Självständigt arbete på grundnivå

Independent degree project - first cycle

Datateknik
Computer Engineering

Title
Designing a centralized, self-organized wireless sensor network

First name Last name
Shixian Wen

Mittuniversitetet
MID SWEDEN UNIVERSITY

Campus Härnösand Universitetsbacken 1, SE-871 88. **Campus Sundsvall** Holmgatan 10, SE-851 70 Sundsvall.
Campus Östersund Kunskapens väg 8, SE-831 25 Östersund.
Phone: +46 (0)771 97 50 00, Fax: +46 (0)771 97 50 01.

Abstract

The objective of this study is to design a wireless sensor network which can automatically build a routing tree from motes to sink after randomly assigning a topology, provide useful information collected from the network such as link quality information, and all of the motes can act properly according to the commands from sink after our user centralized control system processes the data that my system collected. This report mainly focuses on the structure of this system, like how we estimate the channel, how we design a TDMA on motes, how system automatically build a routing tree and so on. In this report, this system provides Felix Dobslaw's Weighted Shortest Path new routing algorithms program using ETX at the sink with link quality information based on packet deliver rate and uses an interface to deliver the TDMA and routing schedule from PC to distributed wireless sensor network and test his Weighted Shortest Path new routing algorithm using ETX as required. The results show the packet deliver rate of 21 different topologies with 3 different user demands from 0.9 to 0.99999. The Conclusions discuss the main contribution of this project and the extendability, user-friendly and adaptability of this system.

Keywords: TinyOS, Link Estimation, TDMA, CSMA/CA, Self-organized

Acknowledgements

First of all, I want to thank my supervisor Prof.Tingting Zhang for guiding me pass a lot of obstacles and difficulties and sacrificing her time to listen my report every week. Then, I want to thank Felix Dobslaw for giving me some advices in designing the interface in my system and providing me with his Weighted Shortest Path using ETX new routing algorithms and letting me to test my system by using his program.

Table of Contents

Terminology

Acronyms/Abbreviations

ACK	Acknowledge. Verification of a correctly transferred message
AWGN	Additive White Gaussian Noise.
TDMA	Time division multiple access
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
WSN	Wireless Sensor Network
RX	Receiver
TX	Transmitter
PDR	Packet Deliver Rate
DP	Duplicate Packet
MAC	Medium Access Control
TPR	Total Received Packets

Mathematical notation

p_{RX}	the received power
p_{TX}	the transmitted power
G_{RX}	antenna gain of the receiver
G_{TX}	antenna gain of the transmitter
h_{TX}	the high of transmitter
h_{RX}	the high of receiver
d	distance between transmitter to current point
v_{max}	Doppler shift

1 Introduction

1.1 Background and problem motivation

Wireless sensor network is playing an ever increasing role in our daily life, providing people with an interface to their surrounding physical environment, such as area monitoring, natural area monitoring and so on. However, due to harsh environment conditions, it's common for sensors cannot get access to a constant supply of power. Thus, power, typically stored in battery in motes, becomes a really precious resource. The biggest part of the power consuming is used for communications between nodes. As a result, efficient and robust routing that not only guarantees the packets of motes being delivered to the sink, but also saves the energy by avoiding re-transmission, attracts a tremendous attention from researchers. Furthermore, because of the goal of companies to produce a low cost tiny sensors, the memory and process speed become another precious resources. As a consequence, how to implement a desired system on motes becomes another hot issue in research field. Otherwise, you will get some unexpected results caused by the lack of kernel protection and process speed in motes system. Thus, a centralized, self-organized wireless sensor network, which can put the calculation burden on the PC side that is connected to the sink and command motes in the wireless sensor network to perform as it's required such as a robust routing, is crucial to improve the system efficiency and improve overall system life expectation.

1.2 Overall aim

The project overall aim is to implement a centralized, self-organized wireless sensor network which can finish following 4 tasks:

1. building an initial routing tree from every mote to the sink and from sink to every mote in the wireless sensor network.

2. providing useful and crucial information such as link quality information from a mote to its neighbors to the sink when there is no initial schedule being made in a randomly generated topology.

3. distributing the commands which tell the motes to do some assigned behaviors at expected time instant from the sink to the whole networks when there is no initial schedule being made in a randomly generated topology.

4. Design a user-friendly interface that can hide physical implementation from my client's centralized control system.

In this report, my system provides Felix Dobslaw's program with a link quality matrix based on Packet Deliver rate and uses Felix Dobslaw's Weighted Shortest Path using ETX new Routing algorithm program to schedule the routing and

TDMA frame structure using a randomly generated topology in the wireless sensor network.

