Enhanced Localization Using Trilateration Technique for Wireless Sensor Networks

Cao Ning

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Computer Engineering
Prince of Songkla University
2010

Thesis Title Enhanced Localization Using Trilateration Technique for Wireless Sensor Networks
Author Mr. Cao Ning
Major Program Computer Engineering

Major Advisor:

....
(Asst. Prof. Dr. Wannarat Suntiamorntut)

Examining Committee:

N. Jindapetch ....Chairperson
(Asst. Prof. Dr. Nattha Jindapetch)

....
(Asst. Prof. Dr. Wannarat Suntiamorntut)

Kamol Kaemarungsi ....
(Dr. Kamol Kaemarungsi)

The Graduate School, Prince of Songkla University, has approved this thesis as partial fulfillment of the requirements for the Master of Engineering Degree in Computer Engineering.

....
(Assoc. Prof. Dr. Krerkchai Thongnoo)
Dean of Graduate School

Thesis Title	Enhanced Localization Using Trilateration Technique for Wireless Sensor Networks
Author	Mr. Cao Ning
Major Program	Computer Engineering
Academic Year	2010

ABSTRACT

The concept of the challenge and importance of localization for wireless sensor networks (WSNs) is worldwide accepted, especially if cost-effective approaches are demanded, because any information of the sensor nodes would be useless without the wanted position for most of the applications.

In this thesis, we present an enhanced trilateration algorithm, which is a range-based localization method employing the received signal strength indicator (RSSI). The reason for choosing RSSI is that no extra hardware is needed for network-centric localization. Firstly, we introduce the corresponding concepts of wireless sensor networks and localization techniques, and present the current condition of study in these fields. Secondly, we discuss the relationship between RSSI and source-receiver distance and introduce a frequently-used mathematical model to express that relationship. In addition, we intensively analyze two factors which affect the measured RSSI values: non-isotropic path losses, and antenna orientation. Finally, we present an enhanced trilateration localization technique, which is expected to perform better than the traditional trilateration algorithm. A type of modified Tmote Sky node named Unode is chosen as experiment tools for all the tests. Then we evaluate the enhanced trilateration technique by comparing its results with the traditional trilateration through real experimental data.

Keywords: Wireless Sensor Networks, RSSI, Distance, Localization, Trilateration

ACKNOWLEDGEMENT

The thesis, together with the 3-year study here is coming to an end. At this moment, I would like to extend my sincere thanks to all those who have concerned about or helped the writer.

My deepest gratitude goes first and foremost to Dr. Wannarat Suntiamorntut, my supervisor, for her instructive advice and useful suggestions on my thesis. She has walked me through all the stages of my work. Her understanding, patience, encouragements have been the most important factors which increase my confidence extremely.

Second, I would like to express my heartfelt gratitude to all the lecturers in Department of Computer Engineering, who helped me a lot on my study. And additionally, I would like to thank the graduate School, the staffs of faculty of Engineering and the staffs of Department of Computer Engineering to give the convenience for me.

Third, I would like to thank my friends in the lab of WSN for their supports and assistances.

Finally, my thanks would go to my beloved family for their loving considerations and great confidence in me all through these years. I also owe my sincere gratitude to my friends who gave me their help and time in listening to me and helping me work out my problems during the work of the thesis.

Cao Ning

CONTENTS

Page

CONTENTS......v

LIST OF TABLES......viii

LIST OF FIGURES......ix

LIST OF ABBREVIATIONS AND SYMBOLS......xi

CHAPTER

1. INTRODUCTION......1

1.1 Motivation......1

1.2 Objective......3

1.3 Scope of work......3

1.4 Work Plan......4

1.5 Outline......4

2. RESEARCH BACKGROUND......6

2.1 Introduction of Wireless Sensor Networks......6

  2.1.1 WSN Applications......6

  2.1.2 WSN Characteristics......8

  2.1.3 WSN Architecture......10

  2.1.4 Power and Network Standards of WSN......10

  2.1.5 Processor Trends......11

2.2 Localization of WSN Systems......12

  2.2.1 Introduction of Location......12

  2.2.2 Location in Physical World......12

  2.2.3 Classification of Localization Techniques......13

  2.2.4 Introduction of Range-Based Localization Techniques......15

  2.2.5 Introduction of Range-Free Localization Techniques......17

  2.2.6 Introduction of Classic Distance Acquisition Techniques......20

  2.2.7 Localization Technique Performance Indexes......21

2.3 Summary......22

CONTENTS(CONTINUED)

Page

3. SIGNAL PROPAGATION 24

3.1 **Relationship between RSSI and Input Power** 24

3.2 **Translation from RSSI to Distance** 26

3.3 **Analysis of Radio Irregularity** 28

3.3.1 Non-Isotropic Path Losses 28

3.3.2 Antenna orientation 30

3.4 Experiment Conventions 31

3.5 Summary 32

4. ENHANCED TRILATERATION ALGORITHM AND TESTING 33

4.1 Enhanced Trilateration Algorithm 33

4.1.1 Review of Traditional Trilateration Algorithm 33

4.1.2 Introduction to the Enhanced Trilateration 36

4.2 Experiment Environment 40

4.2.1 Experiment Model 40

4.2.2 Experiment Procedure 40

4.3 Setup and Configuration 41

4.3.1 Experiment Setup 42

4.3.2 Software Configuration 44

4.4 Testing, Simulating and Result 44

4.5 Comparison with Other Localization Algorithms 47

4.6 Summary 48

5. CONCLUSION AND DISCUSSION 50

5.1 Conclusion 50

5.1.1 Traditional Trilateration Algorithm 50

5.1.2 Enhanced Trilateration Algorithm 50

5.2 Discussion 51

5.2.1 Advantages 51

CONTENTS(CONTINUED)

Page

5.2.2 Limitations 51

5.3 Future work......52

REFERENCES 53

APPENDIX A. HARDWARE AND SOFTWARE 59

A.1 Introduction of Tmote Sky......59

A.1.1 Key Features......59

A.1.2 Module Description......60

A.1.2.1 Power 61

A.1.2.2 Block Diagram 62

A.1.2.3 Microprocessor......62

A.1.2.4 Radio 62

A.1.2.5 Antenna 63

A.1.2.6 External Flash 64

A.1.2.7 Sensors 64

A.2 TinyOS: Software Architecture of Wireless Sensor Networks......64

A.2.1 TinyOS Execution Model 66

A.2.1.1 Event Based Programming......66

A.2.1.2 Task......66

A.2.1.3 Atomicity 67

A.2.2 TinyOS Component Model......67

A.3 Network Embedded Systems C Language......70

VITAE 73

LIST OF TABLES

Table Page

2.1 Standards Comparison between ZigBee/IEEE 802.15.4 and Other Common Protocols..........11

2.2 Comparison of Existing Work..........19

3.1 Output Power Configuration for the CC2420..........24

4.1 Comparison of Existing work with Our Algorithm..........48

5.1 Comparison of Localization Results of Traditional and Enhanced Trilateration Algorithm....51

LIST OF FIGURES

Figure Page

2.1 A Typical WSN System 6

2.2 WSN Application Areas 7

2.3 Typical Sensor Node's Structure 9

2.4 Trilateration 16

2.5 Triangulation 17

2.6 Multilateration 17

2.7 Example of DV-Hop Algorithm 18

3.1 Typical RSSI Value vs. Input Power 25

3.2 Example of RSSI Collection 25

3.3 Experiment Setup of RSSI Collection 26

3.4 Radio Signal Propagation 27

3.5 Signal Strength over Time in Four Directions 29

3.6 RSSI Communication Range 29

3.7 Setup of Experiment of Antenna orientation 30

3.8 Radio Signal Propagation with Different Antenna Orientation 31

4.1 Ideal Situation of Trilateration Method 34

4.2 Real Situations with Errors of Trilateration Methods 35

4.3 A Complicated Case of Trilateration Method 36

4.4 Geometrical Character of Linear Lines and Circles 37

4.5 Situations with no Intersection between Circles 39

4.6 Experiment Model 40

4.7 General Flow Process Diagram of Experiment Procedure 41

4.8 Setup of Experiment of the Enhanced Trilateration Algorithm 42

4.9 Mathematical Model of Experiment of the Enhanced Trilateration Algorithm 43

4.10 Comparison between 4 Anchor Nodes and 5 Anchor Nodes with Path Loss n=2.39 45

4.11 Comparison between 4 Anchor Nodes and 5 Anchor Nodes with Path Loss n=1.89 46

4.12 Comparison between 4 Anchor Nodes and 5 Anchor Nodes with Path Loss n=2.89 46

LIST OF FIGURES (CONTINUED)

Figure Page

4.13 Errors of Different Path Loss with 4 Anchor Nodes....47

4.14 Errors of Different Path Loss with 5 Anchor Nodes....48

A.1 Unode: a Modified Tmote Sky Node....60

A.2 Functional Block Diagram of the Tmote Sky Module, its Components, and Buses....61

A.3 The Antenna We Used on a Unode Mote....63

A.4 External serial flash schematic....64

A.5 Voltage Monitoring for Tmote Sky Modules....65

A.6 AM Messaging Component Graphical Representation....69

A.7 NesC Application Configuration File that Wires together the Blink Application....70

A.8 NesC component File Example....71

LIST OF ABBREVIATIONS AND SYMBOLS

ADC	Analog-to-Digital Converter
AN	Anchor Node
AOA	Angle of Arrival
ASIC	Application-Specific Integrated Circuit
BN	Beacon Node
CEP	Circular Error Probability
CRLB	Cramér–Rao Lower Bound
DAC	Digital-to-Analog Converter
DMA	Direct Memory Access
DV-Hop	Distance Vector-Hop
FCC	Federal Communications Commission
GDOP	Geometric Dilution of Precision
GPS	Global Position System
I^2C	Inter-Integrated Circuit
LQI	Link Quality Indication
MDS-MAP	Multidimensional Scaling MAP
MEMS	Micro Electro Mechanism System
MN	Mesh Networking
MSE	Mean Square Error
NesC	Network embedded systems C
OS	Operating System
RAM	Random Access Memory
RF	Radio Frequency
RMSE	Root Mean Square Error
RoHS	Restriction of Hazardous Substances
RSS	Received Signal Strength
RSSI	Received Signal Strength Indicator

LIST OF ABBREVIATIONS AND SYMBOLS (CONTINUED)

SMA	SubMiniature version A
SPI	Serial Peripheral Interface
SVS	Supply Voltage Supervisor
TDOA	Time Difference Of Arrival
TOA	Time Of Arrival
UART	Universal Asynchronous Receiver/Transmitter
UN	Unknown Node
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Network

CHAPTER 1

INTRODUCTION

1.1 Motivation

It is commonly believed that the developments of micro electro mechanism system (MEMS), wireless communications and digital electronic technology are the foundation of wireless sensor networks (WSNs). A wireless sensor network is a kind of special Ad-hoc networks, which means the distributed nodes communicate wirelessly to a central gateway, which provides a connection to the wired world where you can collect, process, analyze, and present your measurement data.

Wireless sensor networks will surely change the interaction between us human beings and the objective world through deploying sensor nodes to target area. But for most of the WSN applications, the measurements of the deployed sensor nodes would be meaningless without the nodes' location information. Only if the positions are known, sensor nodes can tell us what position or area the special event happens, which is the basis of the external object localization. At the same time, the network designer can optimize the WSN application in other fields by observing the position information of the sensor nodes.

However, manual deployment or installation of global position system (GPS) receiver for every sensor node is restrained from several factors, such as cost, power consumption, and expansibility. And in some circumstances, such sort of deployment methods becomes infeasible and impossible, for instance, the localization result will not be ideal for the signal reflection, radio scattering brought by objects from indoor condition. Thus, some kind of special localization algorithm is needed for the location computation of the WSN system according to different environment conditions.

Since active badge (the first indoor-localization system in the world) was developed by AT&T Laboratories Cambridge in the year 1992 [34], researchers have been focusing on this field for nearly twenty years. In fact, there are plenty of systems and algorithms that can solve the problem of self-localization of wireless sensor networks. But each of these systems or algorithms has its own weakness. They are designed to solve different problems or support specific applications, since different surrounding factors will be encountered for each one of these localization systems or algorithms, for example, physical phenomenon, network buildup, energy consumption, infrastructure construction, and the complexity of time and space. In sensor networks, the energy of every sensor node is limited, the reliability of the system is not stable and acceptable sometimes, the sensor nodes may be deployed in large scale and distributed randomly, the communication range between wireless nodes is restricted. So, the development of localization algorithm of wireless sensor networks is very challenging due to those unfavourable factors.

After the various developments have been proposed in nearly twenty years, the localization systems of wireless sensor networks can be mainly divided into two types, range-based localization and range-free localization (it will be discussed in detail in the next chapter) [4]. In this thesis, we use trilateration, which is one of the most traditional and classic range-based localization algorithms, as the basis of our enhanced algorithm, because of the simplicity of the trilateration algorithm itself. Some of the key characters of the WSN systems should be considered to develop the new trilateration algorithm:

1) All the sensor nodes are distributed randomly, the location of these sensor nodes should be computed through communications between sensor nodes and anchor nodes (one of those anchor nodes is chosen as base station), whose position are set manually beforehand.

2) Measurement errors will be easily encountered because the hardware configuration of the wireless sensor nodes is low, the power storage is limited and the communication between nodes is unstable. So the enhanced algorithm should decrease the influence of measurement error effectively.

3) Try to decrease the complexity of the algorithm of the localization computation to save nodes' energy, so as to extend the wireless network's life cycle.

However, since the distance measurement between two nodes is one of the key issues as long as we take one range-based localization algorithm, we ought to consider the method to translate transmitting signal to distance between nodes. Among those distance computation methods, we choose to use received signal strength indicator (RSSI) as the tool to calculate the distances between nodes. The reason for choosing RSSI is that no extra hardware is needed for network-centric localization.

In this thesis, we firstly test the signal propagation in different surrounding environments to confirm the mathematical model that expresses the relationship between the received signal strength (RSS) and the distance estimation, and discuss a couple of factors that can bring influences to the signal propagation in space. Secondly we introduce an enhanced trilateration localization algorithm which is based on the classic trilateration, then experiment to test the localization algorithm are taken, later a comparison is given to show the differences between the old trilateration algorithm and the enhanced one. Finally, we summarize the thesis based on the data that we collected according to the experiment and provide a conclusion about the research work.

1.2 Objectives

1) To determine parameters of the mathematical model that reflects the relationship between the signal strength and the distance.

2) To test the signal-distance model with sensor nodes in real experiment situations with different surrounding environments and to discuss the factors that can affect the signal propagation.

3) To compute the locations of the deployed sensor nodes by using the traditional trilateration algorithm and the newly enhanced one in this work.

4) To evaluate the enhanced trilateration algorithm and make a conclusion based on the collected data.

1.3 Scope of work

1) To use In-house sensor nodes as the experiment platform.

2) To analyze the shortcomings of the traditional trilateration localization algorithm and the advantages of the enhanced one.

3) To compare the experiments data collected from the tests by using the traditional trilateration algorithm and the enhanced one.

1.4 Work Plan

1) To investigate and research related references to obtain the basic idea of wireless sensor networks and the traditional localization mechanisms of location and tracking systems.

2) To find out a better method for computing the position of the wanted sensor node by using trilateration.

3) Proposal writing.

4) To study the NesC language programming.

5) To build the TinyOS application for localization in a linux-like operating system.

6) To verify the transmitting signal propagation feature in real experimental situations.

7) To set up the project's experimental environment and collect data measured by using both the classic trilateration algorithm and the enhanced one.

8) To analyze implementation data and make conclusions.

9) To collect the results and write the final report.

1.5 Outline

This document is organized into 5 chapters as follows:

Chapter 1, Introduction. In this chapter, the motivation, objective and scope of this thesis are presented. It presents the work plan to evaluate and investigate an enhanced trilateration localization algorithm as well.

Chapter 2, Research Background. It presents the concept, the development, and the characteristics of wireless sensor networks. In addition, the information about the technique of localization in WSN systems is discussed.

Chapter 3, Signal Propagation. In this chapter, we introduce the mathematical model which reflects the signal propagation in space. Furthermore, we intensively discuss several factors that can affect the signal propagation based on experiments. And some improvements are given in real situations to decrease the influences brought by surrounding environment.

Chapter 4, Enhanced Trilateration Algorithm and Testing. An enhanced trilateration localization algorithm is introduced firstly. Then we present related experiments to prove the advantages by using the enhanced trilateration algorithm.

Chapter 5, Conclusion and Discussion. It analyzes the collected data from experiments. In this chapter, we also present some advantages and limitations of the enhanced trilateration algorithm and suggest required future work in this area.

Chapter 3, and Chapter 4 represent the primary work of this thesis and Chapter 5 is the conclusion.

CHAPTER 2

RESEARCH BACKGROUND

2.1 Introduction of Wireless Sensor Networks

Wireless sensor networks (WSNs) are a new class of distributed systems that are an integral part of the physical space they inhabit. Unlike most computers, which work primarily with data created by humans, sensor networks reason about the state of the world that embodies them [35]. Figure 2.1 shows a typical WSN architecture.

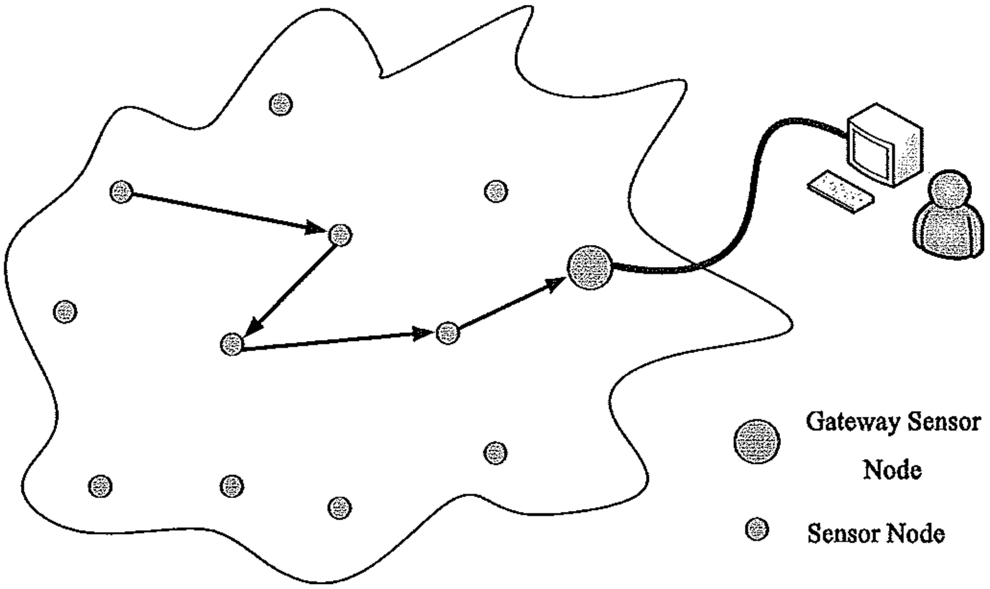
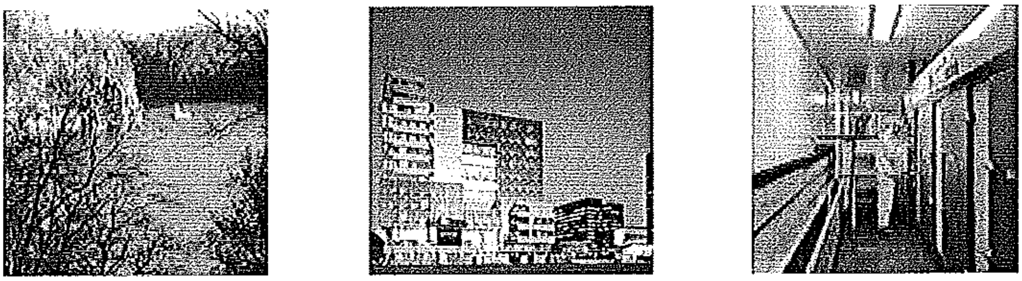


Figure 2.1 A Typical WSN System

2.1.1 WSN Applications

Combining with routers and a gateway, these autonomous devices, or nodes, are used to create a typical WSN system. The distributed measurement nodes communicate to a central gateway wirelessly, which provides a connection to the wired world. Routers can be used to extend distance and reliability in a wireless sensor network. So, an additional communication link between end nodes and the gateway can be gained.

