.
- C and D both require or both provide J. The direction of the arrow indicates that the original wiring is 'C = D'.
- E requires function f, and F provides function f.

8.3 Readme for TOSBase application

Below is detailed the functionality and operations of TOSBase. This application acts as a simple forwarder which listens to messages and forwards them to the PC via the serial port. The PC can also send messages into the network via this application.

README for TOSBase

Author/Contact: tinyos-help@millennium.berkeley.edu

Description:

TOSBase is an application that acts as a simple bridge between the serial and radio links.

TOSBase will copy its compiled-in group ID to messages moving from the serial link to the radio, and will filter out incoming radio messages that do not contain that group ID.

TOSBase includes queues in both directions, with a guarantee that once a message enters a queue, it will eventually leave on the other interface. The queues allow the TOSBase to handle load spikes more gracefully.

TOSBase acknowledges a message arriving over the serial link only if that message was successfully enqueued for delivery to the radio link.

The LEDS are programmed to toggle as follows:

RED Toggle	- Message bridged from serial to radio
GREEN Toggle	- Message bridged from radio to serial
YELLOW/BLUE Toggle	- Dropped message due to queue overflow in either direction

Tools:

`tools/java/net/tinyos/sf/SerialForwarder.`

See `doc/serialcomm/index.html` for more information using these tools.

Known bugs/limitations:

As always, a task post failure somewhere in the serial or radio stacks may cause the TOSBase to hang. The chance of this is lessened by including the `CFLAGS += -DTOSH_MAX_TASKS_LOG2=8` line in the TOSBase Makefile, incleasing the size of the task queue to 256.

8.4 VB.net Code: Packet Loss Analyser

This section contains some of the key parts for the code which analyses the streamed data for lost or repeated packets.

The following code performs some reformatting of the raw data files. The process is as follows:

- Read the current line of the target file
- Check that a line starts with the sync byte (7e, signifying the start of a packet)
- Extract the source node ID and counter
- Convert the hexadecimal information from the packets into decimal
- Write the converted IDs, counters and original packets to a new 'formatted' file
- Skip to read the next non empty line of the raw data file

```

line = oldFile.ReadLine()

While (line <> Nothing)

    If (line.Substring(0, 2) = "7e") Then

        'Surge Message Types:
        'newline = Hex2Dec(line.Substring(24, 2)).ToString("D3") & "," &
        'line.Substring(24, 2) & "," & line.Substring(20, 4) & "," & line
        'Oscope Message types
        '(counter is, strictly speaking number of full buffers,
        'rather than sent messages)
        nodeID = Hex2Dec(line.Substring(16, 2)).ToString("D2")
        'CntToRfm Message types
        newline = Hex2Dec(line.Substring(16, 2)).ToString("D3") & "," & line
        newFile.WriteLine(newline)

    End If

    Label_DebugBox.Text = "Converting File: " & line
    line = oldFile.ReadLine

End While

oldFile.Close()
newFile.Close()

```

The following is a function which converts a Hexadecimal string to a decimal string. This is used in many of the analysis tools created for this project including the ranging data packet analysis and interference investigations.

```

Public Function Hex2Dec(ByVal Hexstring As String) As Integer
'Converts a Hex string to Decimal
'Known to work for 2 character strings
'- probably works for longer ones too but not tested
'(and not required here)
Dim LorRchar As String
Dim I As Integer
Dim TempVal As Integer
Hex2Dec = 0
Hexstring = Trim(Hexstring)
For I = 1 To Len(Hexstring)
LorRchar = Mid$(Hexstring, Len(Hexstring) - I + 1, 1)
Select Case LorRchar
Case "a"
    TempVal = 10
Case "b"
    TempVal = 11
Case "c"
    TempVal = 12
Case "d"
    TempVal = 13
Case "e"
    TempVal = 14
Case "f"
    TempVal = 15
Case Else
    TempVal = Val(LorRchar)
End Select
Hex2Dec = Hex2Dec + TempVal * 16 ^ (I - 1)
Next I
End Function

```

The following code reads a data file containing packets reformatted using the above method. The General process is as follows:

- Read the first two lines of the formatted data file (created by the above process)
- Calculate the counter difference between consecutive lines
- Adjust this number to account for the following known, non error-based effects:
 - The counter going from 255 back to 0
 - The counter number '7e' being replaced with '7d5e'
- If the difference between counters is not -1 (even after correcting for known effects), then write the offending packets to a file
- Increment counters to keep track of the number of lost or repeated packets
- Loop to the next two non-empty lines of the source file