1.3 Scope

The study has its focus on building a centralized, self-organized wireless sensor network. Constrained by the Tossim emulator provided by TinyOS, the channel model of this project do not take any small scale fading into account and it's only valid in static and low mobility condition. The Felix Dobslaw's program also put a constrain that there is no orphan mote in the network.

1.4 Concrete and verifiable goals

1. In order to implement a centralized, self-organized wireless sensor network, finding a suitable development tools to simulate a real environment and the system that designed can be adapted to different scenario and applications is the preliminary.

2. Building a channel model that satisfies the real industrial situations of wireless sensor network and also supported by the tools which can be used to do the simulation. Providing a matrix in which elements indicate the channel quality based on packet deliver rate. Making a connection between the packet deliver rate and the channel model.

3. Constrained by limited power resources, process speed of motes and the payload length of packet, designing suitable data structure for data processing and data storing is fundamental.

4. Because there is no routing trees at the beginning when all of the node are randomly distributed, building an initial routing tree to forward link quality information or other useful data to the sink from the network and distribute the scheduling packets to the network from the sink is really crucial.

5. Designing a logic clock in the mote and synchronizing all the logic clocks in the networks to implement TDMA scheme successfully. Design an TDMA scheme that it stores the time slots of its own and its neighbors slots and knows if there is still times left for the sensor to transmit another packet in the remaining slot time. Also, the register is calculated by 2 power n(like a register is 1 byte represent 1024) and cannot create an exact logic time we want. e.g. using a 1 byte register which counts to 1024 to represent 1000ms, there would be 24 system clocks error.

6. Solving the packet collision problem when motes want to estimate the channel and forward their link quality information or other sensor data to the sink.

7. Designing an interface which can hide physical details from client's program between TinyOS at the mote side and java application at the personal computer side.

1.5 Outline

The structure of this report is stated below: Chapter 1 mainly introduces the background of wireless sensor network and describes the motivation of this project. Chapter 2 lists some major theories that are being used in this project which gives readers the prerequisite knowledge to understand this project. Chapter 3 mainly focuses on the methodology of solving the concrete goals stated in the chapter 1. Chapter 4 gives details of implementing this centralized, self-organized wireless sensor network. Chapter 5 presents the test results of Felix Dobslaw's new routing algorithm which is tested on my system. Chapter 6 summarizes this system, shows an incredible flexibility, user-friendly and extendability of my program and puts forward future works .

1.6 Contributions

All of the codes and tests in this project are done by Shixian Wen. The centralized control system is based on the work of Felix Dobslaw's new Routing Algorithm.

2 Theory

2.1 Wireless sensor network

Originated from research project-distributed sensor networks program at MIT Lincoln lab in the mid 1990s, wireless sensor network has nowadays being widely used not only in the labs, but also in the commercial practice. Although there are a lot of different applications of wireless sensor network, a typical similar structure has been adopted. A wireless sensor network is made up with multiple wireless sensor nodes called motes, which usually used to sense the physical environment, process the data collected from the physical environment, and communicate with other nodes via wireless communication, shown in figure 2.1. According to [1], it puts forward a simple and low cost communication networks that are composed of source-limited devices, mainly constrained by limited power and relaxed throughput requirements. The mainly task of a Wireless Sensor networks is the ability to form a distributed sensing system typically including following three features:

1. collecting data from sensor at different geographic location and time.

2. transferring the collected data to a remote gateway sensor node through network.

3. deduct information by data fusion, which obtained from different locations.

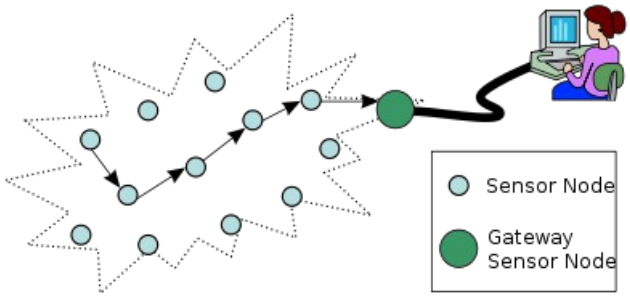

Figure 2.1 An overview on the typical network architecture of wireless sensor network

2.2 Time division multiple access(TDMA) Scheduling

In order to meet guaranteed access or bounded latency of industrial standards, scheduled protocols, such as TDMA[5], has been introduced to satisfy the real-time transmission requirement. TDMA is a digital transmission technology that divides the transmission time into different small time slots and assign it to different users. Thus, it enables a number of users to access a single shared medium without collision. This strategy not only allows multiple nodes to share the transmission medium almost at its full potential, but also saves the power of nodes- nodes will sleep at time slots except their assigned time slots and their

neighbor's time slots. Usually, as shown in figure 2.2, the time slots are combined into frame. Synchronization of clocks between motes and all motes in the network know the start of a time slot is crucial for ensuring a successful TDMA mechanism.