Considering the features of energy-saving and flexibility of the distributed nodes, embedded monitoring system covers a large range of application areas, including those in which power or infrastructure limitations make a wired solution costly, challenging, or even impossible. Wireless sensor networks can be positioned alongside wired systems to create an integrated wired and wireless measurement and control system.



Environmental Monitoring **Resource Monitoring** **Industrial Monitoring**

Water/Soil **Indoor** **Power Monitoring** **Solar Monitoring** **Wind Farm Monitoring**

Figure 2.2 WSN Application Areas

The development of wireless sensor networks was originally motivated by military applications such as battlefield surveillance [36]. However, a wireless sensor network system is ideally suitable for an application like environmental monitoring since the requirements mandate a long-term deployed solution to acquire water, soil, or climate measurements. In a deployed WSN system, such as the utilities of electricity grid, streetlights, and water municipals, wireless sensors can offer a lower-cost method for collecting system information to reduce energy usage and better managements. In other fields, for example, office buildings, hospitals, airports, factories, power plants, and production facilities, related information can be kept monitoring continually by deploying these systems. Figure 2.2 is a presentation of the application areas of WSN systems.

2.1.2 WSN Characteristics

The unique characteristics of a WSN include [2] [3] [4] [5]:

1) **Limited hardware resources**. Due to the restrictions of price, bulk, energy consumption, interfaces and components of sensor nodes, the operation and computation system of the WSN system should not be too complicated.

2) **Limited power** they can harvest or store. Mostly powered by batteries, the energy of each sensor node is very limited. In some extreme conditions, it is not possible to recharge or change batteries to the sensor nodes. Thus, the sensor node will be useless (dead) in the network without any power. Energy conservation should be the precondition of any design of technique and protocol in WSN systems.

3) **Ad-hoc network**. Generally speaking, there is no such strict control center, and all the nodes are identical. A node can join or leave the system freely. The failure of one sensor node will be no influence to the whole system.

4) **Spontaneous organization**. The deployment of a WSN system may be evolved without any pre-set network facilities. Activities of sensor nodes can be coordinated by layer protocol and distributed computation. Sensor nodes can organize an independent network as soon as they are starting up.

5) **Dynamic network topology**. Wireless sensor networks are dynamic, sensor nodes can move anywhere, in or out to the existing network. A node can be out from the system operation because of the exhaustion of power or other failure. And in contrary, a node may be added to the system for the use of work. Thus, dynamic network topology will be useful and necessary for the WSN systems.

6) **Multi-hop routers**. The communication coverage range is limited, mostly around hundreds of meters. So a node can only communicate with its neighbors. If sensor nodes need to communicate with the nodes beyond the communication coverage range, then intermediate nodes must be used as routers. In a cabled network, multi-hop can be achieved by either gateway or routers. However, this kind of important role is taken by common nodes in wireless sensor networks. Hence, a sensor node in a wireless sensor network can be used as either message source or repeater.

7) **Large scale of deployment**. Thousands of sensor nodes can be used to achieve specified application. Sensor nodes can be deployed with high density to keep the wireless sensor network running efficiently.

From the history of the development of the wireless sensor network systems, we can consider that this kind of system is based on the following equation:

WSN Applications = Sensing + CPU + Radio

As the most essential but important element, sensor nodes can be imagined as small computers, extremely basic in terms of their interfaces and their components. They usually consist of a processing unit with limited computational power and limited memory, sensors (including specific conditioning circuitry), a communication device (usually radio transceivers or alternatively optical transceivers), and a power source usually in the form of a battery [37]. Other possible inclusions are energy harvesting modules, secondary application-specific integrated circuit (ASICs), and possibly secondary communication devices. The typical sensor node's structure is shown in Figure 2.3.

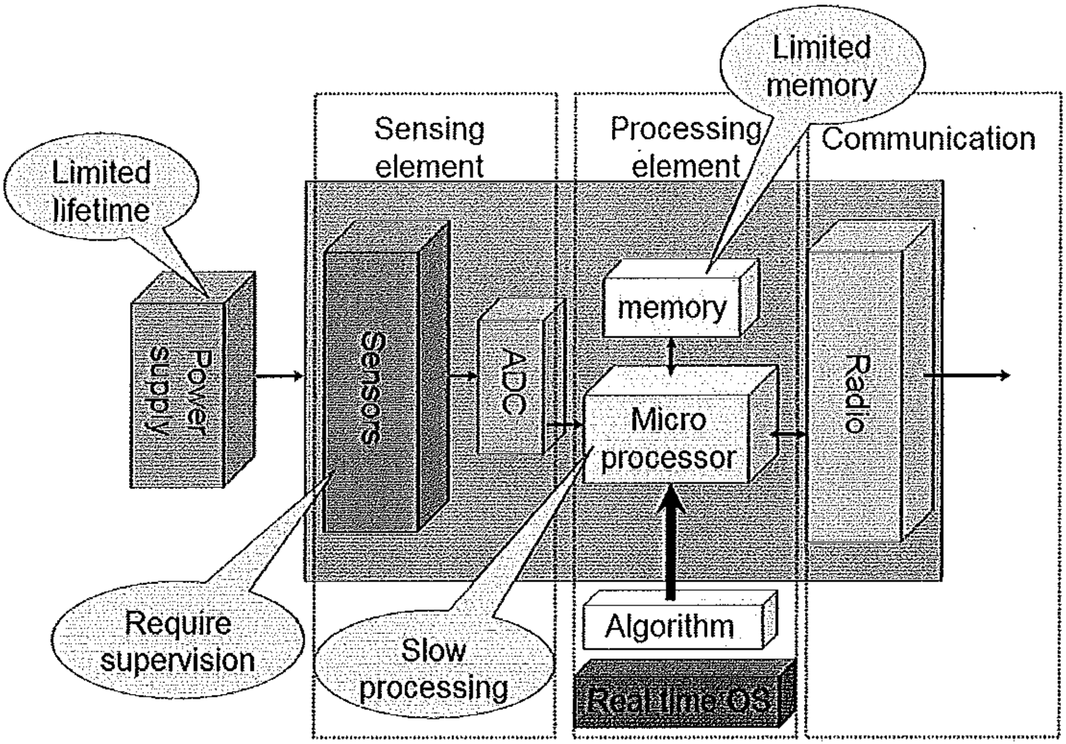


Figure 2.3 Typical Sensor Node's Structure

The base stations are one or more distinguished components of the WSN with much more computational, energy and communication resources. They can act as a gateway between sensor nodes and the end user.

2.1.3 WSN Architecture

The sensor nodes are deployed to acquire measurements such as temperature, signal power, or even dissolved oxygen, in a common WSN architecture. The gateway, which is designed as the administrator of the entire system, provides a critical link between the sensor network and traditional networking infrastructures, including Ethernet, the 802.11 communication standard, and wide-area networks [38]. A typical data processing procedure in a WSN system is like this: the measurement data from each node is firstly collected by the gateway, and then the gateway sends it over a wired connection, typically Ethernet, to a host controller. There, software working platform such as the TinyOS [26] environment can perform advanced processing and analysis and present your data in a fashion way that meets your needs.

2.1.4 Power and Network Standards of WSN

A WSN measurement node contains several components including the radio, battery, microcontroller, analog circuit, and sensor interface. In the systems which are powered by battery, energy is the biggest constraint to wireless sensor capabilities [2]. So we ought to make important trade-offs because higher data rates and more frequent radio usage consume more power. Thus, because WSN mostly running on batteries as energy source, the improvement of low-power management and energy-efficiency are becoming ever important research topics [1].

Often in WSN applications, months or even years of battery life cycle without being serviced is a basic requirement [1]. So many of the WSN systems today are based on IEEE 802.15.4 protocols or ZigBee due to their low-power consumption [7] [8]. The IEEE 802.15.4 protocol defines the Physical and Medium Access Control layers in the networking model, providing communication in the 868 and 915 MHz and 2.4 GHz ISM bands, and data rates up to 250 kb/s. ZigBee is a specification for a suite of high level communication protocols using small, low-power digital radios based on IEEE 802.15.4 standard for wireless personal area networks

(WPANs), such as wireless headphones connecting with cell phones via short-range radio [39]. Providing security, reliability through Mesh Networking (MN) topologies, and interoperability with other devices and standards, ZigBee is targeted at Radio Frequency (RF) applications that require a low data rate, long battery life.

Table 2.1 describes a brief standards comparison between ZigBee/IEEE 802.15.4 and other wireless standards [39].

Table 2.1 Standards Comparison between ZigBee/IEEE 802.15.4 and Other Common Protocols

Market Name Standard	ZigBee 802.15.4	--- GSM/GPRS CDMA/1×RTT	Wi-Fi 802.11b	Bluetooth 802.15.1
Application Focus	Monitoring & Control	Wide Area Voice & Data	Web, Email, Video	Cable Replacement
System Resources	4KB - 32KB	16MB+	1MB+	250KB+
Battery Life(days)	100 - 1,000+	1 - 7	0.5 - 5	1 - 7
Network Size	Unlimited(2^{64})	1	32	7
Maximum Data Rate (KB/s)	20 - 250	64 - 128+	11,000+	720
Transmission Range(meters)	1 – 100+	1,000+	1 – 100	1 – 10+
Success Metrics	Reliability, Power, Cost	Reach, Quality	Speed, Flexibility	Cost, Convenience

2.1.5 Processor Trends

Since it is such a key issue that to save as much energy as possible, if the sensor node is powered by any kinds of batteries, energy saving technique must be applied. A sensor node periodically wakes up to acquire and transmit data by powering on the radio and then powering it back of to conserve energy so as to extend battery life. Thus, the WSN radio ought to transmit a signal efficiently and allow the system to go back to sleep with minimal power usage. Likewise, the technique must also be supported by the processor, so it can be able to wake, power up, and return back to sleep mode efficiently. Microprocessor technology trends for WSNs include reducing power consumption while maintaining or increasing processor speed. This means the power consumption and processing speed trade-off is a key concern when selecting a processor for WSNs, much like your radio choice.

2.2 Localization of WSN Systems

The technology of wireless sensor network has been changing the way people live, and furthermore, WSNs have an endless array of potential applications in booth military and civilian applications. Among those applications, one common feature is shared by all, which is the vitality of sensor location. The core function of s WSN is to detect and report events which can be meaningfully assimilated and responded to only if the accurate location of the event is known. Also, in any WSN, the location information of nodes plays a vital role in understanding the application context. We can get this conclusion because there are three visible advantages of knowing the location information of a WSN node [4]. First, location information is necessary for us to identify the event of interest. For example, a location of a fire accident. Second, location information could be very useful in many other systems, such as network coverage checking. And third, location awareness facilitates numerous application services, such as location directory services that provide doctors with the information of nearby medical equipment. With these advantages, it is natural for location-aware sensor devices to become the standards in WSNs in all application domains that provide location-based service.

2.2.1 Introduction of Location

The concept of location is not limited to the geographic representation of physical location with sets of latitude, longitude, and altitude; it is also applicable to symbolic location in a non-geographic sense such as location in time or in a virtual information space such as a data structure or the graph of a network [40]. Postal zip codes and telephone numbers are good examples of abstractions containing designated location information.

2.2.2 Location in Physical World

Locations in physical world are always viewed as three-dimensional coordinates (x, y, z) in a Cartesian reference coordinate system. On the other hand, many other transformations to other coordinate systems like polar coordinates are equivalent to the Cartesian reference coordinate system [40].

Usually, mere three-dimensional coordinates (*x, y, z*) are not enough for context-aware system services and other information needs to be associated with these position fixes. So, we need to introduce a fourth parameter, *TIME*, to be able to specify where and when a certain event took place resulting in sets of (*x, y, z, t*) for each position fix.

2.2.3 Classification of Localization Techniques

In WSN localization techniques, sensor nodes can be divided into two different kinds. First kind is called as beacon node (BN) or anchor node (AN) and the other kind is so-called unknown node (UN). The proportion of beacon nodes stands at low among all the sensor nodes. Their coordinates are known which can be pre-set manually or by GPS localization equipment. Except for Beacon nodes, all the other nodes in the system are unknown nodes, whose coordinates can be computed by special localization algorithm taking the beacon nodes as references.

The WSNs Localization Techniques can be classified as follows [41]:

1) **Range-based localization** technique versus **range-free localization** technique. Based on measurements of distances between nodes in the processing procedure or not, the localization techniques can be divided into two kinds: range-based localization technique and range-free localization technique. The computation of location by using a range-based algorithm is based on the distance between nodes or some angle information. Then, then wanted location of sensor node can be acquired through the computation using trilateration, triangulation, or multilateration. But no such measurement, whether distance or angle, is needed according to a range-free localization technique. It's based on nothing more than network connectivity. Distance Vector-Hop (DV-Hop) [22] and Multidimensional Scaling-MAP (MDS-MAP) [14] are typical range-free localization algorithm. Furthermore, range-based technique can be used in MDS-MAP to gain more accurate localization results. Some frequently-used measurement techniques are Received Signal Strength Indicator (RSSI), Time Of Arrival (TOA), Time Difference Of Arrival (TDOA), and Angle of Arrival (AOA). Several other techniques can be used as supplementary to the range-based localization techniques, such as multiple measurements [15], and cyclic localization refinement [16]. The core part of this paper presents an enhanced trilateration

localization algorithm, which is a kind of range-based technique.

2) **Absolute localization** versus **relative localization**. Like the physical localization, the result of absolute localization is standard coordinates, such as longitude and latitude. Whereas, in relative localization technique, some special nodes of a WSN system can be taken as references to build a relative coordinates system for the whole network [42]. Absolute localization, which provides the network with unique name space, is hardly influenced by the mobility of sensor nodes. It is more widely applicable than relative localization. But some router protocols (especially based on geo-routing) also can be implemented on the basis of relative localization according to some researches [43]. Moreover, beacon nodes are no longer needed in relative localization technique. Generally speaking, most systems and localization algorithms can achieve the target of absolute localization. Some typical relative localization algorithms are Self-Positioning Algorithm (SPA) [18][58], Local Positioning System (LPS) [19], and SpotON [20]. MDS-MAP can be used as either absolute localization or relative localization according to different network configurations.

3) **Incremental localization** versus **concurrent localization**. According to the differences of nodes' localization computation priorities, the localization techniques can be classified as incremental localization and concurrent localization [44]. The processing procedure to use incremental localization technique is to compute the locations of sensor nodes which are close to beacon nodes, and then the location computation is extended to other sensor nodes. The main disadvantage of this kind of technique is that the accumulated measurement errors which come form the processing procedure. Locations of all nodes are computed at the same time when using the concurrent localization technique.

4) **Centralized computation** versus **distributed computation** [45]. The so-called centralized computation means all the information of the whole network are sent to some kind of central node, such as a server, and all the localization computation works are taken there. Distributed computation means sensor nodes compute their own locations based on interactive information exchange and coordination. The advantages of the centralized computation technique is that it can make overall plans about the system from a global point of view, there is hardly any limits on calculation and storage space, and the computed location can be more accurate than that uses distributed computation. But a quantity of communication work will be undertaken by those

sensor nodes that are more closed to central node, which will be a heavy burden for energy consumption, so those nodes might be out of power too early which can influence the communication and localization computation for the whole system. Convex Optimization [13] and MDS-MAP are typical centralized computation.

5) **Beacon-based localization** versus **beacon-free localization**. The localization techniques can be classified as beacon-based localization and beacon-free localization according to whether beacon nodes are needed [11]. In the beacon-based localization technique, beacon nodes are considered as references of the whole system. They are the basis of the localization processing procedure. An absolute coordinates system will be built after the localization computation. There are no beacon-nodes needed in the beacon-free localization technique. Sensor nodes' positions are the only thing which will be concerned about for the system. The localization computation processing procedure is that, firstly, each node takes itself as the reference; the nodes which are close to it are involved to its own coordinates system. Then after the conversion and combination of the contiguous coordinates systems, a relative coordinates system will be built.

2.2.4 Introduction of Range-Based Localization Techniques

This thesis is based on such kind of range-based localization technique. If an object obtains its distances to multiple devices at known locations, one may estimate its location. A node uses all the distance estimates to compute its actual location. The method employed here to estimate the location depends on the signal features, and can be classified into three main groups as follows [4]:

1) **Trilateration**: This approach is a very distinguished technique in which the positioning system has a number of beacons at known locations. And it uses the geometry of triangles similar to triangulation. However, unlike triangulation which uses AOA to calculate a subject's location, trilateration involves gathering a number of reference tuples of the form (x, y, d). In the term, d represents the distance between the sensor nodes and the beacon with a coordinate (x, y). To compute the actual location of the sensor node using trilateration, a minimum of three reference nodes are needed. Figure 2.4 shows the mathematic model of the trilateration algorithm. In the figure, Node A, B, and C are beacon nodes whose positions are

fixed previously. Node D is a sensor node whose position needs to be calculated. This algorithm will be discussed in detail in Chapter 3.

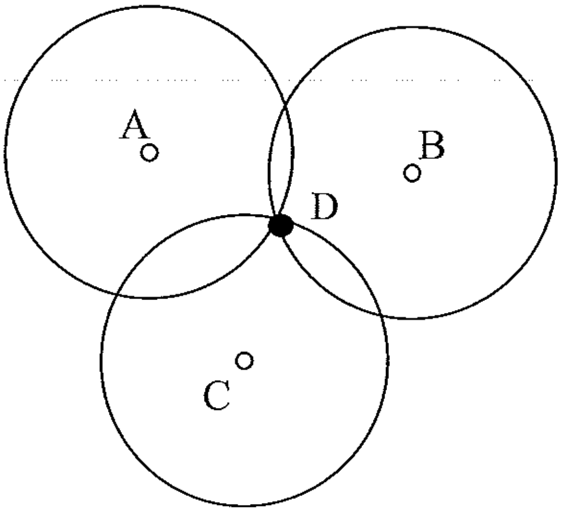


Figure 2.4 Trilateration

2) Triangulation: A large number of methods fall into this class. In short terms, the triangulation method involves gathering AOA measurements at the sensor node at least three beacon nodes whose locations are already known. Figure 2.5 shows the mathematic model of the triangulation algorithm. In the figure, Node A, B, and C are beacon nodes whose positions are (x_a, y_a), (x_b, y_b), and (x_c, y_c). Node D is a sensor node with unknown position. If node D knows the angle to a pair of beacon nodes, for example A and C, it may infer that its position is somewhere on the circle determined by the angle and the position of the two beacon nodes. Assume the position of the center of the circle is O_1 (x_o, y_o), and the radius of the circle is r_1. Then the position of the center O_1 and the radius r_1 can be determined when x_a, y_a, x_b, y_b and angle $\angle ADB$ are known. Similarly, the position of the centers of the other two circles which are found from the node set A, B, D and node set B, C, D and their radiuses can be determined. This may help in transforming a triangulation problem into a trilateration problem if we use the determined centers of the three circles and the radiuses as the distances to create trilateration equations.

3) Multilateration: Multilateration is the process of localization by solving for the mathematical intersection of multiple hyperbolas based on the Time Difference of Arrival (TDOA). In multilateration, the TDOA of a signal emitted from the object to three or more receivers is computed accurately with tightly synchronized clocks. When N receivers are used, it results in N-1 hyperbolas, the intersection of which uniquely positions the object in a $3D$ space. When a large number of receivers are used, $N > 4$, then the localization problem can be posed as an optimization problem that can be solved using, among others, a least squares method. The mathematic model of multilateration algorithm is shown in Figure 2.6. Andreas Savvides and other researchers brought a new multilateration method in [57]. In their work, a sufficient condition which is used to determine whether a node is suitable to participate the collaborative multilateration is presented. Additionally they use kalman filtering technique to improve localization accuracy.

