WiBeaM : Design And Implementation of Wireless Bearing Monitoring System

A Thesis Submitted

in Partial Fulfillment of the Requirements

for the Degree of

Master of Technology

by

VMD Jagannath

to the

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY,KANPUR

Jun 2006

CERTIFICATE

It is certified that the work contained in the thesis titled "*WiBeaM:Design And Implementation of Wireless Bearing Monitoring System*" by *VMD Jagannath* has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Dr. Bhaskaran Raman
Department of Computer Science & Engineering,
Indian Institute of Technology,
Kanpur-208016.

Abstract

A sensor network is composed of a number of tiny sensor nodes. These sensor nodes consist of sensing (internal/external), data processing and communication components. Such nodes combined with MEMS sensors can form a complete replacement for the existing cumbersome predictive maintenance systems used in many industries for maintenance of machinery.

We have designed a system that can be used to monitor the bearing conditions of induction motors in a ship. The system called as WiBeaM (Wireless Bearing Monitoring system) forms a network of sensor nodes that sense the vibrations on a motor and transmit the same to a central base station, where the data is further analyzed through signal processing for defect diagnosis. WiBeaM uses MEMS accelerometers, that are placed on the body of the motors to measure the vibrations generated by the bearings . We have used cheap but reliable off the shelf components to realise a real world application at a very low cost. Considering the cost and the ease of deployment, our application is one of its kind in the domain. The solution has been implemented on *Moteiv's Tmote-Sky nodes* using *ADXL105 accelerometer* from Analog devices.The solution consists of an application layer, transport layer for reliable data transfer and a MAC protocol for low power consumption. The challenges in the implementation span from the hardware to software. The aim was to select a cheap yet efficient sensor that could be compatible with the sensor node and build a software that can automat the otherwise cumbersome process of bearing monitoring.

With proper selection of the hardware and efficiently handling the measured signals we have married the MEMS and wireless technologies which were otherwise being used independently. Whilst developing the application we have found that by proper selection of transmit power of the radio, single hop wireless transmission is more power efficient than the multihop. We have taken an application driven approach to design and implement the *application and transport protocols.*

In sum, we have proved the viability of machinery vibration monitoring with wireless network technology by conducting extensive testbed trials.

Dedicated to

The Almighty and all those people who inspired me.

Acknowledgments

I am deeply indebted to DR Bhaskaran Raman who has readily accepted this idea of mine that I conceived during my tenure onboard INS Ranjit to metamorphose into reality. I am also thankful to Dr Kameswari Chebrolu for her constructive criticism during the initial phase of the problem definition and for initiating the hardware procurement in time. I also take this opportunity to thank all those stalwarts at U/C Berkeley who invented TinyOS and made embedded system programming an easier task. I would like to thank Hemanth Haridas and Nilesh Mishra for their constant help through out duration of the thesis. I would also like to thank M Naveen and Dattatraya Gokhale for all the meaningful discussions, that helped me in improving various aspects of my work. I would like to convey my regards to Lt Cdr LR Prakash without whose motivation I would not have seen this day.

I am grateful to Major Jasdeep Singh who went out of his way to help me by providing the testbed in the vibrations lab for conducting all the experiments in my thesis. I also thank B Laksminath for all the help in writing Perl scripts.

Last but not the least I am thankful to my wife Priya and my parents for all the emotional support they provided during the tough times.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Efficient operation of industrial plant machinery involves efficient monitoring of various operational parameters. In most of the industrial setups, sensing various parameters of running machinery is traditionally done with a human operator monitoring the values physically. In some cases there are semi automated systems where alarms can be triggered when a parameter reaches a threshold value. This can alert the watch keeper who handles the situation further. There are also, some intelligent systems available in the market that can handle critical situations and take preventive measures to avoid a breakdown of the system. But such systems are often expensive and cumbersome.

Condition Based Maintenance (CBM) or Predictive Maintenance (PdM) is a buzzword in any manufacturing plant or an engineering facility. This is a generalized term applied to a family of technologies used to monitor and assess the health status of a piece of equipment. It is basically a schedule involving a technician who takes rounds with a hand-held data collector at stipulated intervals of time (monthly/weekly/daily). Parameters like machine vibration, temperature etc are collected with these data collectors and the data is further analyzed using software like matlab. The machine's health is then projected, there by reducing the possibility of inadvertent failures and machines' downtime. Another form of CBM is the online surveillance system. These systems provide more frequent data than the former. Online surveillance systems include hardwired sensors connected to multiplexers that are networked to a main database computer. Though the online systems prove to be more efficient, they are not the primary choice because of their huge upfront costs in wiring.

Monitoring the condition of bearings in induction motors is a common predictive main-

tenance procedure used in most of the manufacturing plants, textile mills and marine engine rooms. Vibration measurement technique is largely used in bearing condition monitoring and has proved its reliability in practical systems. In this technique the vibration of the bearing is picked up from the body of the motor with the help of a sensor. Though various kinds of sensors like shock pulse meter, velocity meter, etc are available, the ones that are popular are the accelerometers because of their form factor and accuracy in measuring the vibrations.

1.1 Thesis Goals

This section is an enunciation of the problem description in detail.

1. Measure the bearing vibrations on the body of a motor.
2. Process the measured signal to capture the relevant data.
3. Efficient storage of the readings on the nodes for future data recovery in case of a system failure.
4. Form a network of all the sensors in the machinery rooms of a ship.
5. Transfer the data reliably to a centralized node.
6. Find methods to conserve battery power of the sensor node to increase its life time .
7. Find data compression methods.

There are various types (differing in the operation principle) of accelerometers available in the market of which the piezoelectric accelerometers are widely used because of their availability and facility of having integrated electronics. However standard piezoelectric accelerometers despite of their reduced cost are relatively expensive. Cost of a piezoelectric accelerometer from crossbow is 200USD approximately. One way to reduce the cost is to use a MEMS (micro electric mechanical systems) chip in which the accelerometer is one part of an integrated silicon chip. Development of cheap MEMS sensors has been increasing these days. The cost of a MEMS sensors from Analog devices lies between 10-20USD. The performance of micro-machined sensors [3], compared to piezo-electric devices is found to be poor because of their

high noise factor but they offer a cheaper alternative in condition monitoring with possible restrictions set by ambient factors (present MEMS sensors can withstand temperatures upto 70 deg C).

Despite the increased interest in this area, a significant gap remains between the existing designs and the requirements of industrial monitoring primarily because of the overhead of the high cost sensing equipment and initial installation costs. Through careful design and balance of technologies a wireless condition monitoring system that is dependable and economically viable can be built.

1.2 Thesis Contributions

We have designed and implemented a sensor network application that can monitor the bearing condition of induction motors onboard marine ships.Our major contributions to this work are the following

1. We have used a cheap and compact sensor to monitor the bearing vibration of motors.
2. We have successfully integrated the sensor to a wireless sensor node that will transfer the measured data wirelessly to a central base station
3. Venture the possibility of using a hardware solution for data compression. In this we developed a sixth order chebyshev filter for filtering the unwanted frequencies from the measured signal thus reducing the amount of data that needs to be sent to the base station for analysis
4. We have developed an application in TinyOS for measuring the vibration of the motors
5. We have implemented BMAC [25] a MAC layer protocol for reducing the power consumption by the nodes while idle listening. This can be used in case of multihop data transfers.
6. We have implemented a NACK based transport protocol designed by [19] for reliable data transfer

7. we have tested and proved the entire system on a vibration machine fault simulator with high accuracy .

During the laboratory trials we have achieved throughput of 35kbps.The rest of the report is organized as follows, chapter 2 describes the background and chapter 3 describes the related work in this area. Subsequently, chapter 4 explains the overall design of the system and the hardware/ software used and various layers of the network stack that have been considered in building our system. Chapter 5 explains the implementation details and chapter 6 presents various performance evaluation results. Chapter 7 documents the future scope of this project and conclusion.

Chapter 2

Background

The intended end usage of our wireless bearing monitoring system is a shipboard engine room/machinery space. This system would work in other industries too, like textiles and paper mills where there is a large usage of induction motors. This section explains the importance and difficulties involved in measuring the bearing parameters of induction motors. Readers with a marine background may skip this part. The following paragraphs will give a clear understanding of various facets of condition monitoring, we also discuss the theory of bearing defect diagnosis and the operating principle of MEMS sensors here.

2.1 Necessity of condition monitoring of bearings

Induction motors are generally employed on heavy duties where uninterrupted operation is necessary. A typical induction motor has two ball bearings, one on the drive end and the other on the non drive end. Bearings are used in motors to reduce the rolling friction between the rotating part and the stationary part. Induction motors are generally reliable due to their rugged and less moving parts, however most of the defects in an induction motor occur generally due to failure of bearings. Hence these bearings are closely monitored for any impending failure or microscopic defects in order to have enough lead time to repair the motor and avoid any major defects cascading due to the defective bearings.

2.2 Types of defects in ball bearings

A ball bearing consists of an inner and outer ring (case) between which there are metallic balls that rotate. Defects in the inner/outer case generally occur when there is a crack in the ring and the bearing has deformed. Defects in balls occur when one or more balls break or wear out due to excessive usage or high bearing temperatures. When a ball wears out it develops a gap in the ball race and hits other balls, leading to the damage of other balls too.

2.3 Various monitoring methods in practice

2.3.1 Manual methods

This is an age old method where an experienced operator finds the defect in a bearing by hearing the sounds emanating from the motor body with the help of a hollow pipe placed on to body of the motor. Though this method is still in use and accurate to an extent, it does not provide any time for repair, as it can be detected only when the bearing is damaged and it also needs an experienced operator to localize such defects.

2.3.2 Automatic and Semi-automatic methods

The following methods of defect diagnosis require skilled operators to manually go near each motor and measure the bearing condition with the help of hand held instruments.

Shock pulse method

The most common method of measuring bearings is done by a method called the shock pulse testing. A defect in the ball race of a bearing will produce shock waves due to the reasons explained in sec 2.2. These shock waves can be captured on the body of the motor with the help of a transducer. Hand held devices (manufactured by SPM instruments are common) with a probe are available in the market for measuring shock pulses. The operator needs to enter the bearing parameters onto the device and then put the probe on the body of the motor where the bearing is located. In this method the accuracy of the readings highly depend on the correct positioning of the probe and shock pulse measurement is not an accurate measurement of the bearing defect. Following are the disadvantages of such a

system: it is not automated, and needs a trained and experienced operator.

Measuring vibrations

All bearings have a resonant frequency of vibration, known as the characteristic frequency. This characteristic frequency can be calculated by measuring vibrations on the body of the motor, using a vibration meter. Vibration of a body can be measured by measuring the displacement, velocity or acceleration of that body. Various transducers like displacement meters,velocity meters, and accelerometers are available in the market to measure the respective parameters. Of these three types of sensors, accelerometers have the smallest form factor and are also available at reduced costs due to advancements in MEMS technology. Conventionally the vibrations are measured with hand held scopes that have integrated DSP chips and probes with piezoelectric accelerometers/velocity meters. These sensors have to be mounted on to the motors.