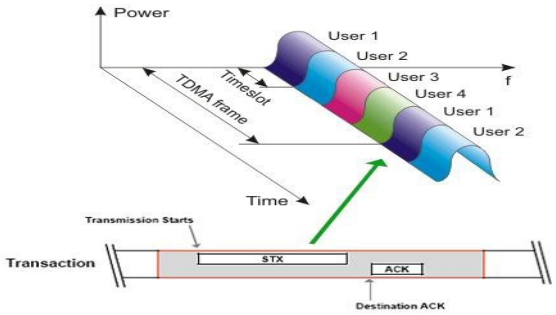

Figure 2.2 TDMA

2.3 Channel model

In telecommunication, a communication channel refers to physical transmission medium such as wire, used to convey an information signal from one location to another location. In wireless sensor network, it uses a wireless channels instead of wired channels differed from by multipath propagation, i.e. the existence of a lot of propagation paths where the signal can be scattered, reflected or diffracted along the way from transmitter to receiver. We can combine all of these physical phenomena to understand the channel. However, another way to look at the channel is that we do not care about how the channel looks like in a specific location. In another word, how is influenced by the multipath propagation. What we interested is the probability that a channel parameters gain certain value. In our case, we mainly interested in the received power or field strength.

2.3.1 The d^{-4} power law

The d^{-4} power law in wireless communications says that the received signal power is inversely proportional to the fourth power of the distance between TX and RX. The equation is:

$$p_{RX}(d) \approx P_{TX} G_{TX} G_{RX} \left(\frac{h_{TX} h_{RX}}{d^2}\right)^2$$

where p_{RX} p_{TX} G_{RX} G_{TX} h_{TX} h_{RX} d are the received power, the transmitted power, antenna gain of the receiver, antenna gain of the transmitter , the high of transmitter, the high of receiver and the distance between transmitter to current point.

For link budget calculation we assume that the power decays d^{-2} until break point $d_{breakpoint}$ and from there decay d^{-n} we can rewrite our equation in another form shown in figure 2.3:

$$P_{RX}(d) = P_{RX}(1m) - 20\log(d_{breakpoint}\,m) - n\,10\log(d/d_{break})$$

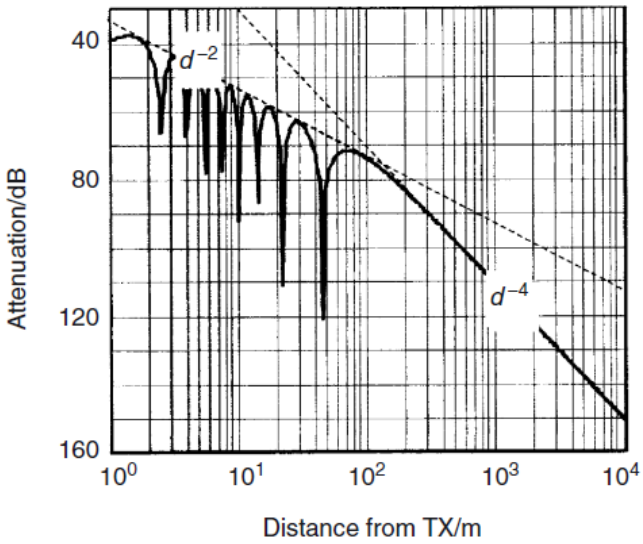

Figure 2.3 propagation over an ideally reflecting ground

2.3.2 Doppler shift

Doppler shift is the change in frequency of a wave for an observer moving relative to its source. The maximum Doppler shift v_{max} can range between 1Hz and 1kHz. The Doppler shift is calculated as

$$v = \frac{-v}{\lambda}\cos(\gamma) = -f_c\frac{v}{c_0}\cos(\gamma) = -v_{max}\cos(\gamma)$$

2.3.3 Gaussian noise

Gaussian noise has a probability distribution function equal to normal distribution. In another word, the value of the noise is normal distributed. The Probability density function P as a function of z is like this.

$$P_G(z) = \frac{1}{(\sigma\sqrt{2\pi})}e^{\frac{-(z-\mu)^2}{(2\sigma^2)}}$$

where z is grey level, μ is mean value and σ is the standard deviation

2.3.4 Noise floor

Noise floor is a parameter to measure the signal strength created from the sum of all other sources of noise like thermal noise, blackbody, cosmic and unwanted signals within a measurement system.