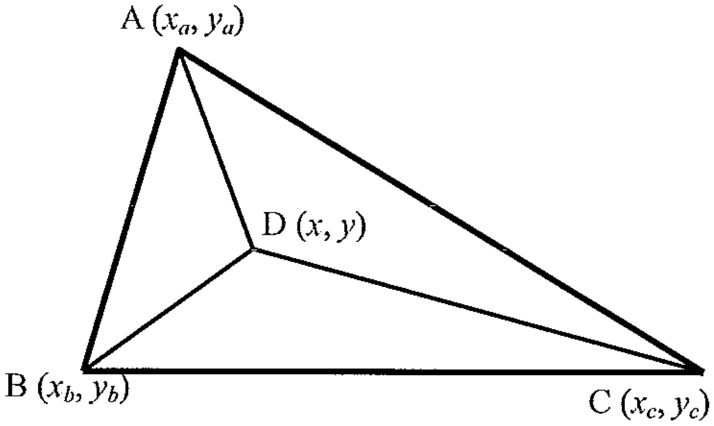


Figure 2.5 Triangulation

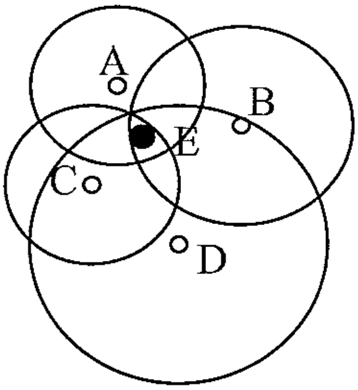


Figure 2.6 Multilateration

2.2.5 Introduction of Range-Free Localization Techniques

Range-free localization never tries to estimate the absolute point-to-point distance based on received signal strength or other features of the received communication signal like time, angle, etc. This greatly simplifies the design of hardware, making range-free methods very appealing and a cost-effective alternative for localization in WSNs. Some examples of range-free localization techniques are introduced as follows:

1) **DV-Hop Localization** [22]. This is the most basic scheme, and it first employs a classical distance vector exchange so that all nodes in the network get distances, in hops, to the landmarks. Each node maintains a table $\{X_i, Y_i, h_i\}$ and exchanges updates only with its neighbors. Once a landmark gets distances to other landmarks, it estimates an average size for one hop, which is then deployed as a correction to the entire network. When receiving the correction, an arbitrary node may then have estimate distances to landmarks, in meters, which can be used to perform the triangulation. The correction a landmark (X_i, Y_i) computes is

$$c_i = \frac{\sum\sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}}{\sum h_i}, \quad i \neq j, \textit{ all landmarks } j$$

In the example in Figure 2.7, nodes L_1, L_2, and L_3 are landmarks, and node L_1 has both the Euclidean distance to L_2 and L_3, and the path length of 2 hops and 6 hops respectively. L_2 computes the correction (40+75)/ (2+5) =16.42, and broadcasts the value of the correction into the network. In this example, assume node A gets its correction from L_2, then its estimate distances to the 3 landmarks would be: to L_1, 3×16.42, to L_2, 2×16.42, to L_3, 3×16.42. These values are then plugged into the trilateration procedure described in the previous section, for A to get an estimate position

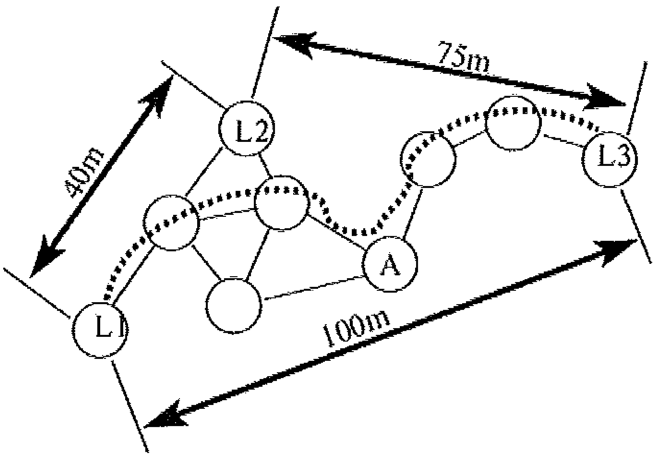


Figure 2.7 Example of DV-Hop Algorithm [22]

2) Self-Positioning Algorithm (SPA) [58]. In this localization algorithm, a bunch of sensor nodes with the highest density in the network will be chosen as location reference group. Furthermore, the node which has the maximum connectivity in the network will be considered as the origin of the established coordinate system. A local coordinate system around each node is firstly built according to the result of the distance estimation between nodes. Then a global coordinate system which is based on the origin will be established by information communication and coordination between nodes. However, the ratio of the communication cost and the amount is almost exponential [58] because all node need to participate in the establishment and transformation of the global coordinate system.

3) MDS-MAP Localization Algorithm [14]. As one of the centralized localization algorithms, MDS-MAP can work in both the range-free and range-based cases. Additionally, it also can be used as either a relative localization technique or an absolute localization technique. It is based on a newly developed technique named multidimensional scaling which comes from the corresponding technique used in psychometrics and psychophysics. This multidimensional scaling technique is frequently used for data analysis and information visualization. The localization procedure of MDS-MAP consists of three steps. First, compute shortest paths between all pairs of nodes in the region of consideration. The shortest path distances are used to construct the distance matrix for MDS. Second, apply classical MDS to the distance matrix, retaining the first 2 (or 3) largest eigenvalues and eigenvectors to construct a 2-D (or 3-D) relative map. Third, given sufficient anchor nodes (3 or more for 2-D, 4 or more for 3-D), transform the relative map to an absolute map based on the absolute positions of anchors.

Table 2.2 Comparison of Existing Work

Reference	Error d	Experiment area A	$d/\sqrt{A}$	Extra Hardware
[59]	7.62	13.71×32	0.360	NO
[60]	2.62	16.40×16.40	0.160	YES
[61]	10	26×49	0.280	NO
[62]	1.83	10×10	0.183	YES
[17]	0.82	5×5	0.164	YES

Table 2.2 show a comparison of existing work of localization algorithms. In each row, we display the reference and the respective average localization error d (unit in feet), the size

of the area of the experiments A (unit in square feet), the ratio $d/\sqrt{A}$ and finally the requirement of extra hardware.

2.2.6 Introduction of Classic Distance Acquisition Techniques

In the range-based localization, the location of the node can be computed to other nodes in its vicinity if the distance between a signal sender and a receiver could be estimated. The accuracy of such estimation, however, is subject to the transmission medium and surrounding environment. So, we can see that the distance estimation is the core of this method. Here, we've got several methods to compute the distance [4]:

1) **Angle of Arrival** (AOA): Range information is obtained by estimating and mapping relative angles between neighbors. Such approaches require an antenna or an array of ultrasound receivers, which can determine the angle and orientation of received signals.

2) **Received Signal Strength Indicator** (RSSI): To obtain the distance between neighbors by using a mathematical model, which translates the received signal strength into distance. This kind of model comes from the phenomenon that signal traveling in a space typically reduce in attenuation with respect to the distance that it travels, and the Received Signal Strength (RSS) can be measured at the receiver's side.

3) **Time of Arrival** (TOA): In the TOA approach, range information is estimated from the signal traveling time between the source sensor and the destination. To use TOA distance estimation, signals that travel at a slower speed are typically needed, say, ultrasound, also, such system needs to be synchronized, which necessitates use of expensive hardware for precise clock synchronization. A GPS is the typical example using TOA estimation.

4) **Time Difference of Arrival** (TDOA): However, this TDOA range estimation is somehow similar to the TOA method. This TDOA approach uses two signals that move at different speeds, such as the radio frequency and ultrasound. And transmission in one direction is sufficient. However, the errors by using TDOA method, just like TOA, come from the clock asynchronous within the system, which means the expensive hardware for precise clock synchronization is basically required.

2.2.7 Localization Technique Performance Indexes

1) Localization Accuracy. Localization accuracy is the most important performance index of localization techniques. The frequently-used performance indexes for localization accuracy are: Mean Square Error (MSE) [46], Root Mean Square Error (RMSE) [47], Cramér–Rao Lower Bound (CRLB) [46], Circular Error Probability (CEP) [48], Geometric Dilution of Precision (GDOP) [49], and so on.

2) Density of Beacon Nodes & Sensor Nodes. The locations of beacon nodes can be pre-set manually or by GPS. Manual deployment of beacon nodes is restrained by the surrounding network environment, and furthermore, the expansibility of the system is limited by deploying the beacon nodes manually. And it will cost much more if the GPS localization technique is introduced to the network system. It is commonly believed that the cost of beacon nodes would be two orders of magnitude as that without GPS technique in the system, which means the cost of the whole network will be 10 times even with only 10% of the sensor nodes are beacon nodes [50]. In WSN systems, the increment of density of sensor nodes will surely increase the cost of the deployment of the network. At the same time, the problem of communication obstruction will be encountered since the communication band width is so limited. Many localization techniques can be influenced by the density of sensor nodes. For example, DV-Hop technique is suitable for localization computation only if the sensor nodes are deployed with high density in the area [22].

3) Cost. The concept of cost in WSN systems can cover several aspects. Energy consumption is one of these aspects. It is one of the factors which makes tremendous influences to the design and application of WSN systems. As a prerequisite, to ensure high localization accuracy, those factors, which are tightly related to energy consumption, such as communication cost, cost of memory, time complexity, and so on, are key indexes, since the power of each sensor node is very limited. Moreover, localization system or algorithm can be judged from several different aspects. The cost of time includes the time of system installation, configuration, and localization. The cost of space includes the needed infrastructure, amount of sensor nodes, and size of hardware.

4) Fault Tolerance & Self Adaptability. Generally, an ideal environment and reliable sensor nodes are quite suitable for any localization system and algorithm. But different problems can be met in real situations, such as signal reflection, diffraction, and communication blind spot. Sensor nodes can be out-of-service because of different reasons such as surrounding environment influences, energy exhaustion, and physical damages. Furthermore, measurement error between sensor nodes can occur by surrounding environment and hardware limitation. But physical preservation or replace of sensor nodes are usually cumbersome or even infeasible. So fault tolerance and self adaptability of the software and hardware of localization system are basic characteristics, so as to adjust automatically, rectify errors, adapt environment, and increase localization accuracy.

5) Other Related Indexes. Different localization systems are applied in different scale of areas, such as in a campus, in a building, or just in a room. Moreover, the amount of located targets, which reflects the capability of localization technique for certain infrastructure during some specific time, is another important index.

Those indexes mentioned here can be used as standards to judge the performances of localization systems and algorithms of WSN systems; however, they also can be treated as the targets to design better systems and algorithms. So, a quantity of research works should be done to achieve the target of optimization. Furthermore, those indexes are related to one another, every aspect of the system should be considered on specific condition, in order to choose or design the proper localization algorithm.

2.3 Summary

In this chapter, we firstly introduced the concept of wireless sensor network, and technique characteristics and system architecture. And then we discussed the basis of localization of WSN system. We focused on the classification of localization techniques and both range-based localization techniques and range-free localization techniques were presented. And then we intensively gave an overview of some classic distance acquisition techniques. The behavior of WSN self localization system and algorithm directly influences its serviceability. It is such a question that how to judge the performance of one localization system. At the end of this chapter, we discussed several frequently-used performance indexes. Each localization system or algorithm has its own characteristics and application area according to existing references. So we need to choose proper systems for specific environments.

CHAPTER 3

SIGNAL PROPAGATION

3.1 Relationship between RSSI and Input Power

The Tmote Sky node uses CC2420 as its wireless communication unit. With sensitivity exceeding the IEEE 802.15.4 specification and low power operation, the CC2420 provides reliable wireless communication. And the radio may be shut off by the microcontroller for low power duty cycled operation. The CC2420 has programmable output power. Common CC2420 register values and their corresponding current consumption and output power are shown in Table 3.1 [52].

Table 3.1 Output Power Configuration for the CC2420

PA_LEVEL	TXCTRL register	Output Power [dBm]	Current Consumption [mA]
31	0×A0FF	0	17.4
27	0×A0FB	-1	16.5
23	0×A0F7	-3	15.2
19	0×A0F3	-5	13.9
15	0×A0EF	-7	12.5
11	0×A0EB	-10	11.2
7	0×A0E7	-15	9.9
3	0×A0E3	-25	8.5

CC2420 has a built-in RSSI providing a digital value that can be read at any time from the 8 bits, signed 2's complement *RSSI.RSSI_VAL* register. Additionally, on each packet reception, the CC2420 samples the first eight chips, calculates the error rate, and produces a Link Quality Indication (LQI) value with each received packet.

The RSSI register value *RSSI.RSSI_VAL* can be referred to the power *P* at the *RF* pins by using the following equations:

$$P = RSSI\_VAL + RSSI\_OFFSET\,[dBm] \tag{1}$$

where the *RSSI_OFFSET* is found empirically during system development from the front end gain. *RSSI_OFFSET* is approximately –45. E.g. if reading a value of –20 from the RSSI register, the *RF* input power is approximately –65 dBm. A typical plot of the *RSSI_VAL* reading as function of input power is shown in Figure 3.1. It can be seen from the figure that the RSSI reading from CC2420 is very linear and has a dynamic range of about 100 dB [25]. The RSSI register value *RSSI.RSSI_VAL* is calculated and continuously updated for each symbol after RSSI has become valid. Figure 3.2 is an example that shows raw RSSI packets collected from sensor nodes.

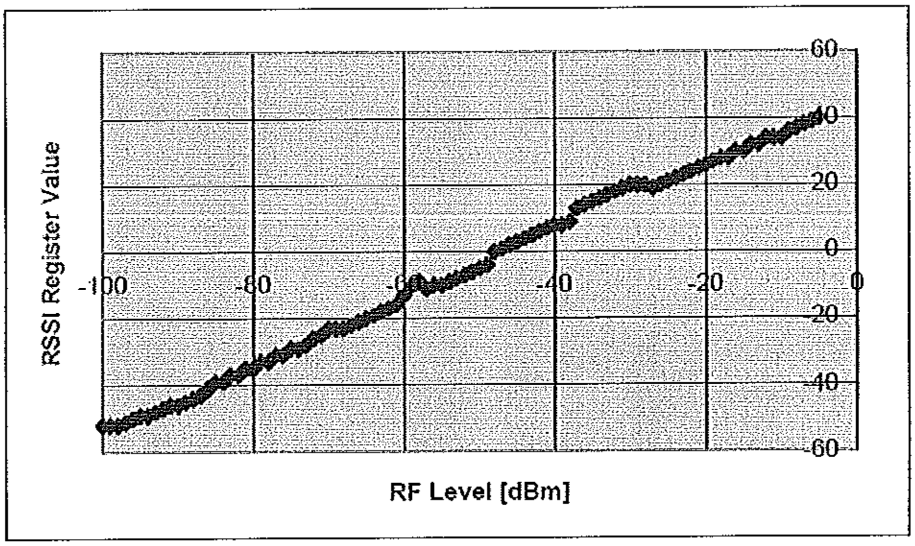
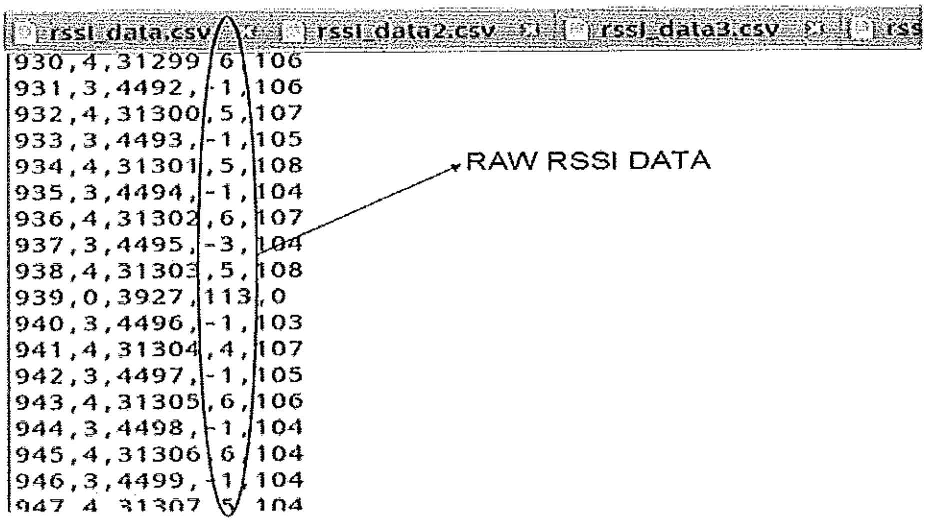


Figure 3.1 Typical RSSI Value vs. Input Power [25]



```
930,4,31299,6,106
931,3,4492,-1,106
932,4,31300,5,107
933,3,4493,-1,105
934,4,31301,5,108
935,3,4494,-1,104
936,4,31302,6,107
937,3,4495,-3,104
938,4,31303,5,108
939,0,3927,113,0
940,3,4496,-1,103
941,4,31304,4,107
942,3,4497,-1,105
943,4,31305,6,106
944,3,4498,-1,104
945,4,31306,6,104
946,3,4499,-1,104
947,4,31307,5,104
```

Figure 3.2 Example of RSSI Collection

3.2 Translation from RSSI to Distance

Since the method we use to calculate the node's position is an enhanced trilateration, one of the range-based algorithms. The role of the translation from RSSI to distance value is very crucial. In the experiment, two nodes were chosen as experimental tools. The distance between the signal sender and the receiver was from 0.1 meters to 4 meters increasing by every 0.1 meters. We measured the RSSI data at each distance for 100 samples. And this experiment was conducted on large wooden tables in an open court near a building, see Figure 3.3.

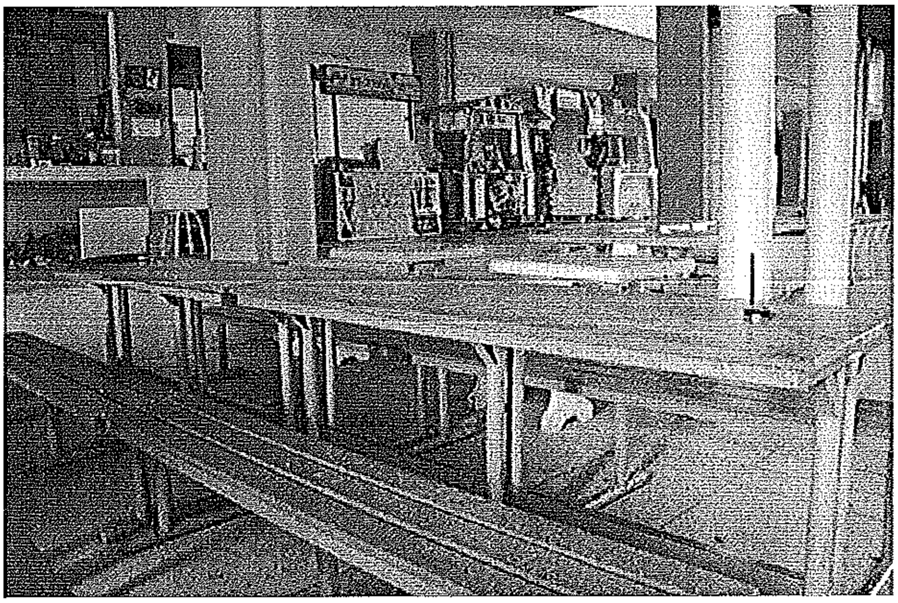


Figure 3.3 Experiment Setup of RSSI Collection

Figure 3.4 shows the performance of radio signal propagation according to increasing distance-values in real experiment environment. The maximum, minimum, and average value of RSSI data samples at each distance is presented in the figure. From this figure, we can see that the received signal strength weakens while the distance between the signal transmitter and receiver increasing, but the relationship between RSSI values and distance-values is non-linear, since a number of factors, say non-isotropic path losses and antenna orientation and so on, are easily encountered. We also noticed that the value of RSSI data become very small when the distance between the sender and the receiver is not large, that's because we used the modified Tmote Sky nodes called Unode modified by the Ubines Laboratory of Prince of Songkla University with external antenna. The signal sent by this kind of node attenuates faster than the original Tmote Sky node.