2.4 Necessity of an automated system

Industries using large number of induction motors for various purposes often rely on manual method of condition monitoring. Although this method looks cheap, it involves certain hidden costs like maintenance of hand held scopes, increased number of man hours in taking measurements and book keeping. To emphasize the requirement of an automated system we would like to give few anecdotal facts and figures derived from real life situation. A destroyer class of warship uses induction motors for various purposes like ventilation, fuel and oil pumping, sea water pumping, and other conversion machinery. The total number of important motors may count up to 400. The primary drawbacks in warships are the space constrain and location of motors that makes it difficult to take measurements, resulting in extra time consumption to carry out the measurements. To ensure that the readings are taken correctly, the probe is positioned at least at 3 different points each on the drive and non-drive end bearing. An anecdotal survey reveals that two men using an SPM meter would be able to take readings of a maximum of 20 motors in a day. To cut a long story short, an automated system with least human intervention would always be useful in these type of scenarios.

2.5 Principle of Operation of MEMS Sensors

MEMS stands for *micro electro mechanical system* In our application we have used a MEMS sensor *ADXL105* for sensing bearing vibrations. ADXL105 is a capacitive accelerometer. The chip consists of a micro machined capacitor and amplifier, the capacitive element of the accelerometer chip consists of electrodes connected to a frame and moving mass and a flexible membrane between them. Acceleration is the measure of capacitance between the moving mass and the supported frame.The structure of the sensing element can be compared to a spring mass damping system that can be described by a linear second order differential equation. More on this can be read from [12] and [13].

2.6 Theory of Bearing Defect Diagnosis

This section discusses detection of defects in a bearing by measuring vibrations. Apart from the traditional methods of bearing defect diagnosis mentioned in the previous paragraphs, there is one more method which has been in research for a while. This is the stator current analysis method. Bearing defects in a motor can cause eccentricity on the rotor, which generates transients in the stator currents. These transients can be picked up by a current transducer and a correlation between the transients and the bearing defect would be deduced. This method was tried out by [3] but it turned out to be inaccurate. Whichever may be the method, but from a practical point of view, it should reliably detect a defect in the bearing. There are other methods like acoustic noise measurement and temperature measurement, but vibration measurement is by far the most reliable method of defect detection in ball bearings.

Ball bearings contain a series of internal elements that roll between an inner and outer race. These types of bearings have very low clearances, hence there is little damping on the rotor. If a motors is used extensively, internal clearances will expand due to the wearing of the internal elements(balls). Due to the above mentioned characteristics these type bearings possess a relatively short life span. Due to this finite lifespan, it is desirable to monitor these bearings for early indication of any impending failure.

Primary dynamic excitation(vibration) in a bearing fundamentally occurs at the shaft rotational frequency. In a roller bearing, this fundamental shaft vibration is supplemented

Figure 2.1: Cross Sectional View of a Ball Bearing - courtesy [26]

by the mechanics inherent to the moving parts of the bearing, detailed information about bearing design can be further read from [9]. Based upon the bearing geometry and the rotational speed, it can be shown that the defects in a bearing can be of three types, the outer cage defect, the inner cage defect and the ball race defect, frequencies generated by all these three defects can be identified by the following formulae :

for defect in the outer race

$$\left[F_{ord} = \left\{\frac{N \cdot RPM}{2}\right\} \times \left\{1 - \left(\frac{d_{ball}}{D_{pitch}}\right) \times cos\alpha\right\}\right]$$

for defect in the inner race

$$\left[F_{ird} = \left\{\frac{N \cdot RPM}{2}\right\} \times \left\{1 + \left(\frac{d_{ball}}{D_{pitch}}\right) \times cos\alpha\right\}\right]$$

for a ball defect

$$\left[F_{ball} = \left\{\frac{RPM}{2}\right\} \times \left\{\left(\frac{D_{pitch}}{d_{ball}}\right) - \left(\frac{d_{ball}}{D_{pitch}}\right) \times (cos\alpha)^2\right\}\right]$$

α is the contact angle

D_{pitch} is the outer pitch diameter

D_{ball} is the ball diameter

and N is the number of balls

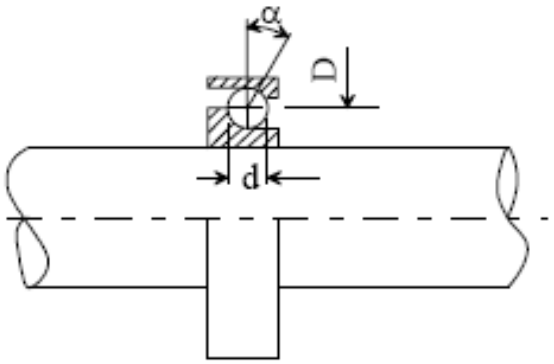

Figure 2.2: Geometry of a ball bearing Courtesy [9]

The above frequencies are for ideal bearings, where the inner cage rotates with the shaft and the outer cage is stationary and the balls roll. Bearing dimensional data is often available directly from the manufacturer. Some suppliers have a list of the fault identification frequencies for their bearings. But in practice the balls not only roll on their races, but they also slide. This sliding motion can be taken care by multiplying a sliding factor ϵ to the above equation. The value of ϵ is usually 0.8 to 1.0. But in practice these bearing parameters are not always available, an alternate way of calculating the frequencies is by using the following approximate equations developed by [10].

Frequency of the outer race defect can be found by $f_o = 0.4Nf_r$

Frequency of the inner race defect can be found by $f_i = 0.6Nf_r$

where N = the number of balls f_r = the rotational speed of the rotor

2.7 Proposed solution

In this thesis we developed a prototype wireless bearing monitoring system consisting of a network of *wireless sensors nodes*. From here on, the word system and WiBeaM are used interchangeably through out this document. These *wireless sensor nodes* would sense the vibrations on the body of induction motors with the help of a MEMS accelerometer (ADXL105 manufactured by the Analog Devices). This measured vibration data (analog) is converted

into digital data and is routed to a centralized node called the base station with the help of sensor nodes. At the centralized node this raw data can be analyzed to extract the bearing condition information. The figure 2.3 shows an overview of the system.

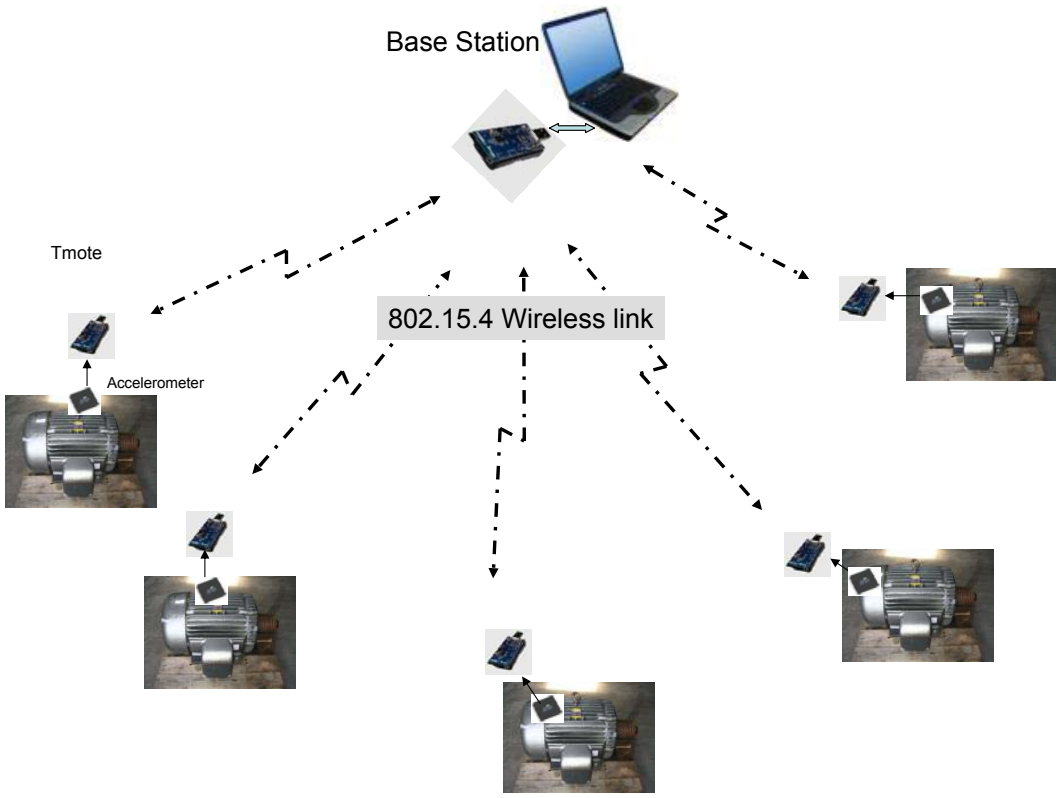

Figure 2.3: Overall View of the System Components

Chapter 3

Related Work

Several companies have developed telemetry systems [5] that are a step in the right direction for automated bearing monitoring, such as Wilcoxon's BlueLynx and Oceana Sensor's ICHM 20/20. In addition, Rockwell has the HiDRA, and SKF promotes its wireless system. These systems are basically a simple, wire replacement between the sensor and the data collectors, hence they do not succeed in cost reduction. They connect a couple of hard-wired sensors to a radio hub, so significant cable installation is still required and the costs of such systems are still high because of the proprietary hardware and development costs. Wireless sensor networks are an emerging technology that can replace these systems with low cost hardware devices called the sensor nodes and open source software like tinyOS etc. These networks use multihop radio communication to provide better fault tolerance. However this comes at a cost of increased complexity in the software.

Various sensor networks projects like structural monitoring [2], habitat monitoring [4], CodeBlue [6] and ZebraNet [7] built by academic institutions are working satisfactorily and are a testimony to the fact that sensor networks are going to make a considerable impact on various fronts.

Habitat monitoring [4] is similar to our work in a broad sense, i.e it deploys nodes on the Great Duck Island to monitor the behavior of the birds. It uses a tiered architecture with both single and multihop topology. The network does periodic collection of small amounts of environmental data. Analysis of the data provides an insight into choices between the single/multihop topology, the network structure and route stability. We have instead focused on the reliable transfer of the data by having a hop by hop recovery and reduction of power

consumption by implementing the B-MAC protocol.

WISDEN [2] is a multihop wireless data acquisition system designed to monitor the health of large structures. This work is an attempt to replace the existing wired system to collect vibration data from various parts of the structure. It employs reliable data delivery mechanisms and a light weight synchronisation mechanism.As the data requirements surpass the radio bandwidth, some intelligent compression methods are employed. The deployment is short term comprising of fourteen sensor nodes and a laptop. There is no tiered architecture implemented in this. Sensor nodes form a network to feed the data to a computer via the basestation. The sensors used here are accelerometers.

Most of the existing systems are intended for deployments of a large number of nodes that make use of traditional in-network aggregation, however most of the industrial monitoring systems cannot make use of this since it is not meaningful to combine data from multiple machines. And also the data rates can be extremely low as the monitoring is done only at regular intervals of time with large gap between intervals (approximately 24 hours). The work which has a very close resemblance to ours is deployment of industrial sensor networks by Intel corporation [7]. This is a large scale deployment of sensor nodes spanning over various machinery spaces of an oil tanker. They have described the experiences and lessons learnt in two settings: a semiconductor fabrication plant and an oil tanker. The authors in this paper have established design guidelines by providing a study of the impact of platform on the architecture by comparing two different sensing platforms Mica2 and Intel motes. They have also provided a comparison of different applications of the same class and have given a demonstration of return on the investments. The sensors used here are piezoelectric accelerometers that are high in performance and cost. They used *pump slow fetch quick* (PSFQ) protocol [24] for reliable data transfer. The architecture is a hierarchical architecture where sensor nodes send data to a gateway, which sends it further to a root gateway. Data is then transfered to an enterprise server via internet.