2.3.5 Fading

The deviation of the attenuation that affects the signal over certain propagation media is called fading. Fading can be categorized into two categories: small scale fading and large scale fading. Small scale fading can be defined by the changing of the total signal amplitude due to interference of the different multi-path propagation components. Large scale fading is caused by shadowing, you have to move over large distances to move from the light to the dark zone shown in figure2.4.

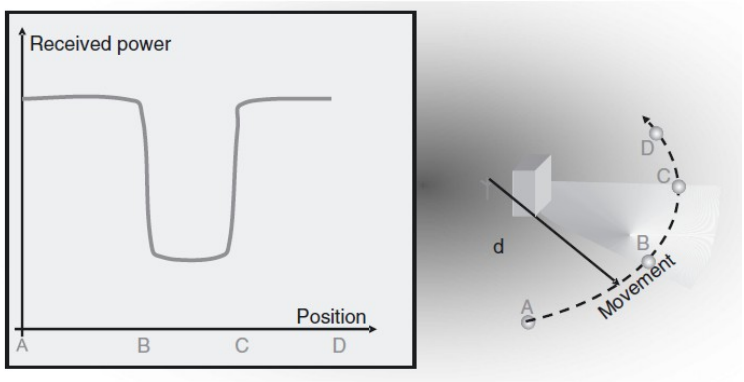

Figure2.4 large scale fading

Fast fading and Slow fading: When the coherence time of the channel is large relative to the delay constraint of the channel, we call it slow fading channel. Otherwise, we call it fast fading channel. In slow fading channel, the amplitude and phase change imposed by the channel can be considered roughly constant over the period of use. However, in the fast fading channel, the amplitude and the phase change imposed by the channel varies vary fast over the period of use.

2.4 Distributed system

2.4.1 Logic clock

Wireless sensor network is a distributed system that has no physically synchronous global clock, refer to [6-7]. So it needs a logic clock for capturing chronological and causal relationship in order to make TDMA mechanism work. In the real implementation, I let a 32bit memory to hold the a logic clock. Every time when the clock fired event caused by physical clock register comes, logic clock plus one.

2.4.2 Symmetric mode of the network time protocol

In order to synchronize all the logic clocks in the wireless sensor network, we can use Symmetric mode of the network time protocol shown in figure 2.5

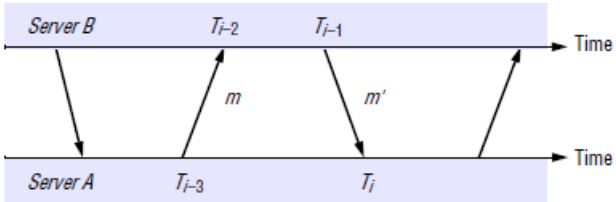

Figure 2.5 message changes between two NTP peers

The estimated offset between these two is $o_i=(T_{i-2}-T_{i-3}+T_{i-1}-T_i)/2$

2.5 MAC Protocols for Wireless Sensor Network

2.5.1 Fundamentals of Wireless MAC Protocols

MAC protocols of the data link layer play an important role to determine the times during which a number of contending nodes being authorized to access to a shared wireless medium. Traditionally, there are three kind of MAC protocols : fixed assignment protocols, demand assignment protocols and random access protocols shown in in [2].

2.5.2 CSMA/CA Protocols

Carrier Sense Multiple Access with Collision Avoidance(CSMA/CA), put forwarded by IEEE 802.11 is a widely accepted protocol, adopting single channel mode shown in [3-4]. When a node wants to transmit a packet, before transmitting it, the node should sense the channel to determine whether another node is using the channel. If the channel is idle for a period of time greater than distributed interface space, the node will be allowed to transmit. Otherwise, If the channel is busy, the node will wait until others finished transmission and choose a random backoff interval.

2.6 TinyOS and its extension

2.6.1 TinyOS

TinyOS is a lightweight, open source operating system which is used to design wireless low-power sensors, having a very aggressive mechanism for saving power. It provide users with a lot of important services and abstractions, such as timers, storage, communication and so on. TinyOS applications are written in nesc language, which gives advantage of reducing RAM and code size, optimizing compiled code, and helping prevent low-level bugs like race condition. Furthermore, TinyOS provides a component model, which can be used to compose into larger abstractions. In this component model, you can write reusable pieces of code in it like the object representation in java. There are two kinds of components model modules and configuration. Modules are components that implement and call functions. Configurations wire or in another word, connect components into larger abstractions.