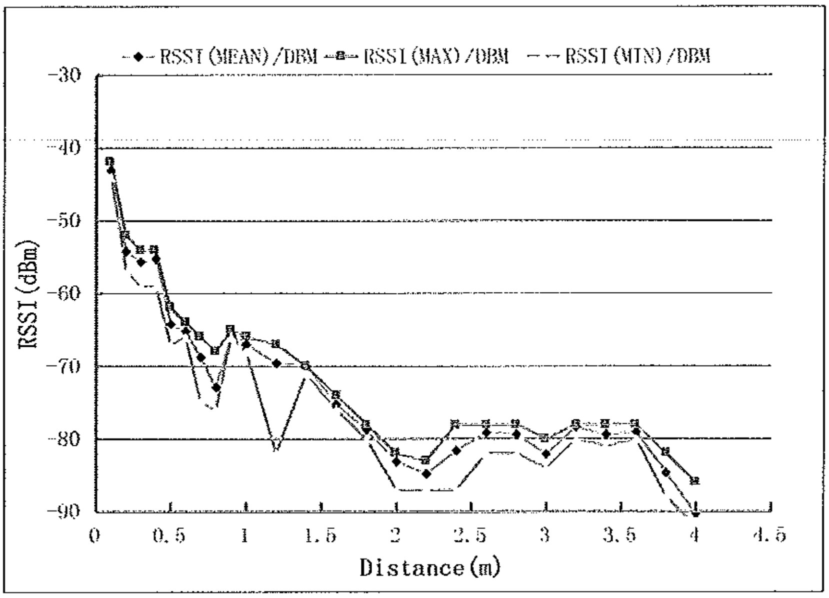


Figure 3.4 Radio Signal Propagation

One of the most common radio propagation is the log-normal shadowing path loss model [28] which also will be adopted in our system. The model is given by:

$$PL(d) = PL(d_0) - 10n\log_{10}(\frac{d}{d_0}) - X_\sigma \qquad (2)$$

where d is the transmitter-receiver separation distance, d_0 is a reference distance, n is the path loss exponent which ranges typically from 2 to 6 [5], and X_σ represents the medium-scale channel fading and is typically modeled as zero-mean Gaussian distribution (in dB) with standard deviation σ (multi-path effects). Assume m means the mean value of n RSSI samples, X_i means the collected value of the i^{th} RSSI sample. Then we can get

$$m = \frac{\sum_{i=1}^{n} X_i}{n} = \frac{X_1 + X_2 + \ldots + X_n}{n}$$

$$\sigma^2 = \frac{\sum_{i=1}^{n} (X_i - m)^2}{n-1}$$

typically, σ is as low as 4 and as high as 12 [5] which implies that the error may be large. In our tests, the value of σ is around 4. In equation (2), $PL(d_0)$ is the signal power at reference distance d_0 (usually at the distance of 1 meter). And $PL(d)$ is the signal power at distance d. The value of $PL(d_0)$ can either be derived empirically or obtained from the WSN hardware specifications. In out experimental environments (nodes placed on large wooden tables, in an open court near a building), the *PL(d$_0$)* is around -65 dBm. When getting RSSI at receiver, the value means *PL(d)* from expression (2), then we can easily calculate the value of path loss exponent *n* by:

$$n = \frac{PL(d_0) - PL(d)}{10\log_{10}(\frac{d}{d_0})}$$

3.3 Analysis of Radio Irregularity

There are lost of factors which can influence the signal propagation for WSN, such as complex terrain, heterogeneous sending power [30], and so on. Generally speaking, the non-isotropic properties of the propagation media and the heterogeneous properties of devices lead to the radio irregularity [29] [30]. Among all these factors, we intensively discuss two main factors in our experimental environment, non-isotropic path losses, and antenna orientation.

3.3.1 Non-Isotropic Path Losses

The property of Non-Isotropic propagation of signal in different directions is a very important issue. Signal can be easily reflected, diffracted, and scattered due to the complexity of the environment in which the signal propagates. Another significant reason for non-isotropic path loss is hardware calibration. It is possible that a node has different antenna gain along all propagation directions.

In practice, we use a pair of Unode motes for our experiments. One of the motes periodically transmits signal packets including RSSI and the other mote samples the RSSI by reading its *RSSI_ VAL* register while receiving these packets. All experiments are conducted on large wooden tables (to decrease the interference from floor or some other hard surface that may bring signal interference, e.g. metal), in open court near a building.

We demonstrate the presence of radio irregularity using two different metrics:

1) The Received Signal Strength In Different Directions: In the first experiment, the receiver is placed 60 centimeters away from the sender (both on the wooden table) and the

received signal strength is measured in four different geographical directions by sampling 100 times in each direction. From Figure 3.5, we can see that the received signal strength in different directions is relatively stable over time. And the received signal is much bigger, when the transmitter lies to the north of the receiver, than those in the other three directions.

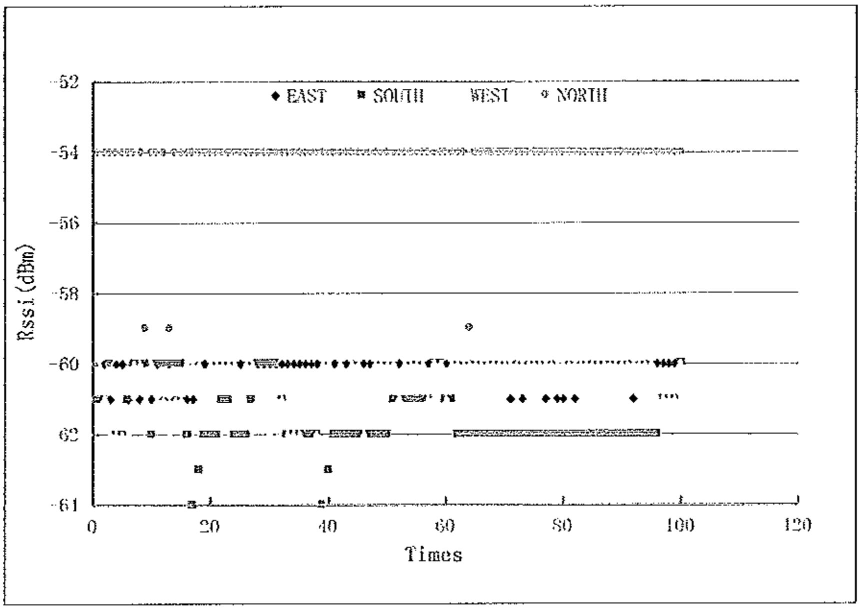


Figure 3.5 Signal Strength over Time in Four Directions

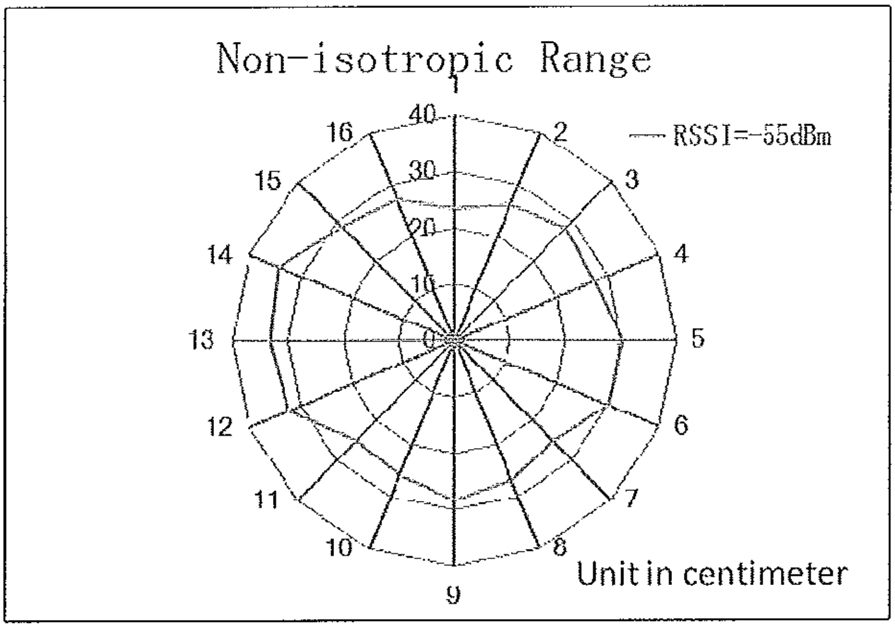


Figure 3.6 RSSI Communication Range

2) The Communication Range: In the second experiment, we fix the received signal strength threshold at -55 dBm. Then with such threshold, we measured the communication ranges in different directions. Figure 3.6 shows that communication range of a mote as the receiver direction varies from degree 0 to degree 359 (the distance measures in unit of centimeters). Comparing the result with Radiated pattern in [52], we can see that the results of communication range of the nodes are related with the specific experimental environment.

3.3.2 Antenna orientation

Antenna orientation is another key issue for the measurement of RSSI collection. In order to test the influence of different antenna orientation of different nodes, we did the following experiment. In this test, we got two Unode motes as experiment tools. One of them was chosen as receiver with its antenna setting horizontal, while the other node was taken as transmitter with its antenna pointing upward. Once the experiment started, the source transmitter began to send signal messages over time. And the receiver collected 100 samples. This experiment was conducted exactly in the same experimental environment as we did to test the performance of radio signal propagation showed in Figure 3.3.

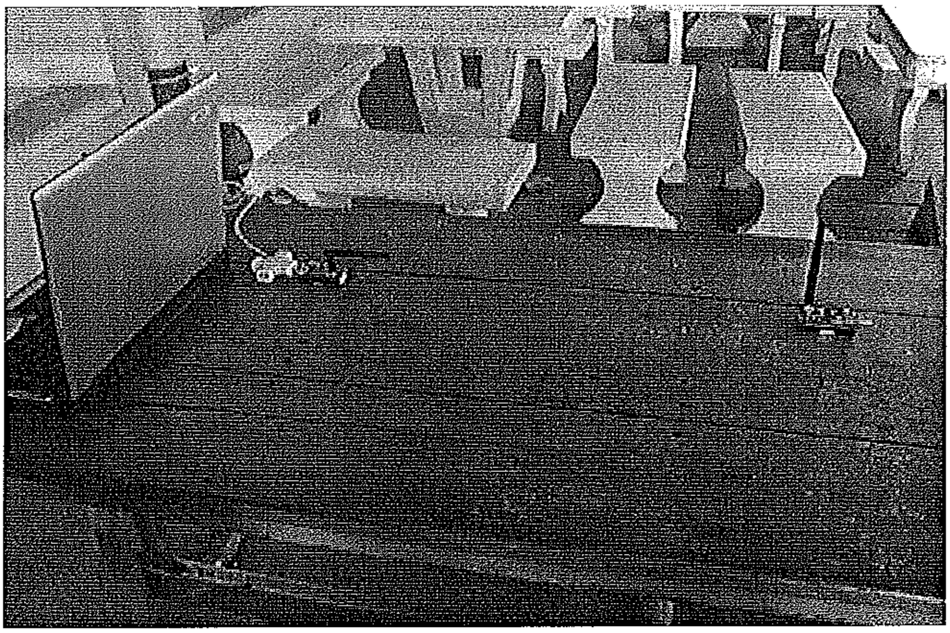


Figure 3.7 Setup of Experiment of Antenna Orientation

Figure 3.8 shows that RSSI become more irregular even if the transmitter-receiver distance is short enough (under 0.4 meters), comparing with Figure 3.3. If we name the angle between the transmitting antenna and vector from the transmitting node to receiving node as angle θ, then the transmit antenna gain should depend primarily on θ. Jude Allred and other researchers have proved in [12] that if $\theta = 0^\circ or 180^\circ$, then the received signal should be weakened because it is transmitted through the hole of the donut, whereas if $\theta = 90^\circ$, the RSSI should be relatively strengthened because it is transmitted through the thickest part of the donut. So, if we deploy the node on tables, then we should set the antennas of the nodes pointing upward.

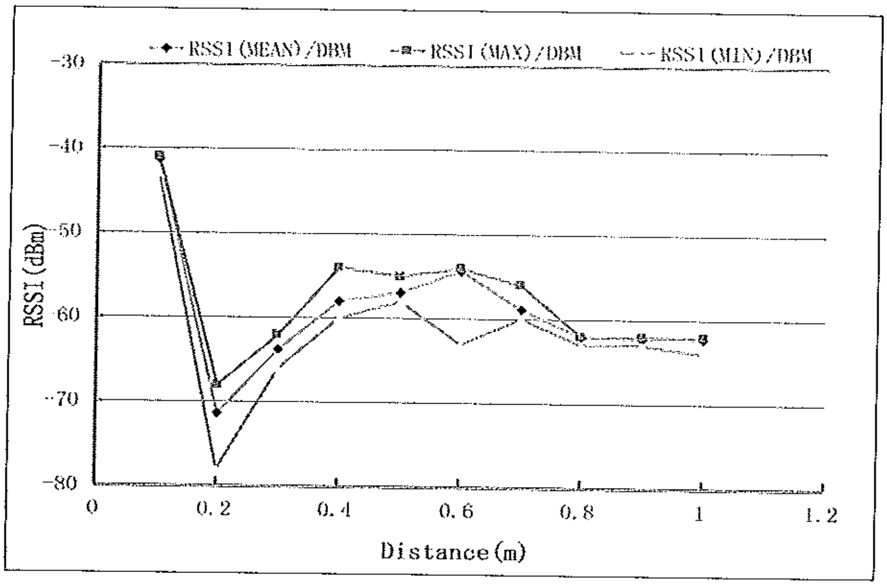


Figure 3.8 Radio Signal Propagation with Different Antenna Orientation

3.4 Experiment Conventions

From the discussion and the experiments we presented in section 3.3, we thus conclude that:

1) To decrease influence produced by non-isotropic path losses, we can test the RSSI-distance relationship in different directions and choose the average value of path loss.

2) For the antenna orientation, this main parameter needs to be carefully considered when analyzing the data collection, we should follow the antenna deployment convention, which is we ought to keep the angle between the transmitting antenna and vector

from the transmitting node to receiving node vertical if possible. In practice if we place nodes on tables, we should have their antennas pointing upwards [32].

3.5 Summary

RSSI, as being used widely because of its simplicity, is one of the core topics of this thesis. In this chapter, we firstly discussed the basis of RSSI and then we introduced a frequently used mathematical model, which are used to translate RSSI to corresponding distance. Furthermore, we pointed out two main factors which can influence the performance of the RSSI collection. Additionally, some experiment conventions are given to improve the performance of radio signal propagation.

Chapter 4

ENHANCED TRILATERATION ALGORITHM AND TESTING

4.1 Enhanced Trilateration Algorithm

There are several algorithms to calculate node's position [4], trilateration is one of the basic methods. Trilateration is a method of determining the relative positions of objects using the geometry of triangles in a similar fashion as triangulation. Unlike triangulation, which uses angle measurements (together with at least one known distance) to calculate the subject's location, trilateration uses the known locations of two or more reference points, and the measured distance between the subject and each reference point. From the range estimation method (here, in this thesis, we use RSSI distance mechanism), we could easily estimate that each range measurement made at a sensor produces a circle centered at that sensor on which the transmitter must lie. So, to accurately and uniquely calculate the position of the transmitter on a *2D* plane by using the circles produced by range measurements of at lease three different sensors, the position can be found at the intersection of the circles.

4.1.1 Review of Traditional Trilateration Algorithm

Figure 4.1 is an ideal situation of trilateration method. *P1*, *P2*, and *P3* are anchor nodes. *A* means a sensor node with unknown position. The solution can be found by taking the formula for three spheres and setting them equal to each other. To simplify the calculations, we apply three constraints to the centers of these spheres: we assume all three spheres are centered on the z=0 plane, one is at the origin (*P1*), and one other is on the x-axis (*P2*). It is possible to transform any set of three points to comply with these constraints, find the solution point *A*, and then reverse the translation to find the solution point in the original coordinate system.

We start with the equations for the three spheres:

$$r_1^2 = x^2 + y^2 + z^2 \tag{3}$$

$$r_2^2 = (x-d)^2 + y^2 + z^2 \tag{4}$$

and $$r_3^2 = (x-i)^2 + (y-j)^2 + z^2 \tag{5}$$

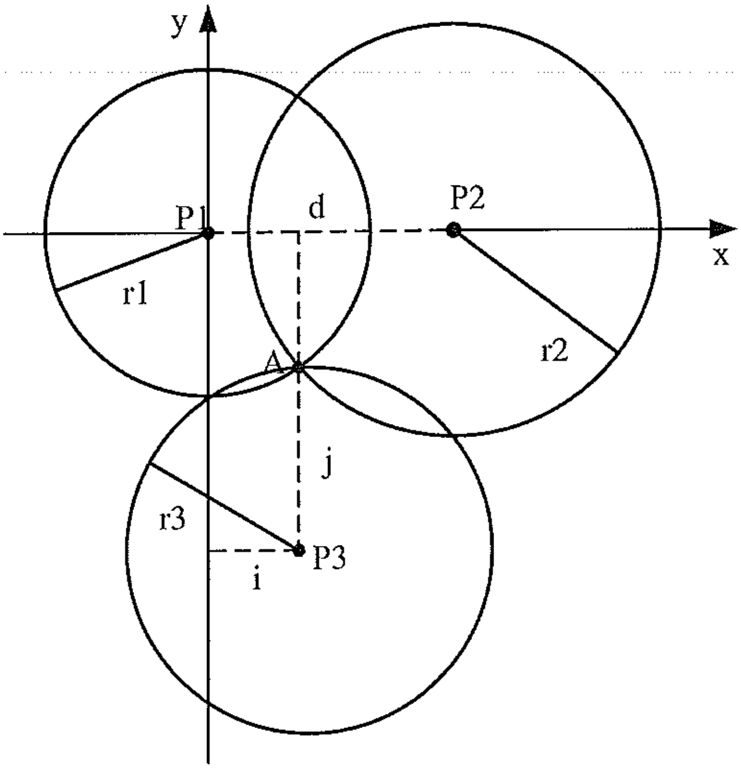


Figure 4.1 Ideal Situation of Trilateration Method. *A* is the position of the target, and *P1, P2, P3* represent the location of the anchor nodes.

Obviously, we need to find the point located at (*x, y, z*) that satisfies all three equations. First we subtract equation (4) from equation (3) and solve for x:

$$x = \frac{r_1^2 - r_2^2 + d^2}{2d}$$

we assume that the first two spheres intersect in more than one point, that is that $d - r_1 < r_2 < d + r_1$. In this case substituting the equation for x back into the equation for the first sphere produces equation (3), the solution to the intersection of the first two spheres:

$$y^2 + z^2 = r_1^2 - \frac{(r_1^2 - r_2^2 + d^2)^2}{4d^2}$$

substituting $y^2 + z^2 = r_1^2 - x^2$ into the equation for the third sphere and solving for y results:

$$y = \frac{r_1^2 - r_3^2 - x^2 + (x-i)^2 + j^2}{2j} = \frac{r_1^2 - r_3^2 + i^2 + j^2}{2j} - \frac{i}{j}x$$

now that we have the x and y coordinates of the solution point, we can simply rearrange the equation for the first sphere to find the z coordinate:

$$z = \pm\sqrt{r_1^2 - x^2 - y^2}$$

Now we have the solution to all three coordinates x, y and z. Because z is expressed as the positive or negative square root, it is possible for it to be zero, one or two roots to the equation.

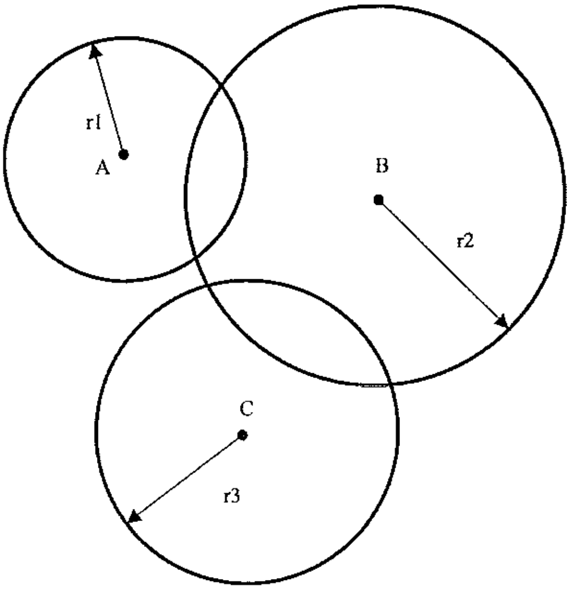


Figure 4.2 Real Situations with Errors of Trilateration Methods

This last part can be visualized as taking the sphere found from intersecting the first and second circles and intersecting that with the third circle:

1) If that circle falls entirely outside of the sphere, z is equal to the square root of a negative number: no real solution exists. This case is presented in Figure 4.2.