CODEBLUE [6] is an application developed for hospitals. This system employs a software framework, that provides protocols for device discovery, publish/subscribe multihop routing and a simple query interface system allowing caregivers to request data from groups of patients. It integrates an RF based localisation system, to track the location of the patients

	Habitat Monitoring	WISDEN	North Sea Deployment	BriMon	CODEBLUE	**WiBeam**
Deplo yment	Long Term	Short Term	Long Term	Long Term	Long Term	Long Term
Hardware	Mica2	Mica2,MicaZ	MicaZ	Tmotes	Mica2,MicaZ	Tmotes
System Replaced	manual	wired	Expensive wireless	manual	Medical Electronics	manual
Archit ecture	Tiered	Flat	Tiered	Tiered	Tiered	Tiered
Sensor	Temp erature Pressure	accelero meter	accelero meter	acclero meter MEMS	Pulse oximeter EKG	accelero meter MEMS
Compression	YES	YES	NO	NO	NO	NO

Table 3.1: Comparison of the existing Sensor Network Systems

and doctors. The publish/subscribe routing layer helps in relaying data from multiple sensor devices to all receivers those who have registered for the data. The routing layer is based on adaptive demand driven multicast routing layer. It also has a Java based user interface that can be understood easily by medical personnel

In our work we have emphasized on building a cheaper alternative of a PdM, with reliable data transfer and ease of installation. The MEMS accelerometer that is used here has a smaller range of measurement than the piezo electric accelerometer and cannot withstand temperatures(above 70 deg C). With the advancement of MEMS technology, production of high performance MEMS accelerometers that can match the performance of the present day piezoelectric counterparts is not far away.

Chapter 4

Design Overview

TopDown and BottomUp are two different design strategies that are generally considered while designing a system. In the topdown approach instead of concentrating on the lower details, an overview of the system is formulated first. Each subpart may then be dealt separately with more focus to the minor details in an iterative process. The top down approach is designed with the help of *black boxes* that are used to build the design. Though there are certain advantages to this approach, it is generally incomplete and insufficient to understand the minor details of any system. In contrast the bottom up approach first looks at the basic building blocks of the system and then integrates these blocks to make them into bigger components and thus has a pyramidical type of approach to a problem. *WiBeam* has been designed with a bottom up approach, with more emphasis on the lower level details. This chapter describes the factors considered for designing the hardware and software components. The rest of the chapter is organised as follows, the section *operation cycle of motors* explains the usage of various types of motors on ships, this would further help in designing the application protocol in software. The *hardware design considerations* section discusses various available sensors in the market and their advantages and disadvantages. The *software design consideration* section explains the reasons for choosing TinyOS and gives an overview of the software architecture of WiBeam.

4.1 Operation Cycle of Various Motors

All the motors in a ship, depending on their role can be broadly classified into three categories, important motors, less Important motors and unimportant motors. These motors follow the following operational cycle respectively:

- All important systems like fire fighting system, refrigeration plant, AC plants etc are employed with more than one motor and these motors are operated in a duty cycle to avoid overloading of any one motor i.e if a motor is switched ON it will run for a certain duration and is then shut down for a while and the second standby motor will be used for that duration. The context here is of naval warships hence all the examples would be specific to the warships, however these same concepts would also be applicable to all marine ships. For example in a destroyer class of ship the fire fighting system is equipped with seven pump motors and at a given time any four motors will be running to maintain pressure in the firefighting system. Similarly there are six air conditioning plants out of which four are run during the operational status and three are run at harbor. A typical duty cycle of a motor in this category is six hours on and six hours off.

- A few motors though important, need to run only when their associated system runs and do not follow any duty cycle. They are operated along with the machinery they are associated with, for ex: the sea water cooling motors for the main engines or the fuel/lub oil pumps.

- And all other motors are operated continuously until their scheduled maintenance period, for ex: the ventilation motors for the living spaces.

WiBeam has been designed keeping in view the generalised case where in motors are operated in a duty cycle. Bearing readings for engine room motors are measured weekly/biweekly, taking readings more than this rate is not practically feasible due to the large number of motors onboard and the added disadvantage of cumbersome procedure of the measurement. Having an automated system would help in taking readings even daily or more than once in a day and would also help in analysing the vibration based defects in a more intelligent

manner. Keeping in view of the duty cycle of the motors, motes are designed to follow a sleep/wake cycle in order to capture the motor readings atleast once in a day. Since the motors run for six hours at a stretch, the sensor nodes can wake up once every four hours and check for an activity on the motor, if an activity is detected the mote will measure the vibrations and store it on to the flash and simultaneously transfer the data to the base station. Once a reading is taken in a day, the mote will go to sleep and will wakeup only the next day. Here sleeping corresponds to the mote's radio being switched off. The other components like the microcontroller and timers will be continuously running. Since the radio is the maximum power consuming in the mote it is essential to switch off the radio when there is no activity on the channel. Conserving the radio activity saves power and therefore increases the motes life time.

Detection of activity on the motor can be accomplished by two ways, one is by threshold based sampling of the vibrations. If the motor is not running the vibrations sensed by the accelerometer will lie below a certain threshold, this threshold can be determined by a trial and error method. The other way of detecting activity of the motor is by sampling the main power supply at the motor terminals. For the mote to be able to monitor the power supply, it is necessary to tap the main power supply of the motor and step down the voltage(with the help of a resistor) and connect it to one of the ADC pins of the mote. Care should be taken so that the input voltage to the mote does not exceed the permissible voltage level that can be safely connected to the ADC pin of the mote. Though this has not been done in the present set of experiments, it can considered as an option. If the main power supply of the motor can be tapped without involving much of hardware, power for the mote could also be supplied from the motor mains itself. This possibility would be studied in detail in subsequent deployments.

The mote needs to take one measurement in a day. So with out switching ON the radio component it can stay in a low power mode with a timer running, this timer will fire once every four hours till the time a measurement is not taken for the day. Once a measurement is obtained (the mote should wakeup when the motor is running) this timer will stop and will fire again after 24 hours. To transfer the measured reading to a centralized base station the node should switch ON its radio and poll the base station. If the base station is free it

	Power	Range	Freq Band	Sensitivity	Noise	Cost
ADXL105 MEMS	$2-5V$	$\pm 5g$	$0-12KHz$	$225-275mV/g$	$225mg$	$14USD$
CXL04 XBow	$\pm 5V$	$\pm 4g$	$0-100Hz$	$500mV/g$	$10mg$	$185USD$
SKF CMSS786A	$18-30V$	$\pm 80g$	$0.5-14KHz$	$95-105mV/g$	$20mg$	$120USD$
CMCP-1100	$8-12V$	$\pm 50g$	$0.3-10KHz$	$100mV/g$	$4-8\mu g$	$130USD$
Wilcoxon 786A	$18-30V$	80g	30KHz	$100mV/g$	–	185USD

Table 4.1: Comparison table of various Accelerometers

will respond with an acknowledgment. On receiving the ack the mote reads the data from the flash and transfers it to the base station for further diagnosis.

4.2 Hardware Design Considerations

The aim of this implementation was to develop a product that involves minimum hardware in a small form factor and is operable at low powers. The system should also be easily mountable on to the motor without any modification to the existing setup. There are many devices available for sensing vibrations like displacement meter, velocity meter etc, the most popular instrument used is the accelerometer. There are various accelerometers available in the market differing on the operating principle for example: charge amplifiers, piezo-resistive and piezo-capacitative accelerometers. We considered using various accelerometers before we chose ADXL105. Table 4.1 shows a comparison of parameters and cost between various accelerometers available in the market. Each of accelerometers other than ADXL105 were not suitable to us in one or the other way.

The table 4.1 shows how one or the other factors are not compatible with our requirements in most of the sensors. Inspite of having the advantages of low noise and power supply the Crossbow sensors are costly, the SKF and Wilcoxon sensors are not suitable because of the high input power supply. We wanted a sensor that could operate on two AA cells. The CMCP accelerometers also have the problem of power supply and cost. ADXL105 was an obvious choice because of its suitable parameters and the small form factor compared to others (not mentioned in the table).

Model	Radio	Data Rates	RAM	Processor	I/OInterface
Tmote Sky	2.4GHz	250kbps	10KB	msp430	10 Pin
Intel	Bluetooth	750kbps	64KB	ARM7TDMI	USB-slave mode
Mica2	916MHz	38.4kbps	4KB	Atmega128	51-pin
Micaz	2.4GHz	250kbps	4KB	Atmega128	51-pin

Table 4.2: Comparison table of various Wireless Sensor Nodes

4.2.1 MEMS Sensors

Conventional piezoelectric accelerometers are accurate, sensitive and have a frequency range that is adequate for measuring the vibration of electric motors, but they fall behind MEMS sensors when compared to form factor and cost.They also need considerable amount of auxiliary electronics in practice. While gaining mileage over the conventional accelerometers, MEMS sensors has a disadvantage of the higher noise level. MEMS sensors produce noise that is ten times more than the conventional piezoelectric accelerometers. Also MEMS sensors have an amplitude range that is much lower than their counterparts. The amplitude range of ADXL105 is only 20mg to 5g, while the others mentioned in Table 4.1 is much higher. Though there exist other MEMS accelerometers with higher acceleration range, but their noise levels are too high and sensitivity too low to be used in electrical motors. Another added advantage of using the ADXL105 is that it has an inbuilt temperature sensor that can measure temperatures upto 70^0C. It also has an inbuilt uncommitted amplifier i.e,the amplifier has not been used and left a the disposal of the user. This amplifier can be used to amplify the output signal of the accelerometers, this facility could be used at places where the vibrations on a motor are very feeble. This amplifier can be used by connecting external resistors as shown in [15].

4.2.2 Wireless Sensor Node

The other major component in the system is the wireless sensor node. A wireless sensor node is a unit consisting essentially a microcontroller, a radio module and sensors, few also have an external memory. These sensor nodes are also known commonly as motes. We are using Tmote Sky sensor nodes manufactured by Moteiv corp [14].

Figure 4.1: TmoteSky - Wireless Sensor Node

There are various type of nodes available in the market. A comparison is shown in Table 4.2 between the most popular nodes that are in use. Though each model has its own unique advantage we chose to use the Tmote sky because of its low cost, common external interface and a radio with high data rates. Moreover it is provided with a USB interface for loading the programs on to the microcontrollers flash and runs on two AA batteries.

4.3 Software Design Considerations

4.3.1 Choice of Operating System

The operating system software we use is TinyOS. This is a component based operating system specifically designed for the sensor network systems by University of California at Berkeley. Since our focus was to develop a sensor network application, we chose to use TinyOS which has already been established itself as the operating system of choice for a number of related applications and it is distributed freely with Tmotes.