- Write the total stats for lost packets etc. to the end of the new file

```

line = oldFile.ReadLine()
line2 = oldFile2.ReadLine()
line2 = oldFile2.ReadLine()

While (line2 <> Nothing)

'For Surge Message types
'diff = CInt(line.Substring(0, 3)) - CInt(line2.Substring(0, 3))
'If diff = 255 Then
'diff = -1
'End If
'For NumMicOScopeRF Message Types

diff = CInt(line.Substring(0, 3)) - CInt(line2.Substring(0, 3))

'*****
'These If statements deal with known factors which affect the count broadcast
'Following If statements account for the reserved 7e always being replaced
'with 7d
If ((line.Substring(0, 3) = 125) And (line2.Substring(0, 3) = 125)) Then
diff = -1
End If

If ((line.Substring(0, 3) = 125) And (line2.Substring(0, 3) = 127)) Then
diff = -1
End If

'need a more riggourous way of dealing with 0-255 transisiton
If diff = 255 Then
diff = -1
End If
'*****

If (diff <> -1) Then

newline = diff & "," & line
newFile.WriteLine(newline)
newline = diff & "," & line2
newFile.WriteLine(newline)
newFile.WriteLine("***Packet:" & c)

If (diff = -2) Then
l = l + 1
End If

If (diff < -2) Then
large = large + 1

```

```

    'This calculation is a bit buggy as it fails for losses >255,
    'or larger losses which straddle 255
    large_tot = large_tot + Math.Abs(diff + 1)
End If

If (diff = 0) Then
    N = N + 1
End If

End If

Label_DebugBox.Text = "Analysing file: " & line
line = oldFile.ReadLine()
line2 = oldFile2.ReadLine()
c = c + 1

End While

newFile.WriteLine("")
newFile.WriteLine("*****")
newFile.WriteLine("          ---STATS--- ")
newFile.WriteLine("*      Total Packets Received:      " & c)
newFile.WriteLine("*      Single Packets Lost:         " & l)
newFile.WriteLine("*      Larger Loss Events:          " & large)
newFile.WriteLine("*      Total packets lost (buggy):  " & (large_tot + 1))
newFile.WriteLine("*      Average Loss size (buggy):   " & ((large_tot + 1) /
(large + 1)))
newFile.WriteLine("*      Packets Copied:               " & N)
newFile.WriteLine("*")
newFile.WriteLine("*****")

oldFile.Close()
newFile.Close()

```

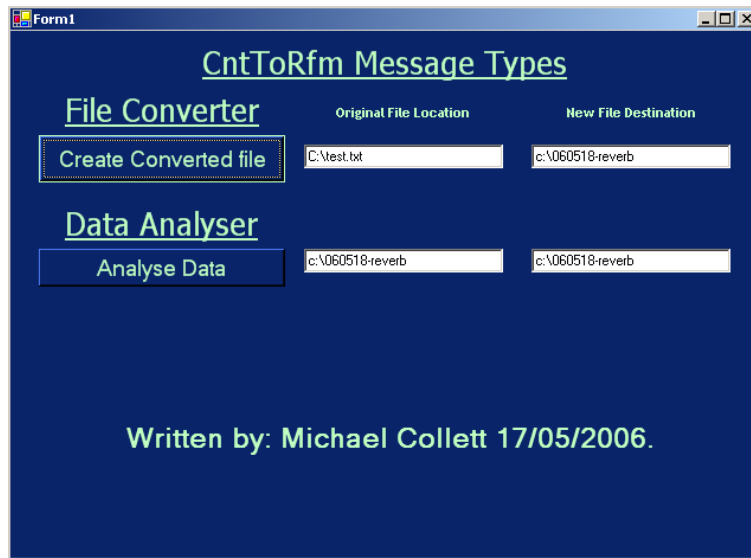


Figure 28: Software Interface for Analysis of Lost Packets

8.5 Readme File from Vanderbilt University on Their Ranging Algorithm

TestAcousticRanging:

Authors/Contact: bogyom@isis-server.isis.vanderbilt.edu (Gyorgy Balogh, ISIS)
and miklos.maroti@vanderbilt.edu (Miklos Maroti, ISIS, Vanderbilt)

DESCRIPTION:

This component exercises the acoustic ranging component (OutsideRanger). It periodically emits an acoustic ranging signals (one radio message followed by several buzzes) and at the same time listens for them. When a ranging measurement is made, then it forwards the distance measurement to the base station. This application runs on the MICA motes only with the standard sensor board (with mic and buzzer).