2) If that circle touches the sphere on exactly one point, z is equal to zero. See Figure 4.1. In this case, the coordinates of point B is the location of the target.

3) If that circle touches the surface of the sphere at two points, then z is equal to plus or minus the square root of a positive number. Figure 4.3 shows that the third circle intersects the intersecting sphere found from circle 1 and circle 2 with two points. In this case the location of the target, point P, can be obtained by calculating the coordinates of the centroid of the triangle found from points A, B, and C.

Alternatively, the traditional Trilateration algorithm can be presented in another brief mathematical form as follows:

$$D_i = \sqrt{(x_i - x_s)^2 + (y_i - y_s)^2} \tag{6}$$

or

$$D_i = \|X_i - X_s\| \tag{7}$$

where D_i represents the distance between the target node and the i^{th} anchor node, x_i and y_i represent the coordinate values of the i^{th} anchor node, and (x_s, y_s) is the position of the target in the coordinate system. And $\|X\|$ denotes the norm of the vector x.

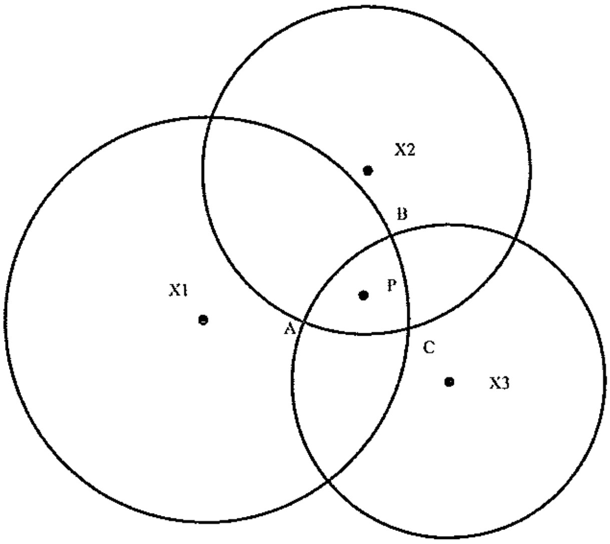


Figure 4.3 A Complicated Case of Trilateration Method

4.1.2 Introduction to the Enhanced Trilateration

Now we know that the traditional Trilateration algorithm is simple and yet effective. But when errors are introduced into the range measurements (this could happen for the inaccuracy of RSSI measurement), the circles do not intersect at a point, or even no intersection at all, which has lead to more complicated algorithm and more cumbersome tasks to calculate the actual position of the source transmitter. Figure 4.2 shows some examples of the situations with errors.

From the previous discussion about the traditional trilateration method, we can see that if there exist more than the minimum anchor nodes (for the *2D* plane, 3 nodes are needed at least), and we could obtain the actual position of the source transmitter by calculating the root of the equations of system, which comes from the equation (6), if we put different variables into equation (6).

In this thesis, we would like to present the possible location of the wanted target in another form, which is, in more details, in the form of the intersection of linear lines. This has the advantage of simpler computation of location. And it could be straightforward applied in some real situation with errors like the cases showed in Figure 4.2 and Figure 4.3. The new location estimation mechanism comes from the truth that what we use in practice is the range measurement based on RSSI to obtain the distance value, which could be easily influenced by surrounding environment.

The new approach to compute the source transmitter's location is based on the observation of Figure 4.4. From the figure, we can see that each circle centered at the sensor node intersect with another circle at two points. And these two points can be used to generate a straight line. Since there are three circles, one linear line could be generated by crossing the two intersection points of every two circles, so, there would be three different lines in total, which have the intersection point just as the three circles do. In Figure 4.4, only two lines are shown, and it could be simply seen where the third line would lie. So, from the analysis of the geometrical character of relationships between the lines and circles, we can conclude that the actual position of the intersection point of the three linear lines is the real location of the source transmitter, and that's what we wish to get here.

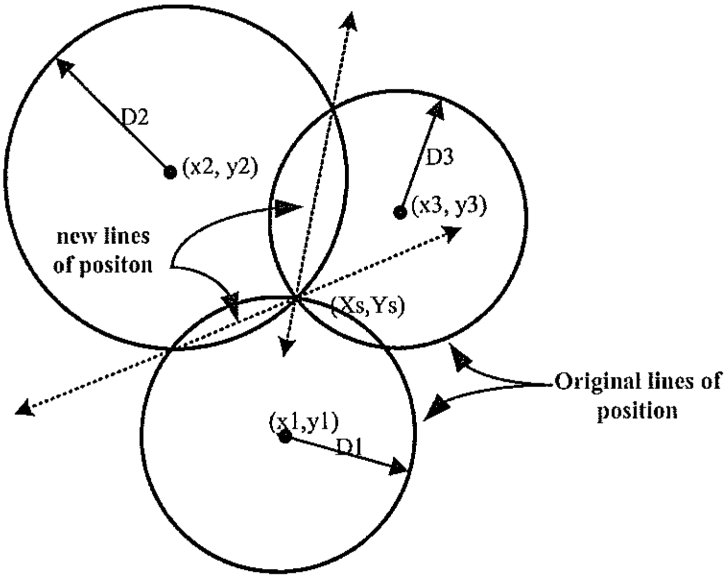


Figure 4.4 Geometrical Character of Linear Lines and Circles

To determine the equations for the new generated linear lines in Figure 4.4, we must start with the original equations for the location estimation. Given i=1,2,3 for equation (6), consider the sensors at x1,x2, and x3. So, firstly we would get the following three equations:

$$D_1 = \sqrt{(x_1 - x_s)^2 + (y_1 - y_s)^2}$$
$$D_2 = \sqrt{(x_2 - x_s)^2 + (y_2 - y_s)^2}$$
$$D_3 = \sqrt{(x_3 - x_s)^2 + (y_3 - y_s)^2}$$

The equation for the straight line which passes the intersection points of circle x1 and x2 can be determined by squaring and some arithmetic operations of the first two equations above, which results in:

$$2(x_2 - x_1)x_s + 2(y_2 - y_1)y_s = (x_2^2 + y_2^2) - (x_1^2 + y_1^2) + D_1^2 - D_2^2$$

where

$$\|X_1\| = \sqrt{x_1^2 + y_1^2}$$
$$\|X_2\| = \sqrt{x_2^2 + y_2^2}$$

so the equation of the linear line determined by circle x_1 and x_2 is

$$(x_2 - x_1)x_s + (y_2 - y_1)y_s = \frac{1}{2}(\|X_2\|^2 - \|X_1\|^2 + D_1^2 - D_2^2) \tag{8}$$

And the equation of the linear line found by sensor 2 and sensor 3 could be determined by using the same procedure, and the result should be:

$$(x_3 - x_2)x_s + (y_3 - y_2)y_s = \frac{1}{2}(\|X_3\|^2 - \|X_2\|^2 + D_2^2 - D_3^2) \tag{9}$$

A similar straight line could be easily generated for sensor 1 and sensor 3. However, the truth is that none of the three linear lines is independent from the other two lines. The reason for this is the third equation could be found just by adding the equations above, which eliminates all terms involve sensor 2, leaving an equation identical to (8) or (9). So from the analysis, we can see that although it's convenient to determine three straight lines generated by the relationships of the three sensors and the signal transmitter, only two of the lines are independent, and it turns out, the third line will always intersect the other two lines at the same intersection point, even if there are range estimation error happens. So, the algorithm based on the intersecting lines only requires two of the three possible generated straight lines.

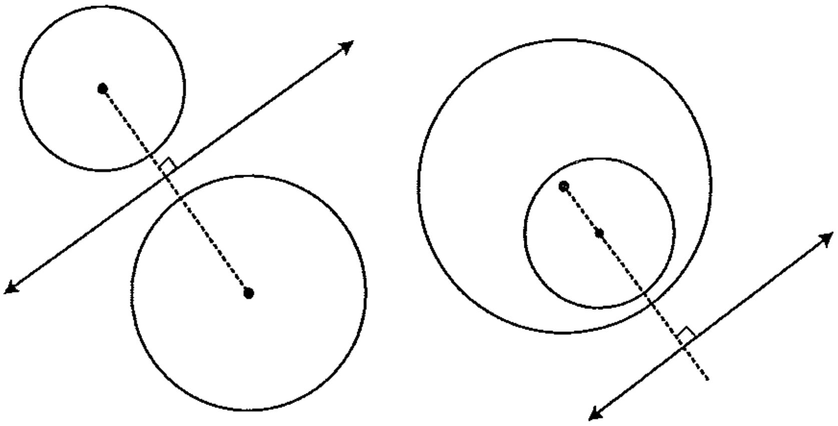
So the location of the source transmitter could be obtained by solving equation (8) and (9) for ys, equating the results and solving for xs, which produces:

$$x_s = \frac{\frac{1}{2}(y_2 - y_1)(\|X_3\|^2 - \|X_2\|^2 + D_2^2 - D_3^2) - \frac{1}{2}(y_3 - y_2)(\|X_2\|^2 - \|X_1\|^2 + D_1^2 - D_2^2)}{(x_3 - x_2)(y_2 - y_1) - (x_2 - x_1)(y_3 - y_2)} \quad (10)$$

substituting x_s into either (8) or (9), and solving for y_s produces

$$y_s = \frac{\frac{1}{2}(x_2 - x_1)(\|X_3\|^2 - \|X_2\|^2 + D_2^2 - D_3^2) - \frac{1}{2}(x_3 - x_2)(\|X_2\|^2 - \|X_1\|^2 + D_1^2 - D_2^2)}{(y_3 - y_2)(x_2 - x_1) - (y_2 - y_1)(x_3 - x_2)} \quad (11)$$

Because of the possibility that the distance measurement error would happen, one consideration must be taken into account, which is what if a circle centered at a sensor might not intersect another circle. Figure 4.5 shows two possible cases that may occur.



(a) (b)

Figure 4.5 Situations with no Intersection between Circles

In both cases above, it is still possible to generate linear lines according to the circles by using the equations (8) and (9). Based on the relationships of the linear lines and the circles, notice that the generated straight line is perpendicular to the baseline connecting the two centers of the circles, and it is also parallel to the tangents of the two circles at which the baseline intersects the two circles no matter whether there are intersections or not between circles. As mentioned previously, even if there is no overlapping, an intersection point would still be determined by generating three linear lines of every two circles, and actually, two of those lines are sufficient to find out the expecting point, because the third line will obviously pass the intersection point of the other two straight lines. So, there are two conclusions here. Firstly, three sensor nodes are still the minimum number to determine the position of the signal transmitter.

Secondly, from the equation (8) or (9), we can see that baseline connecting the centers of the two circles should not be horizontal or vertical, if any of these cases happens, a simple rotation of the coordinate system would help to solve the problem and after the computation, the coordinate system should be rotated back.

4.2 Experiment Environment

4.2.1 Experiment Model

In the experiments, several Unode motes are chosen as resource transmitters with unknown location. Additionally, anchor nodes are deployed in the experiment area, which are used to compute the wanted position values of the resource targets through both the traditional trilateration method and the enhanced one. Figure 4.6 is the situation of the experiment model.

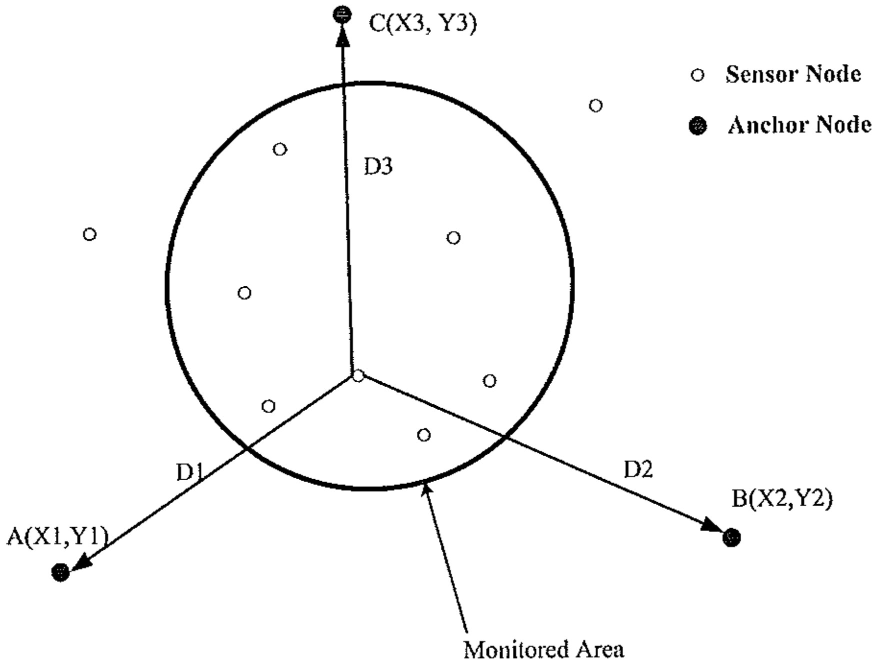


Figure 4.6 Experiment Model

4.2.2 Experiment Procedure

The procedure for our scheme to obtain wanted localization of sensor nodes will be introduced as follows: once the system starts, sensor nodes periodically transmit radio frequency (RF) signal to anchor nodes, whose location information is set beforehand. During this period, anchor nodes will constantly sample received signal strength (RSS) from each sensor node orderly and send those messages which contains the RSSI values collected from the sensor nodes to the data base. The data base connects to the PC. After obtaining enough information, we can calculate the locations of target sensor nodes through our localization scheme by the PC rather than sensor node itself.

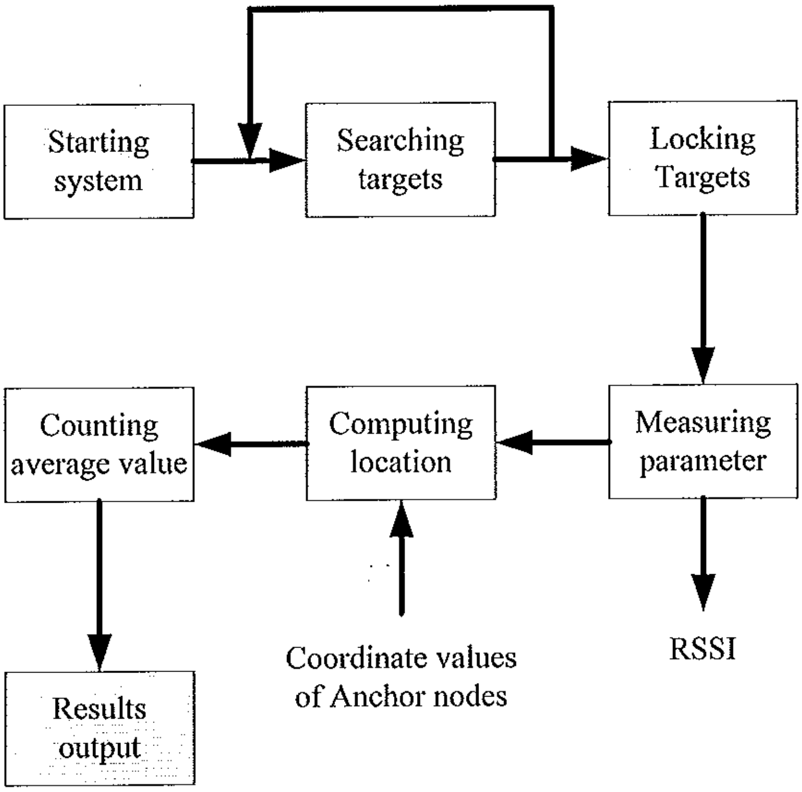


Figure 4.7 General Flow Process Diagram of Experiment Procedure

4.3 Setup and Configuration

According to the real performance of the location and tracking for WSN, accuracy is one of the most important criterions in the system. So, in practice, we would like to use three sensor nodes as the anchor nodes to test the performance of traditional trilateration algorithm and then increase the amount of anchor nodes to four and five to see the advantages of using the enhanced trilateration method.

When four anchor nodes are used, 12 straight lines can be determined by using the 4 subset of every three different sensor nodes. And for every three lines, one of which would be generated by the other two lines, and it will undoubtedly go through the intersection point found by the other two lines. So that third straight lines is redundant for computing the intersection point, which means for the 12 linear lines, four of them are redundant, and four different points can be determined by the rest of the 8 lines. Similarly, ten different points can be obtained by 20 straight lines when five anchor nodes are used.

From those intersection points, the estimation can be calculated by computing the centroid of the polygon formed by the four or ten points, which could be easily obtained by computing the means of the coordinate values of the four or ten points.

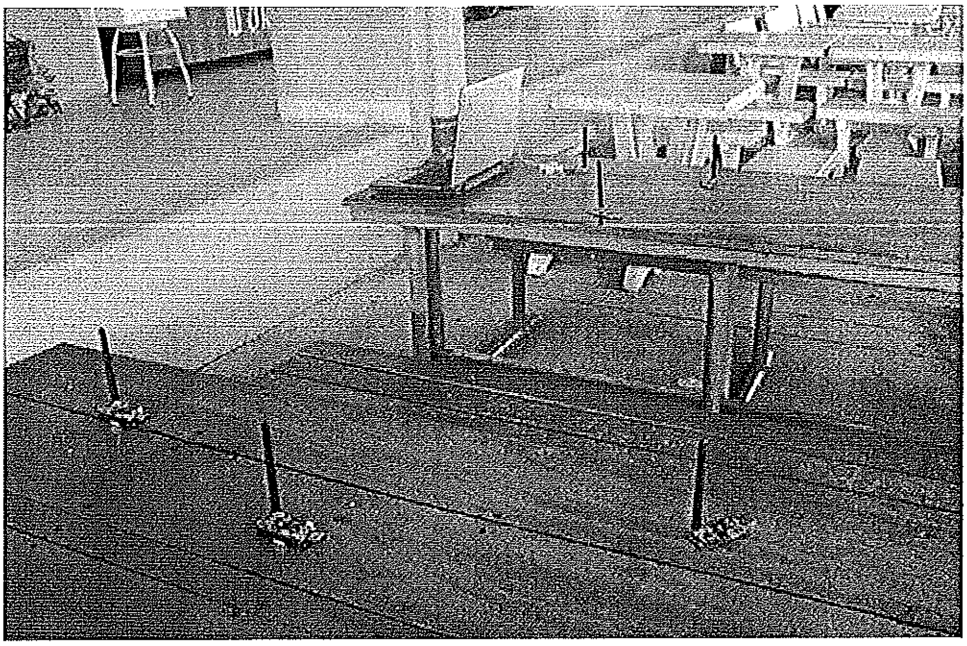


Figure 4.8 Setup of Experiment of the Enhanced Trilateration Algorithm

4.3.1 Experiment Setup

In the set-up of experiment, the modified Tmote Sky nodes are chosen as experimental tools. A sensor node with unknown position was placed under the coverage of the anchor nodes' RSSI collection. Then we tested 15 different positions of the sensor node. And we computed the wanted position values by using both the traditional trilateration method and the enhanced one, which means we firstly set 3 anchor nodes (anchor nodes O(0, 0), A(0, 1.31) and

B(-0.5, 0.5), all in unit of meter)and then increase the numbers of anchor nodes to 4 (anchor node C(-0.54, 0.87)), then 5 (anchor node D(-0.29, 1.29)), to see how accurate it would turns out by using the enhanced trilateration.