The use of TinyOS has the advantage of leveraging a large number of available application components.Further more it has a lot of sample applications that are platform specific and can be used to test the desired feature of a mote. It is based on gcc and can be freely

downloaded from UCB's website [1]. We started developing our application with Tinyos-1.x and halfway through we found that there are few necessary components that are available only in Tinyos-2.x. Redesigning these components for Tinyos-1.x would be re-inventing the wheel, hence we ported our full application to Tinyos-2.x. We have used Boomerang [14] a freely downloadable package that can be used for developing applications on Tmotes. Boomerang is based on Tinyos-2.x.

4.3.2 WiBeam Software Architecture

The system is designed in layers that interact with each other.Layering is a natural choice of implementation of application in networks. Our system basically works on the application layer, transport layer and the physical layer. This concept is also synonymous to the way TinyOS is implemented. TinyOS provides a layering abstraction, where in lower layers provide certain interfaces and upper layers use them. We have implemented a power aware MAC layer called BMAC [25] that would be useful in conserving he idle listening power for multihop routing.

Application Layer Protocol: The application layer handles the measurement of data,storing the data on to the flash and sending the measured data to the next layer i.e, the transport layer. The sequence of data is like this, when ever a timer(for data measurement) fires, the application will start sampling the ADC pin and measures the vibration data.Once data is measured,it is written on to an external flash for future retrievals. Once th data is written on to the flash it is read again from the flash and sent to the transport layer. It accomplishes these tasks by calling various commands in TinyOS. For example it calls a command called SendFile.Send() and gives necessary arguments to the command.This command gets executed and the file is handed over to the transport layer. Once transport layer finishes its job it signals an event SendFile.sendDone(). This is an indication to the application layer that the file has been handed over and it can reset it buffers to get ready for th next file transfer. Similarly it calls a command BlockWrite.write()to write data on to the buffers. Once the data is written on the flash an event BlockWrite.writedone() is fired indicating that the operation is completed.

Transport Protocol: We have implemented a transport protocol designed by one of

colleague [19]. The transport layer is responsible for partitioning the data file received from the application layer into small segments that can be put into the payload. The data is further sent to the MAC Layer for further operations. It does the reverse operation when it receives packets from the MAC layer i.e, it extracts all the application data from the received packet payloads and merges them into a single file and then sends the file up to the application layer. Apart from this the transport layer is responsible for reliable transfer of data. To ensure reliable transfers, the transport layer adds sequence numbers to each packet so that the receiver can keep a track of the missing packets. Once an entire file is transferred the receiver is supposed to send an acknowledgment. If the file is received completely without any packet losses, the receiver sends an ACK message. If some packets are lost it sends an NACK (negative acknowledgment) message. In this NACK message it sends the missing sequence numbers, with the help of which the sender will retrieve the data from flash and resend it. Once an ACK is received the sender will send a tail message to indicate the receiver that the file transfer is completed.

Media Access Layer: We have implemented an existing protocol called BMAC [25]. Radio module is the biggest power consumer in the entire system.This protocol reduces the power consumption of the radio by putting it to sleep when there is no activity on the channel.In other words it reduces the idle listening power of the radio. It also senses the channel for activity before sending a packet. Traditional MAC protocols employ a threshold based scheme for doing a clear channel assessment but BMAC samples the channel continuously and calculates an exponentially weighted moving average of the received signal strength. Based on this moving average it decides whether the channel is free or not. This scheme reduces the chance of false alarms during clear channel assessment. How ever this protocol is useful to reduce power consumption only in a multihop scenario where the nodes need to be awake for routing data.

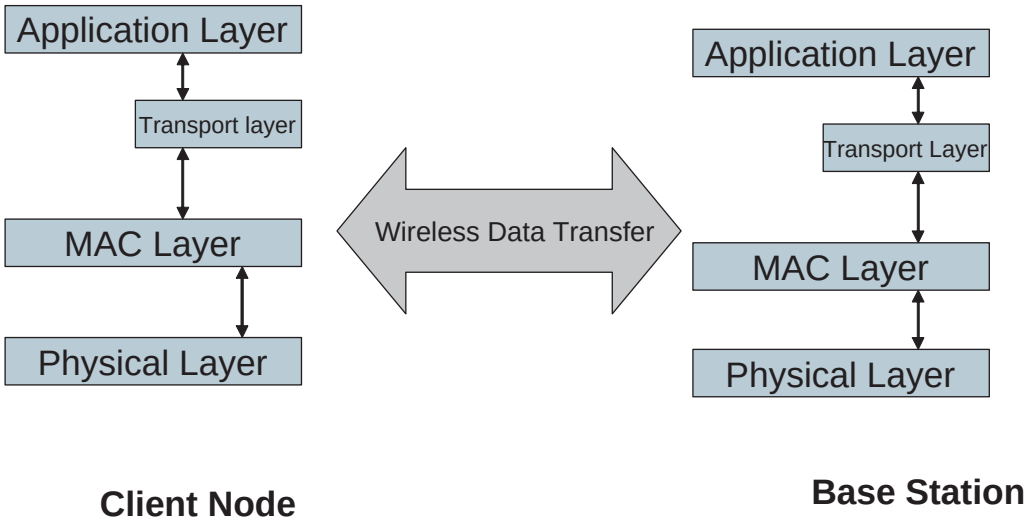

Figure 4.2: Layered Architecture

Chapter 5

Implementation Details

This chapter consists of the implementation specific details of both the software and hardware components. The hardware section explains how the accelerometer is connected to the mote and what precautions are to be taken while connecting the output of the accelerometer to the mote. It also shows where the accelerometer is positioned on the testbed for measurement.The next section is the software section, it explains in detail the interaction of various layers in the application and also the describes the function of various components and their interfaces Logical flow and physical flow of the data is shown in pictorial form. Message exchange between the base station and each node is shown in the form of a interaction diagram. Implementation details like flash storage and data measurement formats are clearly explained.

5.1 Hardware Implementation

The hardware implementation was done in several phases. The output of the accelerometer is a continuous analog voltage proportional to the acceleration being sensed. The output is ratio metric to the input supply. The accelerometers output is $250mV/g$ and for an input voltage 5V the accelerometer gives a 0g output of 2.5V. Since the maximum input that can be applied to tmote at the ADC input pin is 2.5V, the accelerometers 0g output voltage has to be reduced. This can be done by reducing the power supply of the accelerometer. After few trials it was found that giving a 3V input to the accelerometer would give an output of 1.5V at 0g. This also came as an advantage, as the aim was to have only a single source of supply that can power both the mote and the accelerometer. Power can be drawn from the expansion

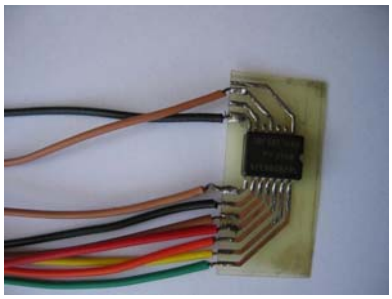

Figure 5.1: ADXL105 soldered on a PCB

connector of the mote and can be given as input to the accelerometer. This arrangement can be done during the actual implementation,but for the purpose of experimentation, the accelerometer was powered with an external battery source(a different set of batteries were used for the mote).

The sensor is an integrated chip, for ease of mounting it has been soldered on a printed circuit board(PCB). Since the shape of a motor is cylindrical the surface has a curvature, hence the PCB needs to be flexible so that it can be bent and glued on to the surface of the motor. For this purpose a special wafer-thin flexible PCB material with a dielectric thickness of 4.0 mils was used. The PCB was designed in a PCB designing software called protel, the tracks were etched inhouse. Care had to be taken while soldering the IC on to the PCB as the distance between the pins of the chip is 0.5 mm only. A snapshot of the IC soldered over the PCB is shown in figure 5.1.

For our initial experimentation we developed a low pass filter to filter out the unwanted high frequency components from the output signal of the accelerometer. The accelerometer has an output of 10Khz bandwidth. To reproduce such a signal faithfully we need to sample it at a frequency of 20Khz. Since we were measuring frequency components of frequencies below 100Hz we could filter out all the frequency components above 100Hz and sample the analog signal at 200 samples per second or may be slightly more as per Nyqyist criteria. But during the course of designing the lowpass filter we found that it is difficult to obtain a perfect filter with the available resources. Though we have developed a working chebyshev sixth order filter, we found that having a hardware filter not only makes the system bulky but

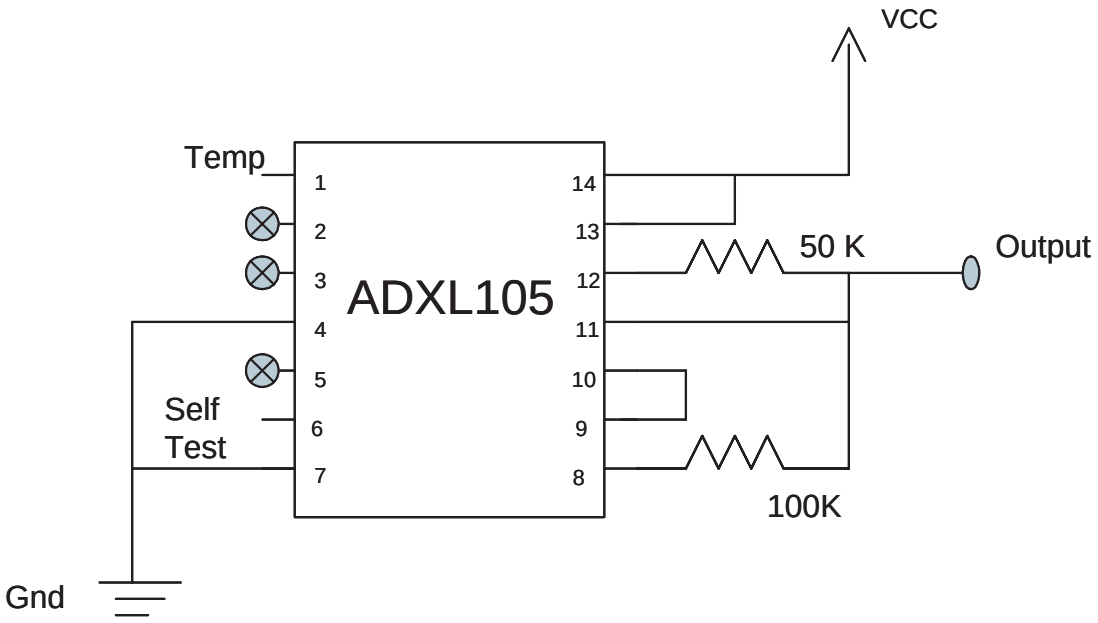

Figure 5.2: Schematic circuit diagram -ADXL105

there is also a chance of the electronic components inducing some vibration and corrupting the measurements(as the resistors and capacitors would be resonating on their legs). If the raw output signal is to be sampled at Nyqyist criteria, it requires the sensor node to sample the signal at a minimum frequency of 20Khz. In the initial stages of development of the software we were unable to sample the signal above 1Khz hence we recoursed to develop the hardware low pass filter that would eliminate the requirement of doing a high frequency sampling. But as the hardware size was increasing we had to rely on the software to do high frequency sampling.