The application turns on the red LED when sending the acoustic ranging signal, turns on the green LED when receiving an acoustic ranging signal, and blinks the yellow LED periodically (10 times per second).

ACCURACY:

For the outside ranger, the maximum range is around 10 meters (depends on GOOD radio and antenna). The average of the absolute value of distance error should be less than 10 or 20 centimeter. This value does not depend on the distance.

The manufacturing differences in the microphones and buzzers should not influence the accuracy of the ranging.

OPERATION:

0. You have to download the FULL minitasks/02/vu CVS directory from SourceForge to a new local directory (do not load it over your tinycos-1.x directory).

1. Program several (up to 10-15) MICA motes with the TestRangingC component. Use unique mote ids. Attach the standard MICA sensorboard (with mic & buzzer) to these.

2. Program one MICA mote with the BufferedGenericBase (or GenericBase) component with id 1973.

3. Turn on each of the motes, waiting a couple of seconds in between.

4. The buzzer motes should buzz around once in every minute, watch the LEDs to monitor their operation.

5. Deploy the motes OUTSIDE, for example in a 10x10 meter area.

6. Connect the base station mote to a PC/laptop and run the following program:

```
java isis.nest.printmsgs.PrintDiagMsgs -comm COM1
```

This program will display the distance measurements on the screen. The format of each line is:

```
"ranging" <buzzer nodeid> <listener nodeid> <distance in centimeters>
```

The distance value -1 will indicate that the radio message was heard, but the acoustic signal was not clear enough to make a good distance estimate.

7. Enjoy. Oh, do not forget to take your measuring tape :)

LIMITATIONS:

- This algorithm is optimized for OUTSIDE and STATIONARY use. Inside buildings there can be substantially more echoes. We plan to release an acoustic ranger for inside use and for moving targets.

- We use a primitive algorithm to avoid two motes sending the acoustic signal at the same time. However, if the motes cannot hear each other (in one hop) or started at the wrong phase, they can still buzz at the same time. Listen for this and stop/restart these motes.

ALGORITHM:

- First we emit a radio signal. The SendMsg.sendDone and ReceiveMsg.receive events are used to synchronize the acoustic beacon and acoustic rangers.

- When receiving the radio signal, we turn on the microphone. Both sides wait 0.5 seconds for the microphone to power up.
- Then we emit 16 buzzes, each of which is 122 millisecond long. The acoustic ranger samples each of the buzzes by streaming the microphone (950 samples per buzz) and adds these samples together. The sampling frequency is around 15600 HZ on the MICA and a little bit higher on the MICA2s.
- By sampling 16 times we increase the signal to noise ratio. Also, we vary the delay between the buzzes to eliminate some of the unwanted echoes and other notes' buzzes.
- Once the recording is done, we use a 35-length digital filter (with integer coefficients) to filter out the 4-5 KHZ range.
- Then we run a nice peak detection algorithm to detect the start of the buzz.
- Finally, we convert the range to centimeters.

8.6 Software for Analysing Ranging Data

In much the same way as described in sections 3.1 and 8.4 the data from all the ranging estimates were extracted from raw XListen outputs. The main difference is that the data were exported Excel spreadsheets. This was chosen predominantly to make sorting the estimates into columns easier. It also enabled some basic statistical analysis and visualisation to be carried out.

The code below reads the raw data file, calls functions to extract sounder and listener information and then sorts the ranging estimates into an Excel spreadsheet. There is a clause which rejects packets below a certain length. This is to prevent the triggering packets being read as data.

```

column = 1
'Loops through raw file, line by line
While (line <> Nothing)

'7e = packet header
'With no extra escaped bytes, line should be 57 characters
'With escaped byte, line will be 59
'56 or less will be a bit of lost data
If ((line.Substring(0, 2) = "7e") And (line.Length > 56)) Then
'DebugMsg type = 177
If (getMsgType(line) = 177) Then
Label_DebugBox.Text = "Converting File: " & line
row = 8
column = getColumn(line) + 1
cell = xl.ActiveSheet.Cells(row, column).Value

While (cell <> Nothing)
row = row + 1
cell = xl.ActiveSheet.Cells(row, column).Value
End While

xl.ActiveSheet.Cells(row, column).Value = getData(line)

End If

End If
line = oldFile.ReadLine()
End While