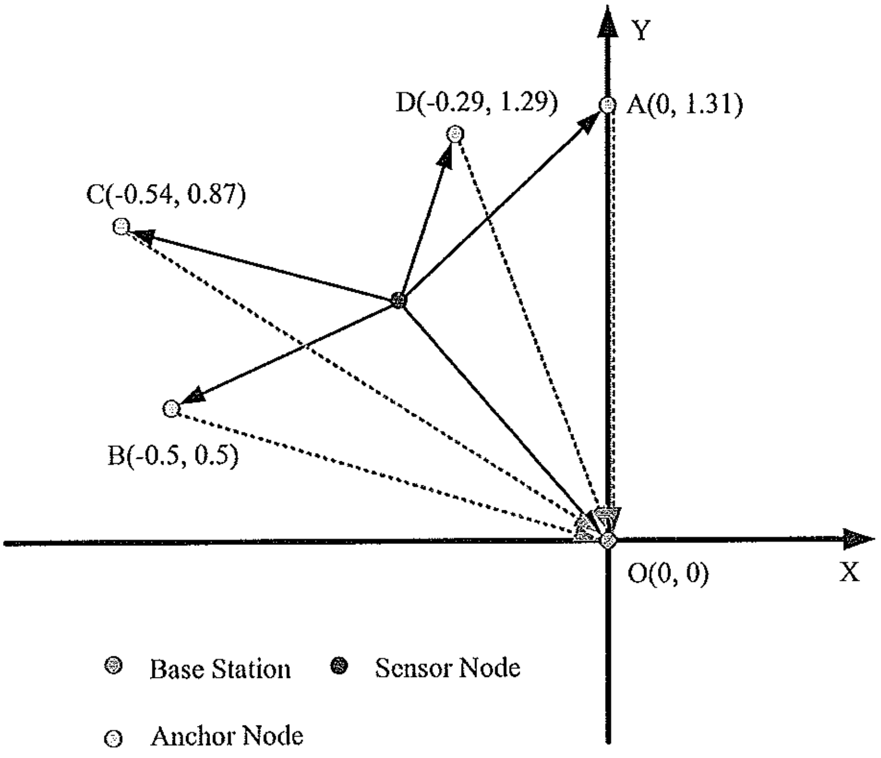


Figure 4.9 Mathematical Model of Experiment of the Enhanced Trilateration Algorithm

Furthermore, considering the conclusions we mentioned in Chapter 3, we choose to test the RSSI-distance relationship in 4 different directions and use the average value to obtain path loss n, in stead of testing the RSSI-distance relationship only in one direction. And in order to decrease the influence brought by different antenna orientation, we set all the antenna of the nodes antenna pointing upward. Additionally, we use the default setting of transmit power level and radio frequency channel in order to make the performance of signal propagation more stable [30], which means PA_LEVEL is 31 and out power is 0 dBm (see Table 3.1), and radio frequency channel which is channel 26 which is Fc = 2405 + 5 (26-11) MHz = 2480MHz.

All experiments were conducted on large wooden tables (to decrease the interference from floor or some other hard surface that may bring signal interference, e.g. metal), in open court near a building.

4.3.2 Software Configuration

All the tests and experiments are done on the platform of TinyOS 2.10, which is properly installed in an Ubuntu 9.04 laptop environment.

From the previous discussion, we can see that all the sensor nodes are classified into 3 different types:

1) Unknown nodes, whose location information are unknown. These nodes are programmed to send signal messages, including their IDs over time.

2) Anchor nodes, whose locations are preset manually beforehand, they firstly obtain the messages sent by unknown nodes, which includes the unknown nodes' ID and the information of weakened signal. And then these messages are retransmitted by anchor nodes to the base station for further use.

3) Base station, which is a node connect to the laptop directly. It is in charge of receiving both the messages sent by unknown nodes and the retransmitted ones sent by anchor nodes. Furthermore, the function of the base station is much more complex, which can be treated as a bridge between the laptop and other sensor nodes.

4.4 Testing, Simulating and Result

After the experiment set-up, we collected RSSI information in four different directions. We obtained the average signal strength PL(d0)=-66.99 dBm. In addition, we tested the relations of received signal power P at some distance d, considering equation (2) and did some mathematical operations, we got corresponding path loss n. After that, we could calculate the average path loss n=2.39 in such special experimental environment. In order to test the accuracy differences of localization brought by different path loss, we simulated the performance of both the traditional and enhanced trilateration algorithm by using n=1.89 and n=2.89 as well.

1) Path Loss n=2.39: The performance of traditional trilateration is very poor, only one fifteenth locations computed, which means the occasion showed in Figure 4.5 happened in all the left 14 positions. In other words, it means that if RSSI measurement error happens in distance estimation, it's impossible to compute the sensor node's position by using the traditional trilateration algorithm. On the other hand, the enhanced trilateration method performed much

better than the traditional one. The average error by using 4 and 5 anchor nodes is 0.44 meters and 0.35 meters respectively. Figure 4.10 shows the localization accuracy by using 4 and 5 anchor nodes with path loss n=2.39. In this figure we can see the localization error of the sensor node from the real position. In all the 15 tested positions, localization results of position NO. 6, 7, 8, 13, 14, 15 shows that the performance of the enhanced trilateration algorithm with 5 anchor nodes is good, with an average position error of around 0.1 to 0.2 meters. However, the localization results of using 5 anchor nodes were a little worse than that used 4 anchor nodes at tested position no. 2, 3, 10 and 14, that's because RSSI collection is so unstable anyway, and the localization algorithm is based on such measurement. Hence, if any RSSI collection becomes terrible of the anchor nodes, it could influence the final result. But in general, the performance of using 5 anchor nodes is better than using 4 or 3 anchor nodes.

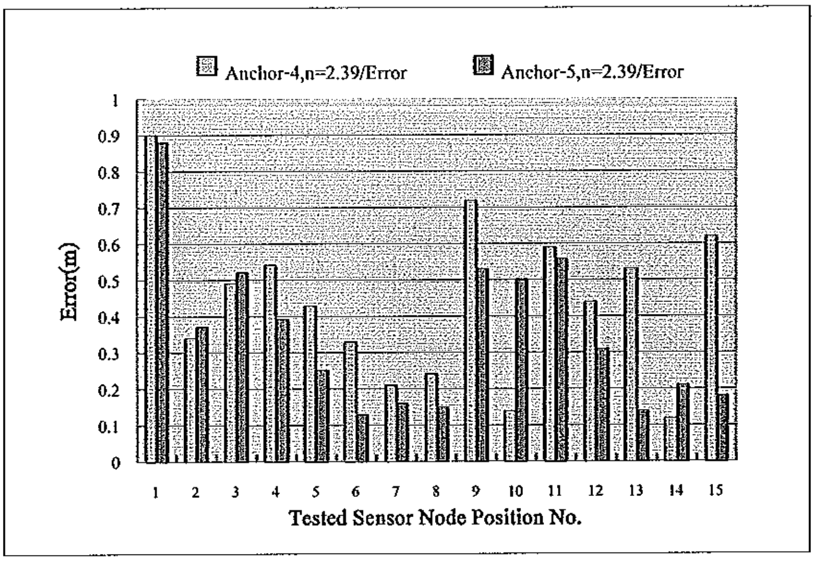


Figure 4.10 Comparison between 4 Anchor Nodes and 5 Anchor Nodes with Path Loss n=2.39

2) Path Loss n=1.89: This is a simulating case. From the result, it turns out that when path loss exponent changes to 1.89, the performance of using 3 anchor nodes is terrible. None of the sensor's position can be computed because of the RSSI measurement error. While the average localization error by using the enhanced trilateration with 4 anchor nodes is 0.45 meters and 0.35 meters with 5 anchor nodes. Figure 4.11 shows the localization accuracy by using 4 and 5 anchor nodes with path loss n=1.89. And the performance of enhanced trilateration algorithm by using 4 and 5 anchor nodes is similar to the first case.

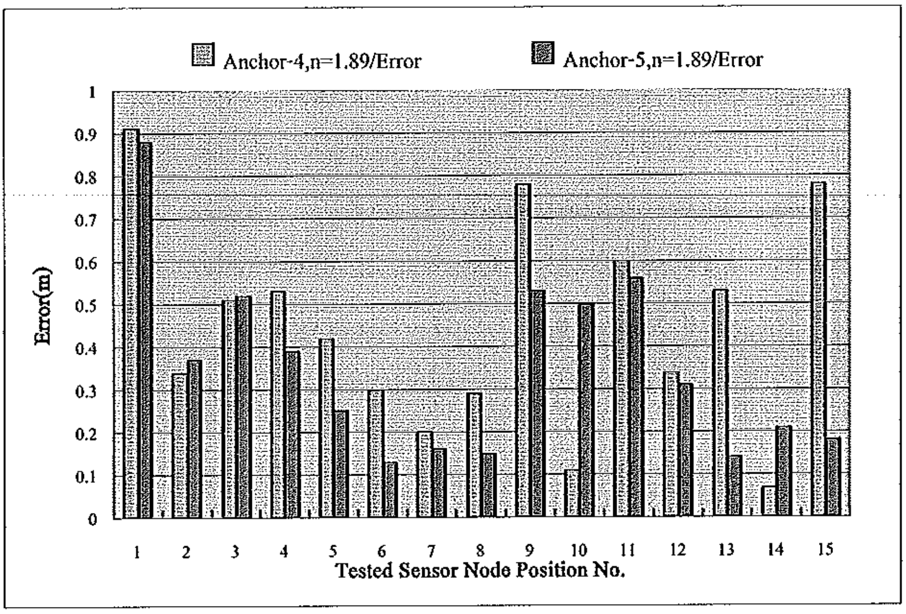


Figure 4.11 Comparison between 4 Anchor Nodes and 5 Anchor Nodes with Path Loss n=1.89

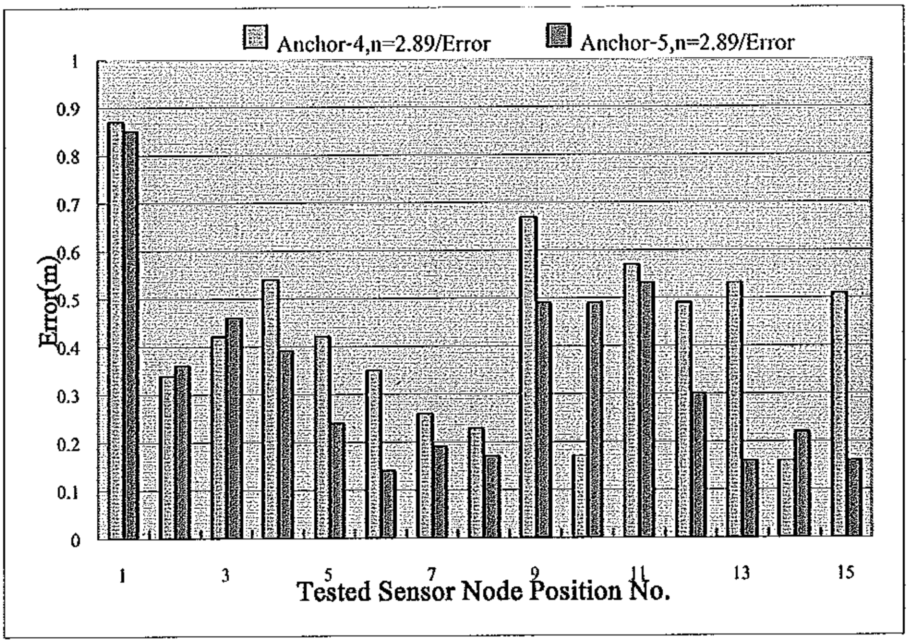


Figure 4.12 Comparison between 4 Anchor Nodes and 5 Anchor Nodes with Path Loss n=2.89

3) Path Loss n=2.89: This is another simulating case. The traditional trilateration performs the best in this case, but still not as good as the enhanced method. We obtained 6 positions that are 40% in total, although the average localization error of these 6 positions is 0.2 meters. But we can see how unstable for using the traditional method. Contrasting the instability of traditional trilateration, we can see the good stable performances using the enhanced method as

before. In this case, the average localization errors with 4 and 5 anchor nodes are 0.44 meters and 0.34 meters. Figure 4.12 shows the localization accuracy by using 4 and 5 anchor nodes with path loss n=2.89. And the performance of enhanced trilateration algorithm by using 4 and 5 anchor nodes is similar to the first case.

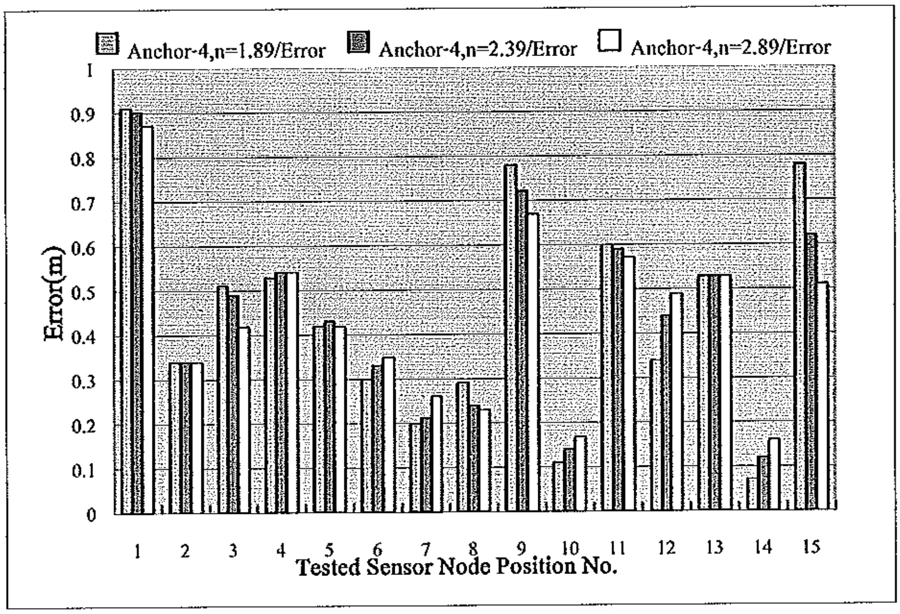


Figure 4.13 Errors of Different Path Loss with 4 Anchor Nodes

Moreover we have noticed that, unlike the obvious difference in performances of the traditional trilateration in those three cases with different path losses, the results of enhanced method are stable although the path loss value changes. The average localization errors by using 4 anchor nodes with different path loss n=1.89, 2.39, 2.89 are 0.45 meters, 0.44 meters, 0.44 meters. Similarly, the errors of 5 anchor nodes are 0.35 meters, 0.35 meters, 0.34 meters. And Figure 4.13 and Figure 4.14 are two visual representation of this information.

4.5 Comparison with Other Localization Algorithms

Table 4.1 show a comparison of existing work of localization algorithms with our enhanced trilateration algorithm. In each row, we display the reference and the respective average localization error d (unit in feet), the size of the area of the experiments A (unit in square feet), the ratio $d/\sqrt{A}$ and finally the requirement of extra hardware. In order to keep the comparison under the same unit, we changed our data which were measured in the unit of meter to the data which were in unit of feet.

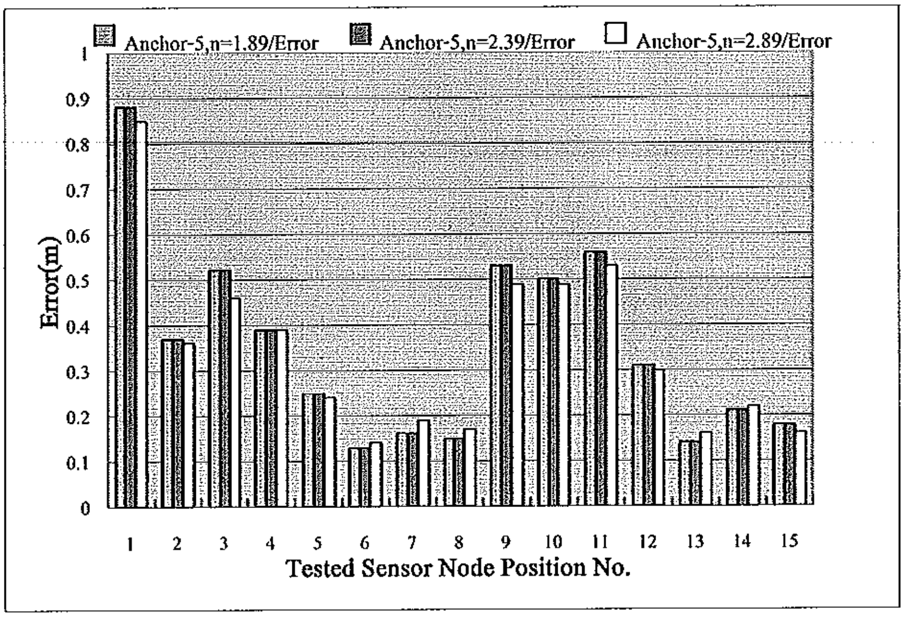


Figure 4.14 Errors of Different Path Loss with 5 Anchor Nodes

Table 4.1 Comparison of Existing work with Our Algorithm

Reference	Error d	Experiment area A	$d/\sqrt{A}$	Extra Hardware
[59]	7.62	13.71×32	0.360	NO
[60]	2.62	16.40×16.40	0.160	YES
[61]	10	26×49	0.280	NO
[62]	1.83	10×10	0.183	YES
[17]	0.82	5×5	0.164	YES
Our scheme (4 anchors)	1.44	6.56×6.56	0.219	NO
Our scheme (5 anchors)	1.12	6.56×6.56	0.171	NO

It confirms the truth that comparing with range-free localization in [60] [62] and [17], the RSSI-based localization algorithms need no extra hardware. However, the localization result usually is not as good as that using a range-free algorithm.

4.6 Summary

In this chapter, we present an enhanced trilateration algorithm and compare the experiment results of using both the traditional and enhanced trilateration methods. In the sight of experiments, we can conclude that although RSSI is under appreciate, we can still try to improve its performance by adjusting some factors which can influence the signal propagation, such as non-isotropic path losses, antenna orientation. Finally, a comparison of Existing work with Our Algorithm is presented.

CHAPTER 5

CONCLUSION AND DISCUSSION

5.1 Conclusion

As one of the core techniques of wireless sensor networks, the localization technique is quite crucial and challenging. This thesis focuses on range-based localization technique for wireless sensor networks, introduces an enhanced trilateration algorithm based on the exploration of correlative techniques of wireless sensor networks. This study discusses factors that can influence signal propagation, moreover, gives some advices to decrease those influences.

5.1.1 Traditional Trilateration Algorithm

This thesis firstly analyzed the mathematical model of the traditional trilateration algorithm, and then tested the performance of it both in real experimental environment and in simulation. From the results, we can see clearly that as one of the range-based localization algorithms, the final results of traditional trilateration method is poor and unstable because there are so many factors that can influence the signal propagation in space.

5.1.2 Enhanced Trilateration Algorithm

The enhanced trilateration algorithm is based on the observation of the traditional trilateration algorithm, but from a different angle. Hence, 3 anchor nodes are still the minimum number to use the enhanced trilateration algorithm. In this thesis, we focus on implementing, investigating and evaluation the performance of the enhanced trilateration algorithm. Additionally, we have the conclusions that are useful for decreasing the influences brought by several factors while radio signal propagating in the space in Chapter 3. From the result, we can see that the enhanced trilateration algorithm performs better, or say more stable, than the traditional trilateration algorithm. And the performance of using 5 anchor nodes is better than using 4 anchor nodes in general. See Table 5.1

Table 5.1 Comparison of Localization Results of Traditional and Enhanced Trilateration Algorithm

Anchor Node	3			4			5		
Path loss exponent	1.89	2.39	2.89	1.89	2.39	2.89	1.89	2.39	2.89
Localization rate	0/15	1/15	6/15	15/15	15/15	15/15	15/15	15/15	15/15
Average error	---	0.08	0.2	0.45	0.44	0.44	0.35	0.35	0.34

5.2 Discussion

The result of this work produces several benefits. However, there are many limitations which require improvements.

5.2.1 Advantages:

1) 100% of the locations of target nodes can be computed by using the enhanced trilateration algorithm. However only part of the locations can be computed if we use the traditional one.

2) The enhanced trilateration algorithm is based on the observation of the mathematical model of the traditional one. In addition, we do not need to upgrade the hardware because it's a kind of RSSI based algorithm.

5.2.2 Limitations:

The main limitation is that as one range-based localization algorithm, the measurement of RSSI directly influences the performance of the enhanced trilateration algorithm, although we have discussed the factors which can affect the signal propagation and given some advices on how to decrease those influences. So, when the measurement of RSSI is terrible, the location of the target may not be accurate enough.

Furthermore, since we use the modified Tmote Sky nodes with external antennas which lead to a quick signal attenuation in practice. Additionally, because of the restriction of experiment field, we just tested the performance of our enhanced algorithm in a small scale. However, theoretically speaking the enhanced trilateration algorithm could work properly in a larger scale.