5.2 Software Implementation

Figure: 5.4 shows how the data flows different layers in our application. The flow is of two types one is the logical flow and the other is the actual flow of data between the layers and between the nodes.

Figure: 5.5 shows the actual interaction between various components(layers) of the soft-

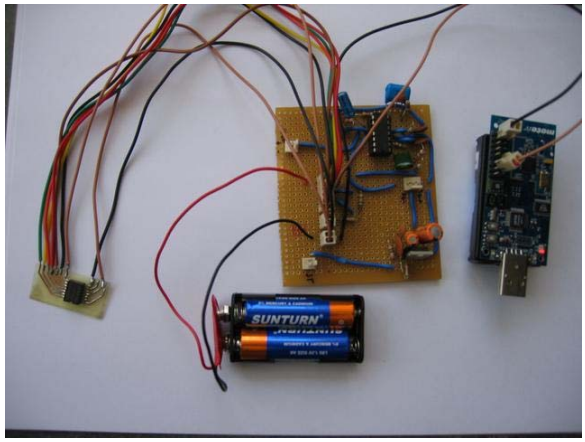

Figure 5.3: Circuit connecting the sensor and the mote

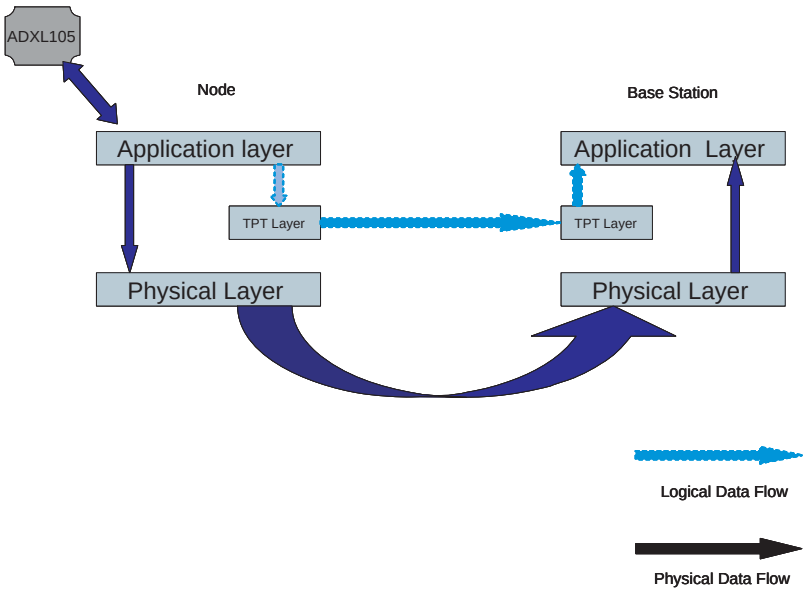

Figure 5.4: Data flow between various components

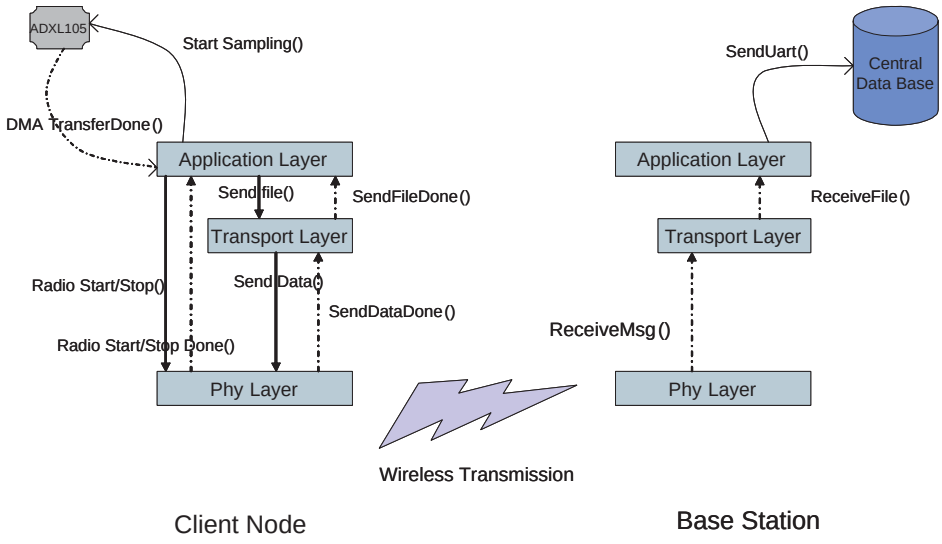

Figure 5.5: Component interaction between various layers

ware. As mentioned earlier components in TinyOS can be broadly classified into two sets, ones that provide interfaces and the others that use these interfaces. We have developed our application on similar lines, where each layer is a separate component that *provides*/*uses* certain interfaces. The whole application is divided into three major components (corresponding to each layer). The application layer component,the transport layer component and the physical layer component. We would like to acknowledge at this point that the idea of Transport Layer component is adapted from one of our colleague's thesis work [19].

The nodes are organised in a star topology where there are several client nodes and one central base station. Each node is connected to a single sensor and is independent from other nodes. Each node measures data and transfers the data reliably to the base station. The base station further transfers the data to a database server for further analysis. Roll of a node (client or base station) will be differentiated by the code running on it. It also need not write the data onto the flash, instead it will be sending the data continuously to the PC via UART. The radio on the base station is never put to sleep mode.

5.2.1 Power Consumption

Since power consumption is a major concern and the application has been designed to minimise the radio usage, as radio is considered to be the major power hogger in the whole application. Though we have implemented the BMAC protocol [25], we have realised that our application doesn't require a multihop transfer of data. We have proved in chapter 6, that single hop transfers consume lesser power than multihop. The power consumption can be further reduced in our application as the measurements are taken only once in a day and the radio need not be ON throughout. An interface called *RadioSleepTime.nc* has been designed for the purpose of radio power consumption. This interface has a command named *sleeptime(uint32_t time)*and an event *continuousAwakeDone()*. Whenever the *sleeptime* interface is called by a higher layer component it will pass a variable named *time* that will put the radio to sleep mode for the amount of *time* specified by the higher layer calling the command. The moment this timer is fired the radio is woken up and the event *continuousAwakeDone()* is signaled to the higher layer indicating that the radio is ON. The implementation of this interface is done by modifying the existing CC2420RadioM component's code.

5.2.2 Reliable Data Transfer

The second major component is the TransportC component. This component manages the reliable transfer of data between the node and the base station. The necessity of a transport protocol in this scenario arises because of the lossy nature of the wireless medium. Since the data is pertaining to a single measurement of the vibration signature of the motor, it is essential that the whole data should reach the base station in order to make any meaning out of the data. The protocol insists on various protocol specific message exchanges between the sender and the receiver to ensure the reliability of the data transfer. These messages are certainly an over head on the application, but are a necessary evil for its existence. Moreover the overhead when compared to the actual data being transfered is not very high. The transport protocol has a header message and a tail message as boundaries for a single file. Here file is referred to the entire data obtained in a single measurement. We treat this data as atomic and all the transfers are in terms of files. In our present setup we are measuring 4096 points of data and each point is a 16 bit ADC data, hence the total data size (file size)

is 8192 bytes. In TinyOS, the standard packet size is of 38 bytes, ten bytes are reserved for the MAC protocol header and rest 28 bytes are available as payload. We will be using this payload to transfer our measured data and other relevant information related to the data. With each packet of data sent we also send a two byte sequence number along with the packet. This sequence number later helps in accounting for the received data at the base station and recovery of lost packets. Since the payload is 28 bytes,and two bytes are allocated for sequence number, there would be 26 bytes left for the data transfer. With a payload of 26 bytes the total number of packets per file will be 316.

Once a node intends to send a data file to the base station it sends a header message giving information about it own id, and file number that will be transfered. On hearing a header message the base station would reply to the node with a header reply. On receipt of this reply, the node commences the data transfer. On completion of the data transfer the base station accounts for the number of packets arrived and sends an ACK message on successful reception of all the packets. If any of the packets are missing the base station would send an NACK message indicating the sequence number of missing packets. On receiving the NACK message the node would retrieve the missing packets from the RAM and retransmit them to the base station. This continues until the base station receives all the packets of a file. There is an upper limit to the retransmissions that can be done, after certain retransmissions the sender node will time out and send a tail message indicating that the file transfer is incomplete. The sequence of message exchange for a single file transfer is depicted in figure 5.6. Apart from sending the data to the base station the node also writes the data onto an external flash available on the mote.

The node acting as the base station would be drawing power supply from the computer it is connected to (unlike the client nodes that are running on battery power) hence it is not necessary for the base station node to go into sleep mode. Moreover since the activity of each node is random and is staggered over the entire day, it is necessary for the the base station to be awake all time. The code running on the base station would be different since it doesn't need the *RadioSleepTime.nc* interface. One more difference between the code in the base station and client is that the base station needs to use the UART interfaces to send the received data to the database via UART.

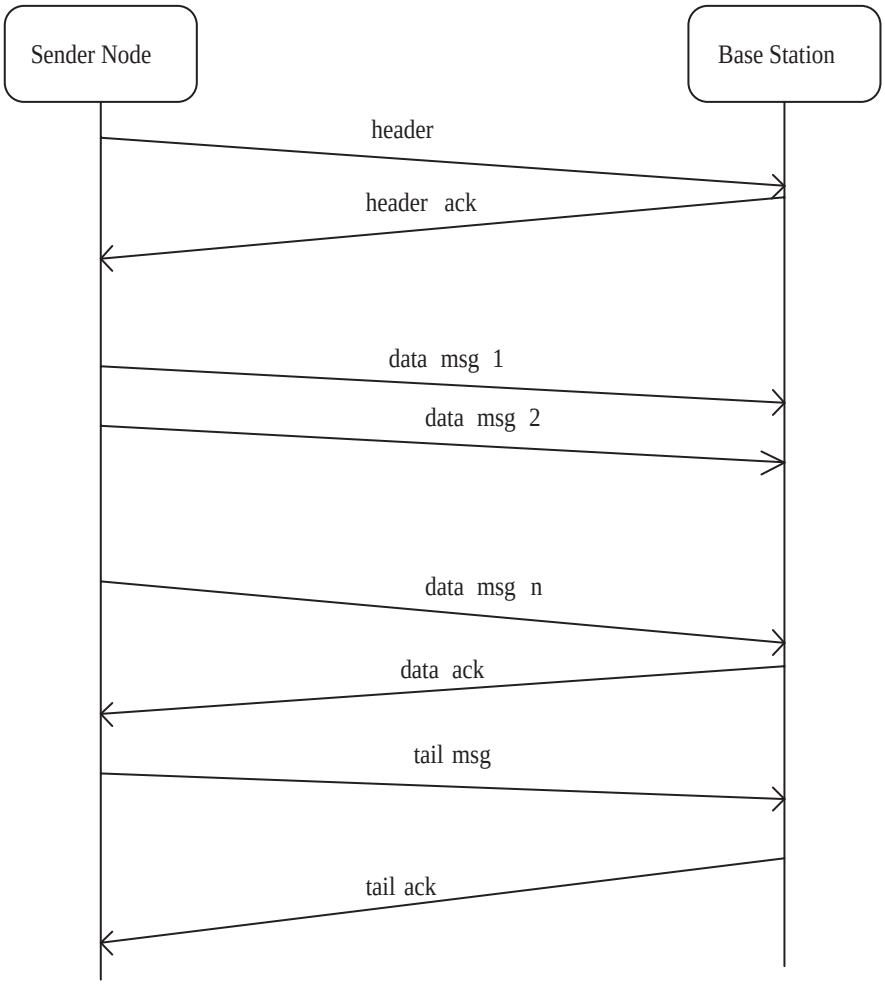

Figure 5.6: Sequence diagram of data flow between a node and the base station

5.2.3 Flash Storage

Flash management is slightly tricky in TinyOS. There are some prerequisites for the usage of the flash. The flash needs to be formatted and mounted before it is being used for writing data. There is an application called *Formatflash* in the sample applications of the TinyOS package. This application should be used to format the flash before loading the application code onto the Tmote. Also before writing to a location, the flash should be erased with the command *BlockWrite.erase()*. Once the *eraseDone()* event fires, the flash can be used for writing any new data. After writing the data onto a location the data should be committed to make it non volatile. *BlockWrite.commit()* is the command for doing this. The data written on to the flash could also be checked for cyclic redundancy check (CRC) errors with the help of *BlockRead.Crc()* command. If the CRC passes an event *BlockRead.CrcDone()* will be fired.