```

Below are functions which are called by the main program. They all extract data from the raw packets and convert the hexadecimal numbers to decimal. The function which extracts ranging data needs extra functionality due to the type of message packet. The payload numbers may vary in type. Thus, to read a number from this part of the packet, a type code must also be read in order to know how to convert the number back to a decimal. The reason for doing this is to increase the potential range of values. For example a standard hexadecimal number could not take negative values. There is also the added error checking

functionality. If the quoted data type is unrecognised, then -3 is returned.

```
Public Function getSounder(ByVal line As String) As Integer
    getSounder = 0
    getSounder = Hex2Dec(line.Substring(34, 2))
End Function
```

```
Public Function getListener(ByVal line As String) As Integer
    getListener = 0
    getListener = Hex2Dec(line.Substring(40, 2))
End Function
```

```
Public Function getMsgType(ByVal line As String) As Integer
    getMsgType = 0
    getMsgType = Hex2Dec(line.Substring(8, 2))
End Function
```

```
Public Function getDataType(ByVal line As String) As Integer
    getDataType = 0
    getDataType = Hex2Dec(line.Substring(46, 2))
End Function
```

```
Public Function getColumn(ByVal line As String) As Integer
    getColumn = 0
    getColumn = 7 * (getSounder(line) - 1)
    getColumn = getColumn + getListener(line)
End Function
```

```
Public Function getData(ByVal line As String) As Integer
    getData = 0
```

```
    If (getDataType(line) = 0) Then
        'Had mistake before:
        'The escape byte in the CRC was generating 7e data
```

```
    If line.Substring((line.Length - 15), 4) = "7d5d" Then
        getData = Hex2Dec("7e")
    Else : getData = Hex2Dec(line.Substring(44, 2))
    End If
```

```
End If
```

```
    If (getDataType(line) = 255) Then
    If (Hex2Dec(line.Substring(44, 2)) = 255) Then getData = -1
```

```

If (Hex2Dec(line.Substring(44, 2)) = 254) Then getData = -2
End If

If ((getDataType(line) <> 255) And (getDataType(line) <> 0))
Then getData = -3

End Function

```

The following piece of code performs some basic statistical analysis of the ranging data. The algorithm loops through each column of data and calculates the mean and standard deviation. It also counts the number of data points and error flags

```

'*****
'Data Analysis
Dim sum, sumsq, mean, stDev, diffSq, sumDiffSq As Double
Dim dataCount, errorCount As Int16

sum = 0
sumsq = 0
dataCount = 0
errorCount = 0
stDev = 0
diffSq = 0
sumDiffSq = 0
row = 8
column = 2
cell = xl.ActiveSheet.Cells(row, column).Value

While (column < 51)

While (cell <> Nothing)
If (cell >= 0) Then
sum = sum + cell
sumsq = sumsq + (cell * cell)
dataCount = dataCount + 1
End If

If (cell < 0) Then errorCount = errorCount + 1

row = row + 1
cell = xl.ActiveSheet.Cells(row, column).Value
End While

If (dataCount > 0) Then
mean = sum / dataCount
Else
mean = 0

```

```
End If

row = 8
cell = xl.ActiveSheet.Cells(row, column).Value

If (dataCount > 0) Then
While (cell <> Nothing)
If (cell >= 0) Then
diffSq = (cell - mean) * (cell - mean)
sumDiffSq = sumDiffSq + diffSq
End If

        row = row + 1
        cell = xl.ActiveSheet.Cells(row, column).Value
End While
End If

If dataCount <> 0 Then
'Estimate of the s.d. of the population:
    stDev = Sqrt(sumDiffSq / (dataCount - 1))
    Else : stDev = 0
End If

'Put calculated stats in table, see headings for layout
xl.ActiveSheet.Cells(3, column).Value() = mean
xl.ActiveSheet.Cells(4, column).Value() = stDev
xl.ActiveSheet.Cells(5, column).Value() = dataCount
xl.ActiveSheet.Cells(6, column).Value() = errorCount

'Reset Counters for next column
row = 8
sum = 0
sumsq = 0
dataCount = 0
errorCount = 0
stDev = 0
diffSq = 0
sumDiffSq = 0

'Increment column and update cell
column = column + 1
cell = xl.ActiveSheet.Cells(row, column).Value

End While

newFilepath =
filepathRaw.Substring(0, (filepathRaw.Length() - 4)) & ".xls"

xl.ActiveWorkbook.SaveAs(FileName:=newFilepath)
```