5.3 Future work

In future work, we will firstly concentrate on those aspects which influence the radio signal propagation, try to find an effective method to decrease the impact brought by radio irregularity on the localization algorithm of wireless sensor networks. Secondly, since the link quality indication (LQI) is another important index of CC2420, we will try to seek a way to make the localization algorithm better by using both RSSI and LQI.

REFERENCES

[1] C. Behrens, O. Bischoff, M. Lueders, and R. Laur, "Energy-efficient topology control for wireless sensor networks using online battery monitoring," *Advances in Radio Science* 5, pp. 205-208, 2007.

[2] John Paul Walters, Zhengqiang Liang, Weisong Shi, and Vipin Chuadhary, "Wireless Sensor Network Security: A Survey," in *SECURITY IN DISTRIBUTED, GRID, AND PERVASIVE COMPUTING*, chapter 16, Auerbach Publications, CRC Press, 2006, pp. 367-404.

[3] F. L. LEWIS, "Wireless Sensor Networks," in *Smart environments: technologies, protocols, and applications*, PART 2, First Edition, Wiley-Interscience, November 2, 2004, pp. 13-46.

[4] Avinash Srinivasan and Jie Wu, "A Survey on Secure Localization in Wireless Sensor Networks," in *Wireless and Mobile Communications*, CRC Press/Taylor and Francis Group, Boca Raton/London, 2007.

[5] C. Papamanthou, F. P. Preparata, and R. Tamassia, "Algorithms for Location Estimation Based on RSSI Sampling," S. Fekete (Ed.): ALGOSENSORS 2008, LNCS 5389, pp. 72-86, 2008.

[6] Texas Instruments MSP430x1xx Family User's Guide [Online], http://ti.com/msp430.

[7] Mikko Kohvakka, Mauri Kuorilehto, Marko Hannikainen, Timo D. Hamalainen, "Performance Analysis of IEEE 802.15.4 and ZigBee for Large-Scale Wireless Sensor Network Applications," *ACM PE-WASUN*, pp. 48-57, October 2006.

[8] Sinem Coleri Ergen, Carlo Fischione, Dimitri Marandin, Alberto Sangiovanni-Vincentelli, "Duty-Cycle Optimization in Unslotted 802.15.4 Wireless Sensor Networks," *IEEE Transactions on Mobile Computing*, 2009.

[9] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less Low Cost Outdoor Localization For Very Small Devices," *IEEE Personal-Communications, Special Issue on Smart Spaces and Environments*, vol. 7, no. 5, pp. 28-34, October 2000.

[10] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-Free Localization Schemes for Large Scale Sensor Networks," *MobiCom*, 2003.

[11] Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan, "The Cricket Location-Support System," in *Proceedings of MOBICOM 2000*, pp. 32-43, ACM Press, Boston, MA, August 2000.

[12] J. Allred, A. B. Hasan, S. Panichsakul, W. Pisano, P. Gray, J. Huang, R. Han, D. Lawrence, and K. Mohseni, "SensorFlock: an airborne wireless sensor network of micro-air vehicles," in *Proc. of the 5th international conference on Embedded networked sensor systems*, Sydney, Austrailia, pp. 117-129, 2007.

[13] Lance Doherty, Kristofer S. J. Pister, Laurent El Ghaoui, "Convex Position Estimation in Wireless Sensor Networks," in *Proceedings of IEEE Infocom 2001*, Vol. 3, pp 1655-1663, IEEE, IEEE Computer Society Press, April 2001.

[14] Yi Shang, Wheeler Ruml, Ying Zhang, "Localization from mere connectivity," *The Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2003.

[15] Chris Savarese, Jan Rabaey, Koen Langendoen, "Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks," *USENIX Technical Annual Conference*, Monterey, CA, June 2002.

[16] ROY WANT, BILL N. SCHILIT, NORMAN I. ADAMS, RICH GOLD, KARIN PETERSEN, DAVID GOLDBERG, JOHN R. ELLIS, AND MARK WEISER, "An Overview of the PARCTAB Ubiquitous Computing Experiment," *Technical Report CSL-95-1*, Xerox Palo Alto Research Center, March 1995 .

[17] A. Krohn, M. Hazas, and M. Beigl, "Removing Systematic Error in Node Localisation Using Scalable Data Fusion," In *Langendoen*, K.G., Voigt, T. (eds.) EWSN 2007, LNCS, vol.4373, pp.341–356, Springer, Heidelberg, 2007.

[18] Hung-Chi Chu, Rong-Hong Jan, "A GPS-less, outdoor, self-positioning method for wireless sensor networks," *J. Ad Hoc Networks*, vol. 5, no. 5, pp. 547-557, July 2007.

[19] Ahuja, David Cooper, Andrew Hardy and Imran Kanji, "A Bluetooth Based Local Positioning System," in *Proceedings of the ENGG 3100: Design III projects*, University of Guelph, 2007.

[20] J. Hightower, G. Boriello, and R. Want, "An Indoor 3D Location Sensing Technology Based on RF Signal Strength," *University of Washington CSE Report #2000-02-02*, February 2000.

[21] R. Nagpal, "Organizing a Global Coordinate System from Local Information on an Amorphous Computer," *A.I. Memo 1666*, MIT A.I. Laborarory, August 1999.

[22] D. Niculescu, and B. Nath, "DV Based Positioning in Ad Hoc Networks," *Telecommunication Systems*, Baltzer, vol. 1, 2003.

[23] L. Lazos, and R. Poovendran, "SeRLoc: Secure Range-independent Localization for Wireless Sensor Networks," in *Proceedings of the 2004 ACM Workshop on Wireless Security*, pp. 21-30, 2004.

[24] C. Liu and K. Wu, "Sensor Localization with Ring Overlapping Based on Comparison of Received Signal Strength Indicator," *IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, pp. 516-518, Oct. 2004.

[25] CC2420 Chipcon's datasheet, http://www.chipcon.com.

[26] TinyOS [Online], http://www.tinyos.net.

[27] nesC: A Programming Language for Deeply Networked Systems [Online], http://nescc.sourceforge.net/.

[28] Scott Y. Seidel and Theodore S. Rapport, "914 MHz Path Loss Prediction Models for Indoor Wireless Communications in Multifloored Buildings," *IEEE Transactions on Antennas and Propagation*, vol. 40(2), pp. 207-217, February 1992.

[29] Abdalkarim Awad, Thorsten Frunzke, and Falko Dressler, "Adaptive Distance Estimation and Localization in WSN using RSSI Measures," *10th EUROMICRO Conference on Digital System Design-Architectures, Methods and Tools (DSD 2007), IEEE*, pp. 471-478, Germany, August 2007.

[30] Gang Zhou, Tian He, Sudha Krishnamurthy, John A. Stankovic, "Impact of Radio Irregularity on Wireless Sensor Networks," *ACM MobiSys 2004*, pp.125-138, ACM Press, New York, USA, 2004.

[31] Kannan Srinivasan, and Philip Levis, "RSSI is Under Appreciated," *The Third Workshop on Embedded Networked Sensors (EmNets 2006)*, Cambridge, MA, 2006.

[32] ENALAB RSSI Data Set 1: Data from Indoor Testbed Experiment [Online],

http://www.eng.yale.edu/enalab/XYZ/data_set_1.htm.

[33] WANG Fu-Bao, SHI Long, REN Feng-Yuan, "Self-Localization Systems and Algorithms for Wireless Sensor Networks," *Ruan Jian Xue Bao (J. Softw.)*, vol. 16, no. 5, pp. 857-868, May 2005.

[34] R. Want, A. Hopper, V. Falcao, J. Gibbons, "The Active Badge Location System," *ACM Transactions on Information Systems*, vol. 40, no. 1, pp. 91-102, January 1992.

[35] C. S. Raghavendra, Krishna M. Sivalingam, Taieb Znati, "Sensor Networks: A Bridge to the Physical World," in *Wireless Sensor Networks*, 1st ed., Springer, Berlin, Germany, 2006, pp. 3-20.

[36] K. Romer and F. Mattern, "The design space of wireless sensor networks," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 54-61, Dec. 2004.

[37] Akyildiz, I.F., Vuran, M.C., Akan, O.B, and Su, W., "Wireless Sensor Networks: A Survey REVISITED draft," *Computer Networks Journal (Elsevier)*, 2006.

[38] J. Hill, M. Horton, R. Kling, and L. Krishnamurthy, "The Platforms Enabling Wireless Sensor Networks," *Communications of the ACM*, vol. 47, no. 6, pp. 41-46, 2004.

[39] ZigBee Home, [Online], http://www.zigbee.org/About/FAQ/tabid/192/Default.aspx.

[40] J. Beutel, "Location management in wireless sensor networks," *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*, CRC Press, Boca Raton, 2004.

[41] Hongyang Chen, "Research of range-based localization techniques in wireless sensor networks," Master's degree thesis, Southwest Jiaotong University, China, 2006.

[42] J. Pugh, and A. Martinoli, "Relative Localization and Communication Module for Small-Scale Multi-Robot Systems," in *Proc. of the IEEE International Conference on Robotics and Automation*, Miami, Florida, USA, May 15-19, 2006.

[43] J. Cho, J. Choe, K, Song, and Y. Shin, "A routing protocol using relative landmark based on virtual grid in wireless sensor network," in *Proc. of the 23rd international conference on Information Networking*, Chiang Mai, Thailand, pp. 377-379, IEEE Press, NJ, USA, 2009.

[44] N. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller, "Anchor-free distributed localization in sensor networks," *Technical Report TR-892*, MIT LCS, Apr. 2003.

[45] Tian He, Chengdu Huang, Brian M Blum, "Rang-free Localization Schemes in Large

Scale Sensor Networks[C]," in *Proceedings of the 9th annual international conference on Mobile computing and networking (MobiCom)*, San Diego, California, USA: ACM Press, pp. 81-95, 2003.

[46] R. Kakarala, and A. O. Hero, "On Achievable Accuracy in Edge Localization," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 7, pp. 777-781, 1992.

[47] Xu Yan, Shi Jianghong, Wu Xiaofang, "An Enhanced Localization Algorithm Based on RSSI-margin in WSN[J]," *Journal of Xiamen University (Natural Science): 043820479*, 2008.

[48] K. S. Chong and L. Kleeman, "Accurate odometry and error modeling for a mobile robot," Intell. Robot. Res. Centre, Dept. Elect. Comput. Syst. Eng., Monash Univ., Clayton, Australia, Tech. Rep., 1996.

[49] R. B. Langley, "Dilution of Precision," *GPS World*, vol. 10, no. 5, pp.52-59, 1999.

[50] M. L. Sichitiu and V. Ramadurai, "Localization of wireless sensor networks with a mobile beacon," Center for Advances Compuing Communications, North Carolina State Univ., Tech. Rep. TR-03/06, Jul. 2003.

[51] Tmote Sky Brochure, www.moteiv.com.

[52] Tmote Sky Datasheet, www.moteiv.com.

[53] IEEE std. 802.15.4 – 2003:Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low Rate Wireless Personal Area Networks (LR-WPANs), http://standards.ieee.org/getieee802/download/802.15.4-2003.pdf.

[54] M25P80 Datasheet, www.datasheetcatalog.org/datasheet/stmicroelectronics/8495.pdf.

[55] M. Lang, "TinyOS," *Ambient Intelligence Seminar*, Leopold-Franzens-University of Innsbruck, Institute for Informatics, November 2006.

[56] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System architecture directions for networked sensors," in *Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems*, Cambridge, MA, USA, Nov. 2000, pp. 93-104, ACM.

[57] A. Savvides, H. Park, and M. B. Srivastava, "The bits and flops of the *N*-hop multilateration primitive for node localization problems," In *Proc. of the 1st ACM Int'l Workshop on Wireless Sensor Networks and Applications*, Atlanta: ACM Press, pp. 112-

121, 2002.

[58] S. Capkun, M. Hamdi and J.P. Hubaux, "GPS-Free Positioning in Mobile Ad-Hoc Networks," In *Proceedings of HICCSS '01*, Maui, Hawaii, January 2001.

[59] P. Prasithsangaree, P. Krishnamurthy, P.K. Chrysanthis, "On indoor position location with wireless LANs," In *Proc. IEEE Int. Symposium on Personal,* Indoor and Mobile Radio Communications (PIMRC), pp.720–724, 2002.

[60] Gong Silai, "The improvement of DV-Hop in wireless sensor networks," Sciencepaper Online, http://www.paper.edu.cn/en/paper.php?serial_number=200707-249.

[61] K. Yedavalli, B. Krishnamachari, S. Ravula, and B. Srinivasan, "Ecolocation: a sequence based technique for RF localization in wireless sensor networks," In *Proc. Int. Conf. on Information Processing in Sensor Networks (IPSN),* pp.38, 2005.

[62] N. Bulusu, J. Heidemann, D. Estrin, "GPS-less low cost outdoor localization for very small devices," *IEEE Personal Communications Magazine*, vol. 7, no. 5, pp. 28-34, 2000.

Appendix A

HARDWARE AND SOFTWARE

A.1 Introduction of Tmote Sky

Tmote sky is the next-generation mote platform [51] for extremely low power, high data-rate, sensor network applications designed with the dual goal of fault tolerance and development ease. Tmote sky boasts the largest on-chip RAM (Random Access Memory) size (10Kb) of any mote, the first IEEE 802.15.4 radio, and an integrated on-board antenna providing up to 125 meter range. Tmote sky offers a number of integrated peripherals including a 12-bit ADC (Analog-to-Digital Converter) and DAC (Digital-to-Analog Converter), Timer, I^2C (Inter-Integrated Circuit), SPI (Serial Peripheral Interface), and UART (Universal Asynchronous Receiver/Transmitter) bus protocols, and a performance boosting DMA (Direct Memory Access) controller. Tmote sky offers a robust solution with hardware protected external flash (1Mb in size), applications may be wirelessly programmed to the Tmote sky module. In the event of a malfunctioning program, the module loads a protected image from flash. Toward development ease, Tmote sky provides an easy-to-use USB protocol for programming, debugging and data collection.

A.1.1 Key Features [52]

1) 250kbps 2.4 GHz IEEE 802.15.4 Chipcon Wireless Transceiver.
2) Interoperability with other IEEE 802.15.4 devices.
3) 8 MHz Texas Instruments MSP430 microcontrollers (10k RAM, 48k Flash).
4) Integrated ADC, DAC, Supply Voltage Supervisor (SVS), and DMA Controller.
5) Integrated onboard antenna with 50m range indoors / 125m range outdoors.
6) Integrated Humidity, Temperature, and Light sensors.

7) Ultra low current consumption.

8) Fast wakeup from sleep (<6μs).

9) Hardware link-layer encryption and authentication.

10) Programming and data collection via USB.

11) 16-pin expansion support and optional SMA (SubMiniature version A) antenna connector.

12) TinyOS support: mesh networking and communication implementation.

13) Complies with FCC (Federal Communications Commission) Part 15 and Industry Canada regulations.

14) Environmentally friendly complies with RoHS (Restriction of Hazardous Substances) regulations.

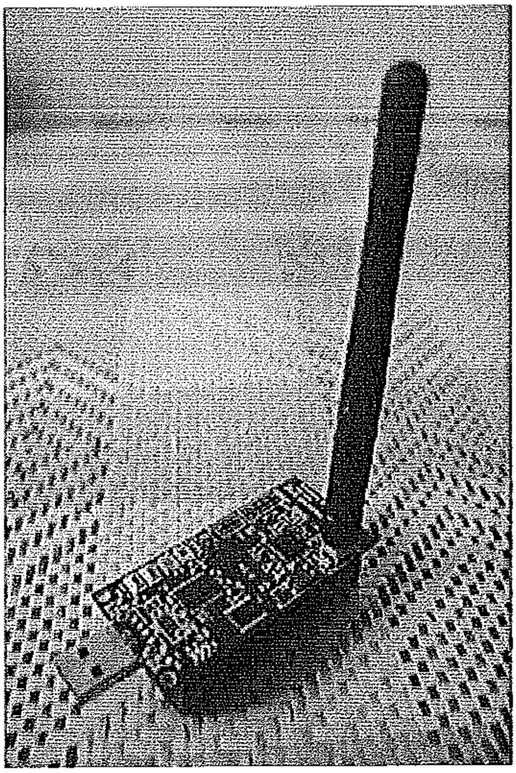


Figure A.1 Unode: a Modified Tmote Sky Node

A.1.2 Module Description

The Tmote Sky module is a low power "mote" with integrated sensors, radio, antenna, microcontroller, and programming capabilities.

A.1.2.1 Power

Tmote Sky may be powered by two AA batteries. The module was designed to fit the two AA battery form factor. AA cells may be used in the operating range of 2.1 to 3.6V DC [52]; however the voltage must be at least 2.7V when programming the microcontroller flash or external flash. If the Tmote Sky module is plugged into the USB port for programming or communication, it will receive power from the host computer. The mote operating voltage when attached to USB is 3V. If Tmote will always be attached to a USB port, no battery pack is necessary. The 16-pin expansion connector can provide power to the module. Any of the battery terminal connections may also provide power to the module. At no point should the input voltage exceed 3.6V which may damage the microcontroller, radio, or other components.

A.1.2.2 Block Diagram

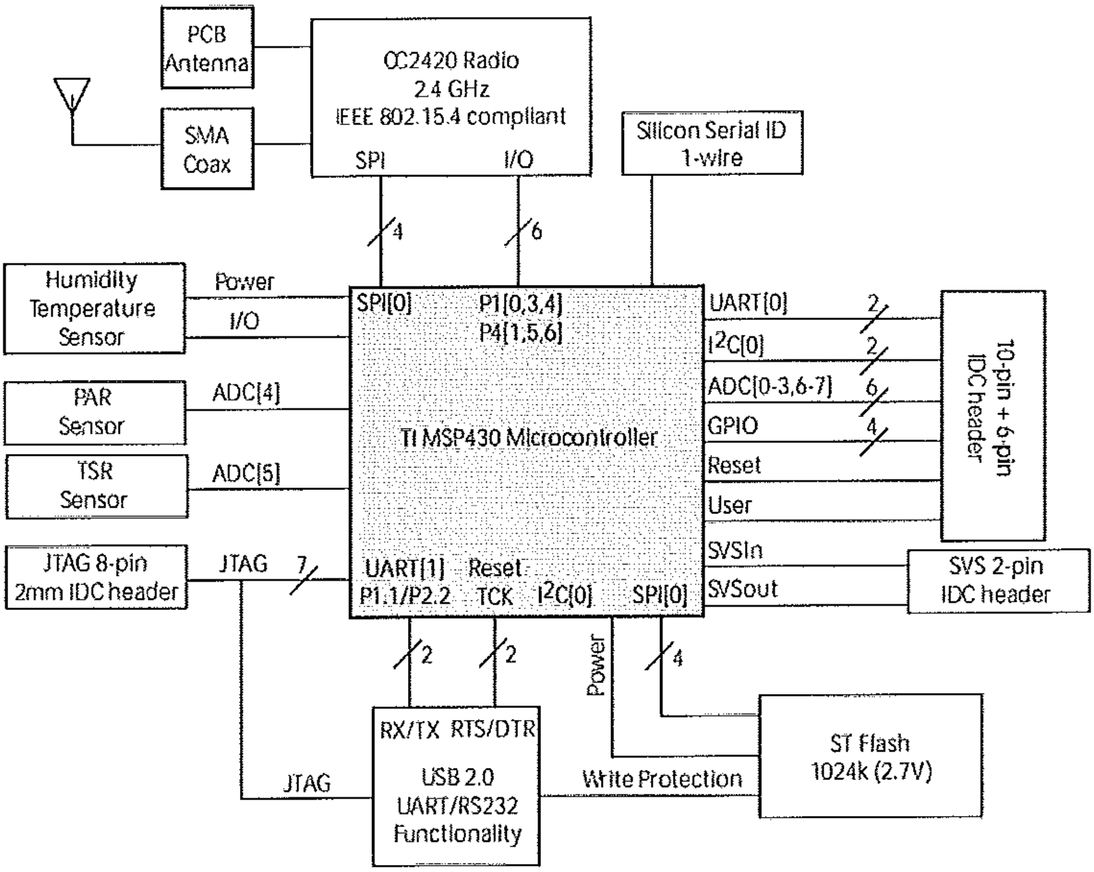


Figure A.2 Functional Block Diagram of the Tmote Sky Module, its Components, and Buses [52]

A.1.2.3 Microprocessor

The low power operation of the Tmote Sky module is due to the ultra low power Texas Instruments MSP430 F1611 microcontroller featuring 10kB of RAM, 48kB of flash, and 128B of information storage [6]. This 16-bit RISC processor features extremely low active and sleep current consumption that permits Tmote to run for years on a single pair of AA batteries [52]. The MSP430 has an internal digitally controlled oscillator (DCO) that may operate up to 8MHz. The DCO may be turned on from sleep mode in 6 μ s [52], however 292ns is typical at room temperature. When the DCO is off, the MSP430 operates off an eternal 32768Hz watch crystal. Although the DCO frequency changes with voltage and temperature, it may be calibrated by using the 32 kHz oscillator.