5.2.4 Data Measurement

The next part of the application is the data measurement. This is the most important part of the application. The signal to be measured is available at the ADC pin of the mote, it is a continuous analog signal. TinyOS uses ADC interface to do analog to digital conversion of the signal. Prior to using the ADC interface, preliminary information like which pin to be sampled, reference voltage etc are to be initialized. This initialization can be done by changing the parameters in the relevant ADC header files. There are two reference voltages available for the ADC conversion, 1.5V and 2.5V. Once this is done the the command *ADC.getData()* can be called. As soon as this command is called the microcontroller does an ADC conversion and returns a 16 bit digital equivalent value of the analog signal. Specific sampling rate can be achieved by employing timers that would fire at preset sampling rates, whenever the timer fires, a conversion is done. But this type of design may not suit all applications. In our application we need to sample the data at 20KHz, this demands the timer interface to fire once after ever 50 microseconds. Regular *Timer.nc* interface has a least count of one millisecond i.e the smallest interval that it can fire after is 1 millisecond. The configuration *TimerC* also has another interface named *TimerJiffy.nc* which is supposed to fire at intervals of jiffy (one jiffy = $1000000/32768 = 30.5us$ apprx). For example, to achieve a 3KHz sampling the timer event should tick at every $1000000/3000 = 333$ μs, hence $333/30.5 = 11$ jiffies approximately.

We have tried using the *timerjiffy.nc* interface for the ADC measurements, but we observed that though the timer is firing at the set intervals, the conversions are not being done at this rate, because of which we could not achieve the desired sampling rate. We couldn't figure out the actual problem of this. To overcome this problem we used the DMA(direct memory access) facility of MSP430 to achieve the necessary sampling rate. The interface DMA.nc is used in conjunction with the ADC interface. To use this facility we need to give DMA interface, a pointer to an empty buffer where the measured samples can be stored and set the necessary sampling rate in the ADC interface. Once the command *ADC.getdate()* is called DMA takes over the bus and transfers the samples to the specified location. As soon as the transfer is done, an event *DMATranferDone()* is fired, as this event fires, the values in the buffer need to be moved out so that the buffer is ready to store the next set of samples.

Chapter 6

Performance Analysis

This chapter consists of all the experimental results and analysis. Here we evaluate the performance of the hardware and software we have developed. We would also be substantiating our design decision of choosing a single hop network over a multihop network by experimental data. We have conducted experiments on vibration test bed to prove the hardware and we also conducted experiments in a indoor environment to evaluate the data transfer mechanism from the nodes to the base station. The rest of the chapter is organised as follows, the first section proves that single hop is better than multihop by analytical data, in the next section we would be showing the obtained data rates and the reliability of the data transfer. In the last section we show how accurate are the vibration measurements obtained from the testbed trials.

6.1 Singlehop Vs Multihop

Whilst developing the application we had to make a design decision whether to use singlehop of multihop. Traditionally multihop is considered to be more power efficient in sensor networks. Applications like [2], [8], [4] employ multihop routing protocols to transfer the data to the base station. But we argue that employing a singlehop or multihop routing protocol in a sensor network depends upon the underlying hardware used and the internode distances in the deployment. We substantiate our argument with necessary experiment and calcuations. Tmotes use CC2420 [22] radio to transmit and receive the data. This radio consumes 18.8mA current during reception and 17.4mA current while transmitting at full power. However there

TinyOS Power Value	Transmit Power(dBm)	Current Consumption(mA)
31	0	17.4
27	-1	16.5
23	-3	15.2
19	-5	13.9
15	-7	12.5
11	-10	11.2
7	-15	9.9
3	-25	8.5

Table 6.1: CC2420 Radio - Transmitted Power and Power Consumption [22]

is a facility in CC2420 to conserve the transmit current by reducing the radio transmit power.

In a recent work done by our colleagues [23], it has been proved that if the distance between the radio is less than 10 metres, the radio needs to transmit only at the minimum power(-25 dBm). This explains that the ranges achieved by CC2420 when transmitted at minimum power are around 10 mtrs. Taking this fact into consideration, we carried out an experiment in our laboratory 12mtrs by 12mtrs in size, to see what transmit powers would the radio require when the distances between each node and the base station is within 10 to 20 mts(with natural obstructions like walls and furniture). We have set up a base station and five client nodes that would be transmitting data to the base station.A schematic diagram of the set up is shown in 6.1.

The node inside the room (node 1) had only the obstruction of furniture, cubicles and computers. Node 2 was placed in the printer room, there is a cement wall in between the node and the base station, nodes 3, 4 and 5 are placed outside the lab. There is a brick wall with glass windows and furniture in between the nodes 3, 4, 5 and the base station. With this setup we have run a power negotiation algorithm developed by [23]. In this algorithm each node would ping the base station with max power first, if the get a reply from the base station they would reduce the power to half and then ping again, if they get a reply they reduce the power further. At any stage if they dont receive a reply they would increase the power and ping again. After few iterations they would finally settle at a minimum power that is enough to communicate with the base station. The base station in this case would

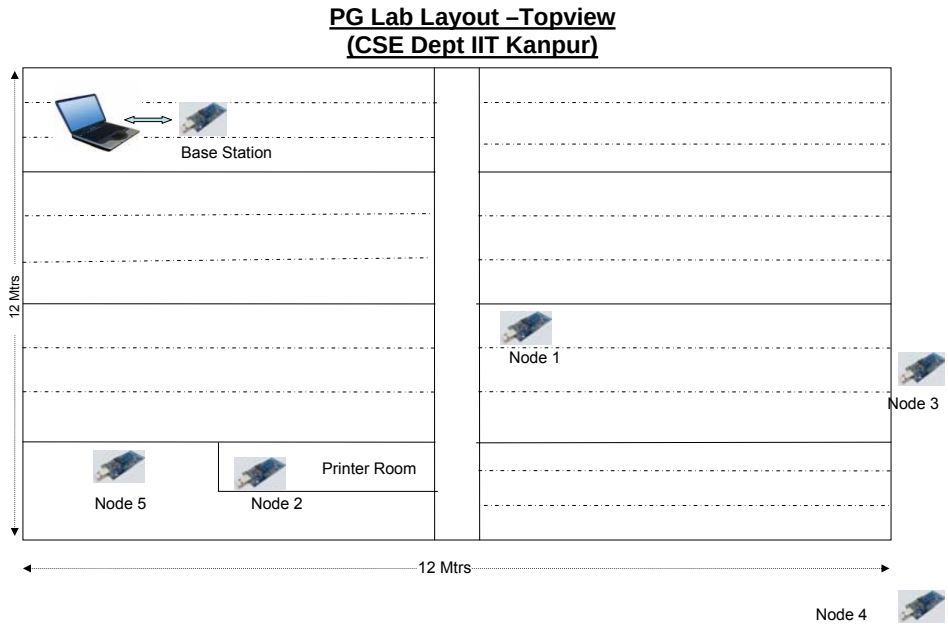

Figure 6.1: Experimental Setup

Node ID	Transmit Power(dBm)
1	-25
2	-9
3	-12
4	-18
5	0

Table 6.2: Transmit power of each node after running the power negotiation algorithm

transmit replies in full power all the time.

We have run this power negotiation algorithm on all the five nodes and the results are tabulated in 6.2. From the table we can infer that though the nodes are within a close range of the base station,they need different power level to reach the base station. With the same setup we have run the algorithm again between node 5 and node 2(acting as base station). Now we found that node 5 can reach node 2 with a minimum transmit power of -25dBm. Similarily we found that node 3 can reach node 1 with a minimum power of -25dBm. In a conventional setup one would decide to have a multihop setup in such a scenario and let node 5 transmit data to the base station via node 2 and node 3 transmit data to the base station via node 1. This also looks logical because normally if node 5 wants to transmit data to the base station directly, it would have to transmit at the maximum power, but if it transmits via node 2 it needs to transmit only at the lowest power(-25dBm). This would have been fine if the power consumed for reception is low. In our case CC2420 consumes 18.8mA current for reception. We have calculated the power consumptions for node 5 and node 2 in both cases and found that in case of a multihop transfer from node 5 to the base station via node 2, node 5 would save power by transmitting at minimum power but node 2 would consume 3 times more excessive power than if ndoe 5 would have done a direct transfer with maximum power to the base station. Since node 2 would receive all the packets from node 5 and transmit them to base station, it consumes a lot of power in receiving. So in such cases it is always better to have a single hop than multihop. The power calculation to substantiate our conclusion is presented in the following paragraph

Let us first consider the case of multihop, in our case, node 2 would be receiving packets from node 5 and will route these packets to the base station. Knowing that each file in our system consists of 316 packets and a header and a tail packet(if there are no packet losses). So let us first calculate the power consumed in recieving these packets.
Assume that the battery voltage is constant at 3V
The power consumed by a node can be calculated by the formula $(V \times I)\, watts/hour$
The time taken for receiving 318 packets at a data rate of 35kbps $= (318 \times 38 \times 8)\,/35000 \cong$ 2.76 seconds (35Kbps is the maximum throughput achieved in our experiments)
It needs the same amount of time to send the packets to the base station, that is 2.76 secs
So the total period of time the radio has to be ON is 2.76 + 2.76 = 5.52seconds
Current needed to transmit the packets to base station by node 2 is 11.2mA (power negotiated by node 2 is -9dBm)
So the total current consumption in transmission and reception is 18.8 + 11.2 = 30mA
Now the total power consumption by node 2 $= (30 \times 3 \times 5.52)\,/3600 = 0.138mW$
Suppose if node 5 sends them directly at 0 dBm it will spend only 17.4mA for a period of 2.76 seconds
So the power consumed will be $(17.4 \times 2.76 \times 3)\,/3600 = 0.0402mW$
This indicates that there is clearly a loss of *97.8 micro watts* by node 2 by doing a multihop transfer

In order to setup a power efficient single hop network, we need to run the power negotiation algorithm on all the nodes before they start their actual functionality. Though power negotiation places an overhead of upto 50 packets exchange between the node and the base station, it would be benefecial in the longrun to save the power. This algorithm needs to be run once after the initial deployment and once the optimum power is established,the node should settle at that power and continue all its transmissions with that power only. If the connectivity is lost by any unforeseen reasons, each node should maintain a time out after which they will run the power negotiation algorithm and negotiate transmit power with the base station once again.