In addition to the DCO, the MSP430 has 8 external ADC ports and 8 internal ADC ports. The ADC internal ports may be used to read the internal thermistor or monitor the battery voltage. A variety of peripherals are available including SPI, UART, digital I/O ports, Watchdog timer, and Timers with capture and compare functionality. The F1611 also includes a 2-port 12-bit DAC module, Supply Voltage Supervisor, and 3-port DMA controller [6].

A.1.2.4 Radio

Tmote Sky features the Chipcon CC2420 radio for wireless communications. The CC2420 [25] is an IEEE 802.15.4 compliant radio providing the PHY and some MAC functions. With sensitivity exceeding the IEEE 802.15.4 specification and low power operation, the CC2420 provides reliable wireless communication. The CC2420 is highly configurable for many applications with the default radio settings providing IEEE 802.15.4 compliance [25].

The CC2420 is controlled by the TI MSP430 microcontroller through the SPI port and a series of digital I/O lines and interrupts. The radio may be shut off by the microcontroller for low power duty cycled operation.

IEEE 802.15.4 specifies 16 channels within the 2.4 GHz band, in 5 MHz steps, numbered 11 through 26 [25]. The RF frequency of channel k is given by [53]:

$$Fc = 2405 + 5\ (k-11)\ \text{MHz},\ k = 11, 12, 13, \ldots, 26$$

For operation in channel k, the FSCTRL.FREQ register should therefore be set to [25]:

$$\text{FSCTRL.FREQ} = 367 + 5\,(k - 11)$$

A.1.2.5 Antenna

Many techniques, which determine the location of an unknown node using RF, have been proposed. Most of them, however, assume ideal conditions, where the pattern of horizontal radiation is circular and the radio range is the same for each node. However, because of the presence of antenna orientation problems inherent in a real world environment, these localization algorithms only work where the assumptions hold. Antenna orientation is a problem defined as the RSS of the receiver varies as the pair wise antenna orientations of the transmitter and the receiver are changed. In other words, an irregular radiation pattern means that the measured RSS may vary according to the orientation of the transmitting and receiving nodes, although both the transmitter and the receiver are located in a fixed position.

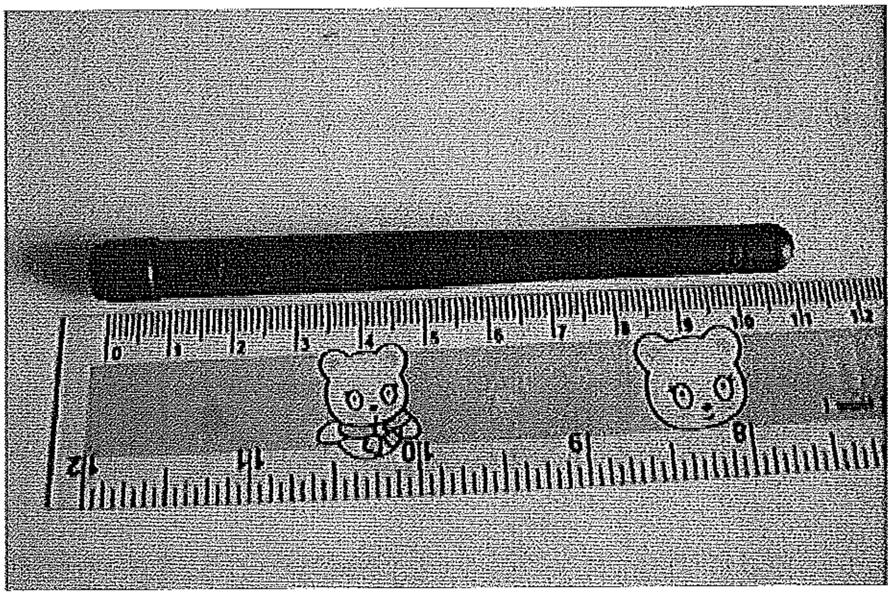


Figure A.3 The Antenna We Used on a Unode Mote

Tmote Sky's internal antenna is an Inverted-F microstrip design protruding from the end of the board away from the battery pack [52]. The Inverted-F antenna is a wire monopole where the top section is folded down to be parallel with the ground plane. Although not a perfect omnidirectional pattern, the antenna may attain 50-meter range indoors and upwards of 125-meter range outdoors.

A.1.2.6 External Flash

Tmote Sky uses the ST M25P80 [54] 40MHz serial code flash for external data and code storage. The flash holds 1024kB of data and is decomposed into 16 segments, each 64kB in size. The flash shares SPI communication lines with the CC2420 transceiver. Care must be taken when reading or writing to flash such that it is interleaved with radio communication, typically implemented as a software arbitration protocol for the SPI bus on the microcontroller.

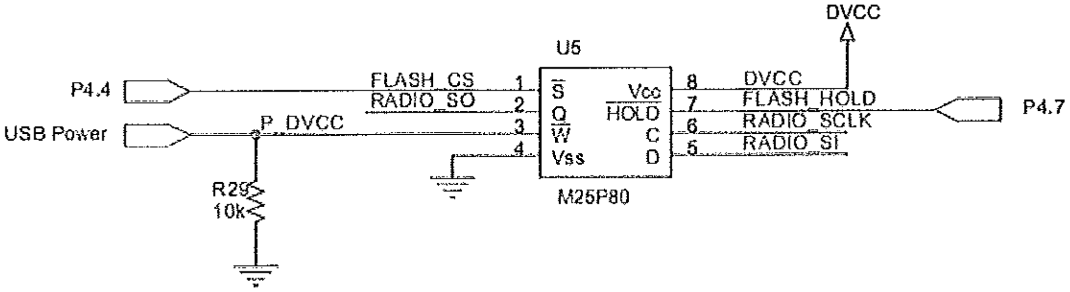


Figure A.4 External serial flash schematic [52]

A.1.2.7 Sensors

The MSP430 microcontroller has internal temperature and voltage sensors that may be used through the microcontroller's ADC interface. The voltage port (input 11) on the 12-bit ADC monitors the output from a voltage divider. See Figure A.5.

A.2 TinyOS: Software Architecture of Wireless Sensor Networks

A critical step towards achieving the vision behind wireless sensor networks is the design of a software architecture that bridges the gap between raw hardware capabilities and a complete system. The demands placed on the software of wireless sensor networks are numerous. It must be efficient in terms of memory, processor, and power so that it meets strict application requirements. It must also be agile enough to allow multiple applications to simultaneously use system resources such as communication, computation and memory. The extreme constraints of these devices make it impractical to use legacy systems [55]. TinyOS is an operating designed explicitly for these reasons [26].

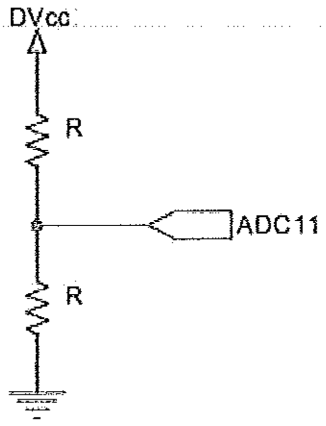


Figure A.5 Voltage Monitoring for Tmote Sky Modules [52]

TinyOS is an open-source operating system designed for wireless embedded sensor networks [26]. It features a component-based architecture which enables rapid innovation and implementation while minimizing code size as required by the severe memory constraints inherent in sensor networks. TinyOS's component library includes network protocols, distributed services, sensor drivers, and data acquisition tools – all of which can be used as-is or be further refined for a custom application. TinyOS's event-driven execution model enables fine-grained power management yet allows the scheduling flexibility made necessary by the unpredictable nature of wireless communication and physical world interfaces.

TinyOS draws strongly from previous architectural work on lightweight thread support and efficient network interfaces. Included in the TinyOS system architecture is an Active Messages communication system. There is a fundamental fit between the event based nature of network sensor applications and the event based primitives of the Active Messages communication model.

TinyOS has been ported to over a dozen platforms and numerous sensor boards. A wide community uses it in simulation to develop and test various algorithms and protocols. New releases see over 10,000 downloads. Over 500 research groups and companies are using TinyOS on the Berkeley/Crossbow Motes, such as Tmote Sky [26]. Numerous groups are actively contributing code to the sourceforge site and working together to establish standard, interoperable network services built from a base of direct experience and honed through competitive analysis in an open environment.

A.2.1 TinyOS Execution Model

To provide the extreme levels of operating efficiency required in wireless sensor networks, TinyOS uses event based execution. The event model allows for high levels of concurrency to be handled in a very small amount of space.

A.2.1.1 Event Based Programming

In an event based system, a single execution context is shared between unrelated processing tasks. In TinyOS, each system module is designed to operate by continually responding to incoming events. When an event arrives, it brings the required execution context with it. When the event processing is completed, it is returned back to the system.

In addition to efficient CPU allocation, event-based design results in low power operation. A key to limiting power consumption is to identify when there is no useful work to be performed and to enter an ultra-low power state. Event-based systems force applications to implicitly declare when they are finished using the CPU. In TinyOS all tasks associated with an event are handled rapidly after each event is signaled. When an event and all tasks are fully processed, unused CPU cycles are spent in the sleep state as opposed to actively looking for the next interesting event. Eliminating blocking and polling prevents unnecessary CPU activity.

A.2.1.2 Tasks

A limiting factor of an event based program is that long-running calculations can disrupt the execution of other time critical subsystems. If an event were to never complete, all other system functions would halt. To allow for long running computation, TinyOS provides an execution mechanism called tasks. A task is an execution context that runs to completion in the background without interfering with other system events. Tasks can be scheduled at any time but will not execute until current pending events are completed. Additionally, tasks can be interrupted by low-level system events. Tasks allow long running computation to occur in the background while system event processing continues.

Currently task scheduling is performed using a simple FIFO scheduling queue. While it is possible to efficiently implement priority scheduling for tasks, it is unusual to have multiple outstanding tasks. A FIFO queue has proven adequate for all application scenarios we have attempted to date.

A.2.1.3 Atomicity

In addition to providing a mechanism for long-running computation, the TinyOS task primitive also provides an elegant mechanism for creating mutually exclusive sections of code. In interrupt-based programming, data race conditions create bugs that are difficult to detect. In TinyOS, code that is executed inside of a task is guaranteed to run to completion without being interrupted by other tasks. This guarantee means that all tasks are atomic with respect to other tasks and eliminates the possibility of data race conditions between tasks.

Low-level system components must deal with the complexities associated with reentrant, interrupt based code in order to meet their strict real-time requirements. Normally, only simple operations are performed at the interrupt level to integrate data with ongoing computation. Applications can use tasks to guarantee that all data modification occurs atomically when viewed from the context of other tasks.

A.2.2 TinyOS Component Model

In addition to using the highly efficient event-based execution, TinyOS also includes a specially designed component model targeting highly efficient modularity and easy composition. An efficient component model is essential for embedded systems to increase reliability without sacrificing performance. The component model allows an application developer to be able to easily combine independent components into an application specific configuration.

In TinyOS, each module is defined by the set of commands and events that makes up its interface. In turn, a complete system specification is a listing of the components to include plus a specification for the interconnection between components. The TinyOS component has four interrelated parts: a set of command handlers, a set of event handlers, an encapsulated private data frame, and a bundle of simple tasks. Tasks, commands, and event handlers execute in the context of the frame and operate on its state. To facilitate modularity, each component also declares the commands it uses and the events it signals. These declarations are used to facilitate the composition process.

In TinyOS, storage frames are statically allocated to allow the memory requirements of the complete application to be determined at compile time. The frame is a specialized C Structure that is statically allocated and directly accessible only to the component. While TinyOS does not have memory protection, variables cannot be directly accessed from outside of a component. In addition to allowing the calculation of maximum memory requirements, pre-allocation of frames prevents the overhead associated with dynamic allocation and avoids pointer related errors. This savings manifests itself in many ways, including execution time savings because variable locations can be statically compiled into the program instead of accessing state via pointers.

In TinyOS, commands are non-blocking requests made to lower level components. Typically, a command will deposit request parameters into its local frame and conditionally post a task for later execution. It may also invoke lower commands, but it must not wait for long or indeterminate latency actions to take place. A command must provide feedback to its caller by returning status indicating whether it was successful or not, e.g., buffer overrun. Event handlers are invoked to deal with hardware events, either directly or indirectly. The lowest level components have handlers that are connected directly to hardware interrupts, which may be external interrupts, timer events, or counter events. An event handler can deposit information into its frame, post tasks, signal higher level events or call lower level commands. A hardware event triggers a fountain of processing that goes upward through events and can bend downward through commands [56]. In order to avoid cycles in the command/event chain, commands cannot signal events. Both commands and events are intended to perform a small, fixed amount of work, which occurs within the context of their component's state.

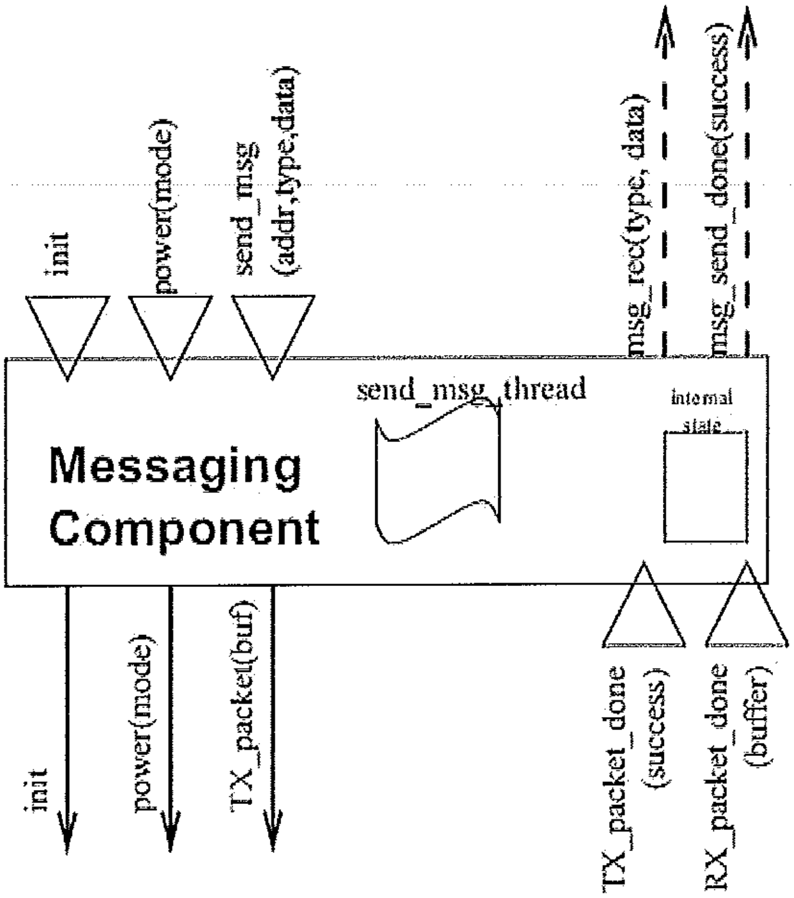


Figure A.6 AM Messaging Component Graphical Representation [56]

Tasks perform the primary work in a TinyOS application. They are atomic with respect to other tasks and run to completion, though they can be preempted by events. Tasks can call lower level commands, signal higher level events, and schedule other tasks within a component. The run-to-completion semantics of tasks make it possible to allocate a single stack that is assigned to the currently executing task. This is essential in memory constrained systems.

Tasks simulate concurrency within each component, since they execute asynchronously with respect to events. However, tasks cannot block or spin wait or they will prevent progress in other components. While events and commands approximate instantaneous state transitions, task bundles provide a way to incorporate arbitrary computation into the event driven model.

A typical component including a frame, event handlers, commands and tasks for a message handling component is pictured in Figure A.6. Like most components, it exports commands for initialization and power management. Additionally, it has a command for initiating a message transmission, and signals events on the completion of a transmission or the arrival of a message. In order to perform its function, the messaging component issues commands to a packet level component and handles two types of events: one that indicates a message has been transmitted and one that signals that a message has been received.

A.3 Network Embedded Systems C Language

NesC (Network embedded systems C) is a component-based, event-driven programming language used to build applications for the TinyOS platform. The basic concepts behind NesC are [27]:

1) Separation of construction and composition: programs are built out of **components**, which are assembled ("wired") to form whole programs, which is presented in figure A.7. Components have internal concurrency in the form of **tasks**. Threads of control may pass into a component through its interfaces. These threads are rooted either in a task or a hardware interrupt.

```
configuration BlinkAppC
{
}
implementation
{
  components MainC, BlinkC, LedsC;
  components new TimerMilliC() as Timer0;
  components new TimerMilliC() as Timer1;
  components new TimerMilliC() as Timer2;


  BlinkC -> MainC.Boot;

  BlinkC.Timer0 -> Timer0;
  BlinkC.Timer1 -> Timer1;
  BlinkC.Timer2 -> Timer2;
  BlinkC.Leds -> LedsC;
}
```

Figure A.7 NesC Application Configuration File that Wires together the Blink Application

2) Specification of component behavior in terms of set of **interfaces**. Interfaces may be provided or used by components. The provided interfaces are intended to represent the

functionality that the component provides to its user; the used interfaces represent the functionality the component needs to perform its job.

```
module AMStandard
{
  provides {
    interface StdControl as Control;
    interface CommControl;

    // The interface are as parameterised by the active message id
    interface SendMsg[uint8_t id];
    interface ReceiveMsg[uint8_t id];

    // How many packets were received in the past second
    command uint16_t activity();
  }

  uses {
    // signaled after every send completion for components which wish to
    // retry failed sends
    event result_t sendDone();

    interface StdControl as UARTControl;
    interface BareSendMsg as UARTSend;
    interface ReceiveMsg as UARTReceive;

    interface StdControl as RadioControl;
    interface BareSendMsg as RadioSend;
    interface ReceiveMsg as RadioReceive;
    interface Leds;
    //interface Timer as ActivityTimer;
  }
}
```

Figure A.8 NesC component File Example

3) Interfaces are bidirectional: they specify a set of functions to be implemented by the interface's provider **(commands)** and a set to be implemented by the interface's user **(events)**. This allows a single interface to represent a complex interaction between components (e.g., registration of interest in some event, followed by a callback when that event happens). This is critical because all lengthy commands in TinyOS (e.g. send packet) are non-blocking; their completion is signaled through an event (send done). By specifying interfaces, a component cannot call the send command unless it provides an implementation of the sendDone event.

4) Components are statically linked to each other via their interfaces. This increases runtime efficiency, encourages rubust design, and allows for better static analysis of programs.

5) NesC is designed under the expectation that code will be generated by whole-program compilers. This should also allow for better code generation and analysis.

As the critical part of NesC, the component can be represented as a bundle of tasks, a block of state (component frame), a set of commands (upside-down triangles), a set of handlers (triangles), solid downward arcs for commands they use, and dashed upward arcs for events they signal. Figure A.8 shows how all of these elements are explicit in the component code.

VITAE

Name Mr. Cao Ning

Student ID 4910120107

Educational Attainment

Degree	Name of Institution	Year of Graduation
Bachelor of Engineering	Jiangxi University of Science and Technology	2006

List of Publication and Proceedings

[1] N. Cao, and W. Suntiamorntut, "RSSI & Location Analysis in Wireless Sensor Networks," *Proceedings of the ICESIT International Conference (ICESIT 2010)*, Chiang Mai, Thailand, Feb. 2010.