Delay(msec)	Throughput(kbps)	%Packet Loss(per file)
10	17.5	4
5	23	6
2	32	8
1	35	9

Table 6.3: Throughput and Packet loss at various delay intervals

6.2 Data Transfer Analysis

To analyse the quality of data transfer between the nodes and the base station we have measured the latency,throughput and RSSI values for various number of files. We have set up the nodes as per Figure6.1 and we have connected a node running the *TOSBase* application on it to snoop all the packets in the air.We had to do some modifications in code of the application TOSBase to do these measurements. We have added the interface *LocalTime.nc* in the TOSBase application inorder to get the timestamp of the packet arrival. TOSBase snoops any packet in the air and send it via the UART to the PC. It uses a command UARTSend.send() for this purposes. TinyOS installation comes with a Java program that can dump all the packets received on the UART on to the screen. This data can be redirected to be saved on to a trace file. We have written a perl script to analyse the trace file. This perl script parses the tracefile and calculates the number of packets received per file per node. It also measures the latency per file from the timestamps in the packet payload. From this data it calculates the throughput per file. The throughput is calculated by the formula,

Throughput = $(NoofPacketsreceived) \times (Noofbitsreceivedperpacket)/(delayinsecs)$

We have conducted various experiments with time interval between at 10 milliseconds, 5 milliseconds, 2 milliseconds and 1 millisecond. The best throughput obtained was 35kbps in an indoors environment. The statistics at various delay intervals is tabulated in table 6.3, this is the MAC level throughput. Throughput observed in the experiments carried out by [19] is also around 27kbps(approximately). Though CC2420 claims to have a throughput of 256kbps the actual throughput obtained is not more than 30 to 40 kbps.

We are also measuring the RSSI(received signal strength value of the packet at the receiver). RSSI can be measured by reading the configuration register CC2420_RSSI or by

Figure 6.2: Test Bed for Trials (figure source www.spectraquest.com)

reading the strength field of each packet received.During our experiments we found that at times the throughput falls rapidly, we have taken several readings to find a correlation between the RSSI and throughput but couldnt understand the reason for throughput variation.This may probably be attributed to personnel movement in the vicinity, but we are not sure about this.

6.3 Vibration Measurements

We have carried out our experiments on a test bed that is used to simulate and analyse the vibration signatures of common machine faults. Figure 6.2 shows the testbed used for conducting the experiments. It is a machine fault simulator manufactured by Spectraquest Inc [17]. The simulator consists of a variable speed motor and an extended shaft assembly. Since the bearing details of the machine were not known we have conducted the experiments with the motor running at different speeds and measured the resonant vibration of the motor at that speed. If the motor was running at 25Hz, the body of the motor vibrates at 25Hz and this should be picked up by our accelerometer.

The output of the accelerometer is a 10Khz continuous analog signal. The amplitude of the signal varies with the measured acceleration. This analog signal would be first sampled at a predetermined sampling rate and converted into digital voltage samples with the help of an analog to digital converter also known as ADC.We have used the ADC that is inbuilt

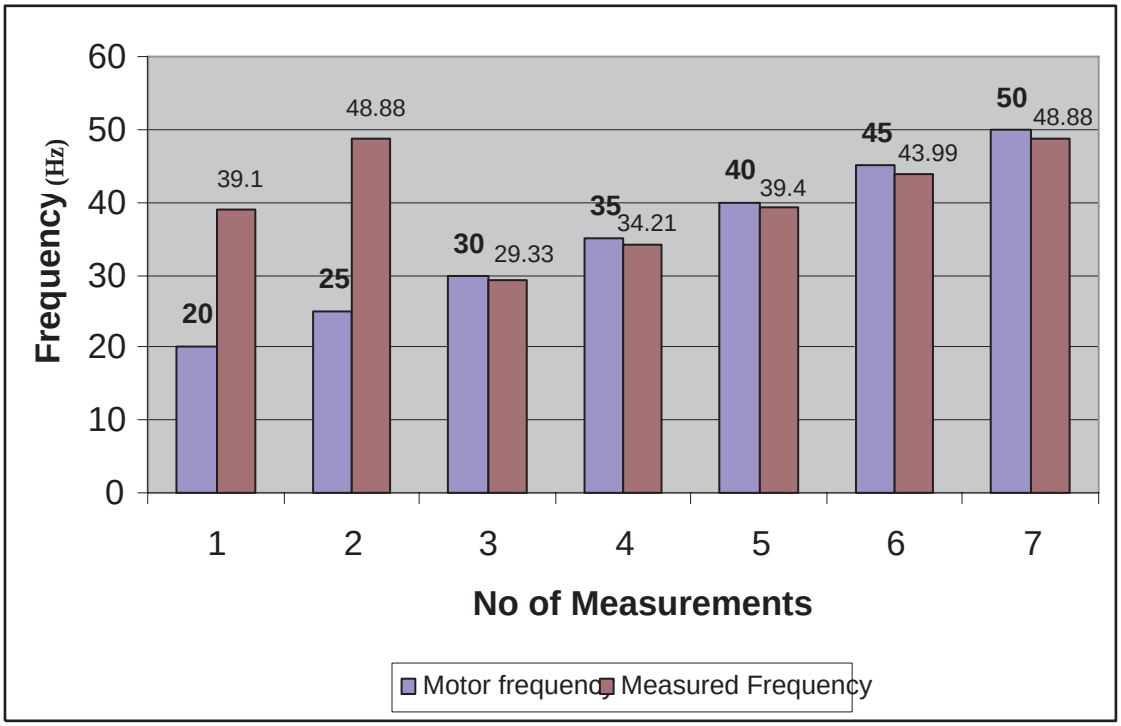

Figure 6.3: Vibration readings measured with ADXL105

in the microcontroller MSP430 of the Tmote. Inorder to faithfully reconstruct the signals and extract the necessary frquency components,we need to sample the signal at twice the frequency as per the Nyqyist criterion of sampling. Once the sampling is done these samples would be given as an input to the FFT algorithm. Fast Fourier Transforms [17] are used to convert time domain samples to frequency domain components, these frequency components are further plotted for vibration analysis. We have taken readings at seven different frequencies starting from 20Hz and increased gradually in steps of 5 upto 50Hz. We had to restrict at this speed because of the limitations of the simulator. We have obtained desired results on all the frequencies measured except for 20Hz and 25Hz. The first observable dominant component at these frequencies was the second harmonic second harmonic component i.e, we observed a dominant component at 39.1Hz when the motor was running at 20Hz and at 25 Hz the dominant component was observed at 48.88Hz.

Testing the application for bearing faults necessitates a test bed where the complete details

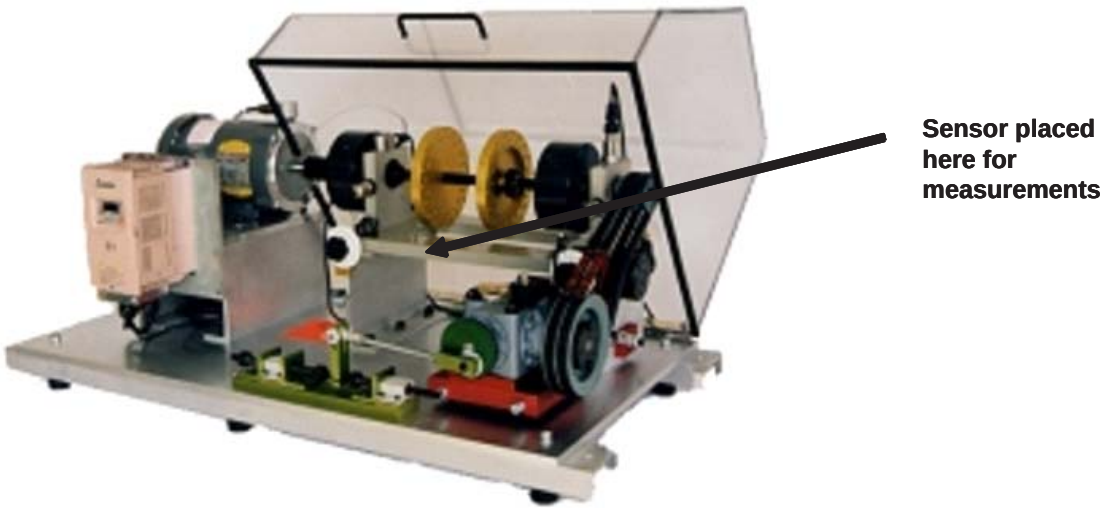

Figure 6.4: Position of sensor on testbed

of the bearing in a motor are known . If the details of the bearing mentioned in section 4.1 are known we can calculate the necessary frequencies that we should look for while taking the measurements(as per the equations mentioned in section 4.1). An exhaustive testing would involve mounting bearings with known defects, like hole in the outer cage or bearing with damaged balls etc. In our case we have a test bed that is being used for carrying out vibration analysis of motors that have dynamic load imbalances. Since the test bed was not exclusively available at our disposal we could not conduct extensive experiments. Instead we have tried to get a proof of concept of the whole system. We have run the motor at a known speed (the motor has variable speed control) and we measured the frequency at which the frame of the motor is vibrating. If there are no external forces acting on the motor the frame should vibrate at the frequency at which the motor is rotating.Since the test bed is made to precision, we could successfully measure the vibration with the help of our apparatus.We have also compared our measurements with the precision vibration sensors mounted on the testbed.We found that our sensor has displayed a near optimal performance in most of the measurements. Out of measurements taken at seven different frequencies we were able to measure the frequency correctly for five frequencies.During the experiments we have found

that ADXL105 was under performing at lower frequencies (below 25Hz). The reason for this under performance could not be localised, the probable reason could be that the vibrations generated at 25Hz may be very low and below the sensitivity of the accelerometer. This can be improved by increasing the gain of the internal amplifier of the accelerometer.

For the purpose of experiments we have sampled the signal at 20 KHz and obtained 4096 data points per measurements. This combination was chosen for the following reasons. The frequency resolution of the FFT would depend on the sampling rate and the number of data points measured. It is calculated by the formula

$\delta F = 1/(\delta T \times N)$

where δF is the desired frequency resolution

δT is the time between two samples(depends on the sampling rate)

N is the number of samples obtained

in our case $\delta F = 1/(50 \times 10^{-6} \times 4096) = 4.88Hz$

To obtain a frequency resolution of 5Hz this was the best combination. The frequency resolution can be further reduced by a different permutation of the sampling frequency and the number of measured data points. One more reason to sample the signal at 20Khz is to alleviate the possibility of having any aliasing affect on the measured signal.

For the purpose of measurement the accelerometer is placed on the frame of the shaft assembly and the output of the accelerometer is connected to the ADC pin on the expansion connector of the mote. A total of seven sets of readings were taken starting from a speed of 20Hz up to 50Hz in steps of 5Hz. Defective bearing frequencies generally lie in between 50Hz and 200Hz [8], from the above measurements we can assert that the sensor would affectively pick up the bearing frequency components too.

6.4 Network Lifetime

Sensor network applications are commonly battery driven, hence energy consumption plays a critical role in determining the lifetime of any sensor network installation. Also these applications are expected to run uninterrupted for long durations(few months), as changing the batteries frequently may disturb the environment in some cases [4] and in few other cases

	Current Consumption	**Power consumption per day**
Microcontroller (A)	1800μA	= $1800\times10^{-6}\times24$ = 0.0432 AH
Radio (B)	17.4mA	= $17.4\times10^{-3}\times(2.76/3600)$ = 13.34×10^{-5} AH
Flash (C) (Erase + Write)	20mA + 20mA	= $(40\times10^{-3}\times44\times10^{-3})/3600$ = 4.8×10^{-7} AH
ADXL105 (D)	2mA	= $(2\times10^{-3}\times1.2)/3600$ = 6.67×10^{-7} AH
Total power consumption per day (A + B + C + D)		= 0.0433AH
Battery capacity		= 2.2AH
NODE LIFE		= (2.2/0.0433) **= 50.8 days (apprx)**

Figure 6.5: Network Lifetime

where the nodes are randomly deployed with the help of helicopters etc it may be difficult to change the batteries frequently. Hence it is essential to have an accurate estimate of the lifetime of the nodes inorder to optimize the battery lifetime. Accurate estimation of power consumption can be achieved by measuring the current consumed by various modules of the application using precision instruments. We found that these type of measurements have been done by [27] but they are cumbersome and beyond the scope of our work. Instead we have estimated the power consumption of each node by taking into consideration the current consumption of various components of the hardware. All the values have been obtained from the respective datasheets.

Tmotes run on a pair of AA batteries, making a conservative estimation that the batteries would be able to supply 2200mAh current at 3 volts [4] we have made our calculations and arrived at the figure shown as in figure 6.5. From the figure 6.5 one can makeout that the major power consumer in our application is the microcontroller(as it is not duty cycled), so we need to put the microcontroller power down state when not in use. We would be hansling this issue in our future deployments.

Chapter 7

Future Work and Conclusion

7.1 Future Work

Our work is aimed at showing a proof of concept. In this work of ours we tried to show that by leveraging various technologies we can create useful real world applications. Our work in its present state has been proved only in a laboratory environment. It needs to undergo arduous trials before it could emerge as a usable product. We still need to do the following to optimise the application interms of performance and robustness.

Data Compression In the present form the application is handling large chunks of measurement data(in the tune of kilobytes). Handling large chunks of data not only hogs the constrained resources like RAM, but also results in lots of power consumption while transferring it to the base station. This can be over come by introducing some intelligent processing on the node itself. In our case we are collecting the raw data and transferring it to the base station. At the base station the data is passed through an FFT algorithm in a desktop computer. If we can push the FFT computation part on to the node we can reduce the amount of data being transferred from the node to the base station from few kilobytes to few bytes.Once if FFT of the time domain samples are calculated on the node we need to transfer only the relevant frequency components to the base station for analysis. Calculating FFT would involve N(LogN) multiplications [17] where N is the number of frequency components we want. So in our case where we chose to get a 4096 point FFT, the number of multiplications required to be done will be quite large(49000 approximately). This requires a large amount of processing power but the advantage of using *Tmote* is that it has an inbuilt hard-

ware multiplier [20]. Compute intensive operations like FFT calculation etc can be forced to the hardware multiplier with a compiler option in gcc.

There is another technique to compress the measured data. The analog to digital converter module in MSP430 supports only 12 bit ADC conversions, when we use the *ADC.getdata()* command in tinyOS it returns a 16bit integer value in which only the LSB 12 bits are of relevance.So by some intelligent methods we can use the 4 bits that are free in the 16 bit integer to compress the data. With a slight over head of compression and decompression we can achieve a compression of 1/3, of the measured data straightway

Packaging The accelerometer and the mote in their present form cannot be deployed in the industry as they do not have a protective covering. We need to manufacture a housing for the accelerometer and the mote. We also need to reduce the amount of wiring in between the mote and the accelerometer. It should be confined to a single weather proof unit, so that the deployment would be easier.

Extensive Industry Trials With the number of measurements done and the testbed used we cannot assertively tell that the system is fully operational and it can be readily deployed. We need to carry out tests on different sizes of motors with known defects in the bearings so that we can tabularise the vibrations with the bearing defects and make the whole process of defect diagnosis aotomatic. We also need to check the robustness of the code and prove all the corner cases.

Site Survey Prior to installation of the sensor nodes in a shipboard engine room we need to conduct a site survey to address the issues of safety and interference. In a ship safety of personnel and the equipment is of foremost importance.We need to check that the transmission doesn't interfere with the safe operation of the existing equipment. A survey of the RF coverage also is required to be done before actual deployment. This identify shadow zones caused by the existing equipment.[21] have studied extensively about the radiowave propagation in industrial environments.[8] have also done a similar site survey before the actual deployment of their sensor network in a ship.[8] have found during their that the connectivity between the sensor nodes is excellent.This high performance of the radio frequency transmissions were attributed to the steel material at the site. Unlike in a normal deployment where the walls are of cement, shipboard rooms have walls made of steel. Cement absorps the RF energy

where as steel reflects it, this could be the probable cause of the high performance.

User Friendly Frontends Since the end user of this system are unskilled/skilled technicians who may not have an in-depth knowledge of the software/hardware used in the system, there is a necessity of developing a user friendly frontend software for the system.

Security Necessity of security is not felt as of now,because the ships are generally employed with perimeter(physical) security. In certain deployments, this aspect may need further consideration.

Low power operation of Mote In the present design we are switching off only the radio component. But the microcontroller and all other auxillary hardware components on the mote would still be consuming power unnecessarily. The mote needs to be put to a low power operation by switching off all the unnecessary components when they are not in use. This would help in conserving the battery power a lot.

7.2 Conclusion

Marine shipboard engine rooms are usually harsh, with hazards ranging from strong mechanical vibrations, high temperatures, noisy electrical equipments. It is expensive and difficult to install wiring in these environments, which range from oil tanks, water tanks,lubrication oil tanks to main engines,generators and compressors etc. Thus these environments are suitable targets for low power wireless sensor networks.We have developed a cheap and reliable system that can be used to measure bearing vibrations on motors and send them wirelessly over a radio channel to a central base station where this data can be analysed. This type of system would replace the present existing both cumbersome manual procedure of measuring the bearing vibrations and expensive online systems.

This technology can further be extended to measure vibrations of other equipment like engines, generators and shaft journal bearings too. By using suitable sensors the system can be used effectively to automate the existing predictive maintenance procedure that is being followed from ages.

Bibliography

[1] David Gay, Phil Levis, Eric Brewer,David Culler,Matt Welsh,"The nesC Language: A holistic approach to Network Embedded System" *PLDI03,SanDiego California(Jun,2003).*

[2] Ning Xu, Sumit Rangwala, Krishna Kant Chintalapudi, Deepak Ganesan, Alan Broad, Ramesh Govindan, Deborah Estirin, *" A Wireless Sensor Network for Structural Monitoring" SenSys'04 (November 3-5, 2004)*

[3] Tuomo Lindh, Jero ahola, Jarmo Partanen, "An Evaluation of micromachined accelerometer in the Condition Monitoring of Induction Motor Bearings", *VANEM 2002 3rd International Seminar on Vibrations and Acoustic Noise of Electric Machinery, Lodz, Poland, 17-19 October 2002.*

[4] Alan Mainwaring, Joe Poalstre, Robert Szewczyk, David Culler, John Anderson. " Wireless Sensor Networks for Habitat Monitoring", *WSNA 02, September 28,2002, USA*

[5] Sensor Networks Magazine. Condition Based Monitoring `http://www.sensorsmag.com/articles/0602/14/main.html`

[6] Victor Shnayder, Bor-rong Chen, Konrad Lorincz, Thaddeus R.F. Fulford-Jones and Matt Welsh,"Sensor Networks for Medical Care", *Technical Report TR-08-05, Division of Engineering and Applied Sciences, Harvard University,2005*

[7] Pei Zhang, Christopher M.Sadler, Stephen A.Lyon and Margaret Martonosi, "Hardware Design Experiences in ZebraNet" *SenSys'04, Baltimore USA*

[8] Lakshman Krishnamurthy, Robert Adler, Phil Buonadonna, Jasmeet Chhabra, Mick Flanigan, Nandakishore Kushalnagar, Lama Nachman, Mark Yarvis, "Design and De-

ployment of Industrial Sensor Networks:Experiences from a Semiconductor Plant and the North Sea" *SenSys'05, November 2-4,2005, San Diego.*

[9] Shock and Vibration Handbook by Cyril.M.Harris

[10] Schiltz, R. L., *Forcing frequency identification of rolling element bearings, Sound and vibration, pp. 16-19, May 1990.*

[11] `http://www.xbow.com/Products/Product_pdf_files/Accel_pdf/LP_Series_Datasheet.pdf`

[12] `http://www.ecs.soton.ac.uk/~mk1/2prin.pdf`

[13] `http://retina.et.tudelft.nl/data/artwork/publication/mems/ectm039.pdf`

[14] `http://www.moteiv.com`

[15] Datasheet ADXL105

[16] `http://www.spectraquest.com/products/whymfsl.html`

[17] `http://en.wikipedia.org/wiki/Fast\_Fourier\_transform`

[18] `http://www.dliengineering.com/vibman.htm`

[19] Design and Implementation of Bridge Monitoring Application,MTech thesis report H.Hemanth,May 2006, Indian Institute of Technology, Kanpur.

[20] MSP430 microcontroller datasheet.Available for download at Texas Instruments' website.

[21] kjesbu,S and Brunsvik,t.Radiowave propagation in industrial environments.*In proceedings of the IEEE internatinal conference on Industrial Electronics,Control and Instrumentation(IECON 2000)(Nagoya,Japan,october 2000)*

[22] Datasheet of CC2420 radio module.Available for download at`www.chipcon.com/files/CC2420\_Data\_Sheet\_1\_3.pdf`

[23] DY Gokhale and M Naveen.Topology Control in Linear Wireless Sesnsor Networks.Term project report for CS725(Topics in Networking) May2006, Indian Institute of Technology Kanpur.

[24] Chien-Yih Wan,Andrew T.Campbell and Lakshman Krishnamurthy.PSFQ:a reliable transport protocol for wireless sensor networks.In WSNA'02:*Proceedings of the first ACM internantional workshop on Wireless sensor networks and applications,pages 1-11, NewYork, USA.2002.*ACM Press

[25] J Polastre,Jason Hill,David Gay.Versatile lowpower media access for wireless sensor networks.*SenSys04, November 35, 2004, Baltimore, Maryland, USA.2004 ACM 1581138792/04/0011*

[26] `Science.howstuffworks.com`

[27] Olaf Landsiedel, Klaus Wehrle, Stefan Gtz: Accurate Prediction of Power Consumption in Sensor Networks, In *Proceedings of The Second IEEE Workshop on Embedded Networked Sensors (EmNetS-II),* ISBN 0-7803-9246-9, Sydney, Australia, May 2005