Unlike personal computer system, there is no memory protection, the stack can easily overflow onto the heap or data shown in figure 2.6, refer to [8]

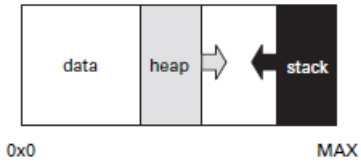

Figure 2.6 typical memory layout on a micro-controller

Since the TinyOS uses split-phase operation which does not provide threads. when a TinyOS system is sleep, the module variables represent the entire software state of the system. Hence, we should pay great attention to the pointer and stacks when we code the program that could corrupt the system memory and cause some unexpected results.

2.6.2 TOSSIM and TOSSIM-live extension

TOSSIM is a discrete event simulator to simulate entire TinyOS applications refer to[9]. Users can run their motes on PC instead of using real nodes which allows them to test, debug and analyze their algorithms in a repeatable and controlled environment. While, TOSSIM-live is an extension of TOSSIM refer to [10], allowing users to forward their serial messages (PC to motes) to the virtual motes.

2.7 Contiki

Contiki, first created by Adam Dunkels, is an open source operating system for limited memory and networked system refer to [11]. Contiki is designed for running on the device that severely suffer from the limited size of memory and power constrained and can support 3 different network mechanisms: TCP/IP stacks, the uIPv6 stack and the Rime Stack.

2.8 The Difference between TinyOS and Contiki

The TinyOS has a bigger community and the better documentation and open source code. Refer to [12-13], TinyOS has better abstractions of platforms,sensorboards, chips and so on. While, when you use Contiki, building system can be invoked from simulator. TinyOS can automatically optimize your resulting system by using preprocessed source that means it generates faster code leading to few CPU cycles which save the energy in the wireless sensor nodes. However, Contiki has not built anything yet on the actual hardware.

3 Methodology

3.1 Find suitable development tools

In order to achieve goal (1) to implement a centralized, self-organized wireless sensor network, as discussed in Chapter 2. There are many languages to implement such as TinyOS, LiteOS, Contiki and so on. There many pros and cons of these languages discussed in Chapter 2.8 in details which leads me to choose TinyOS language which not only provides me an opportunity to implement this system on the Micaz motes, but also make the experiment variables controllable and experiment repeatable. In the implementation, I also used TOSSIM and TOSSIM-LIVE, java, TinyOS API and the jar file wrote by Felix Dobslaw.

3.2 Build channel model and provide quality matrix

To fulfill goal (2) to provide a matrix in which elements represent the packet deliver rate from each sensor to its neighbors and make a channel model that can simulate the real industrial environment. I refer to [14] and [15] and build a channel that uses the d^{-4} power law to simulate a large scale fading and calculate the received power and add a Gaussian noise into channel and use covariance matrix to simulate an asymmetric channel between motes pairs by considering shadowing effect. In TOSSIM simulation, I use the noise and interference collected from the library of Stanford referring to [16]. Then, I set the parameters which is suitable for a static and low mobility building aisle environment.

3.3 Design suitable data structure and build initial routing tree

In order to make goal (3) and (4) possible, I design an algorithm to let motes automatically build an initial topology by using the node level and the link quality information in order to deliver the first link estimation information and distribute the first schedule information. I mainly use link table and pool data structure to save the link quality and routing information and defines 5 different message types . Also, I have quantized the packet deliver rate information from 0-255 which can be represented by 1 byte because of limited payload length of packets.

3.4 Design TDMA mechanism, synchronization and logic clock

Let us look at goal (5) which wants to design a logic clock and synchronize logic clocks between motes and solve the register problem. First, TOSSIM guarantees all the motes in the network having synchronized logic clock so that I do not need to use network time protocol to make synchronization between nodes. However, I made a mechanism to let mote check whether their actual start point of the time slot is exactly equal to the theoretical value and correct the difference according to the value they get. Third, I successfully made the motes knowing that if the remaining time slot is enough to send another packet.

3.5 Solve packets collision

As for the goal (6), because of no schedule of motes at the beginning, I implemented an CSMA/CA protocol for nodes to deliver and relay the link quality information packet to the sink and used TDMA mechanism to let motes to estimate the channel sequentially.

3.6 Design an interface

In order to achieve goal (7), I designed a data structure used for programs to communicate which hides the physical implementation by using mig(Message Interface generator) automatically build methods and classes from the data structure. The interface can create a mark file which being used to notice their counterpart to take some specified action.

4 Implementation

This designed complete system shown in figure 4.1 is to satisfy the aim of flexibility of modules where module has its own function and can be encapsulated for other uses.

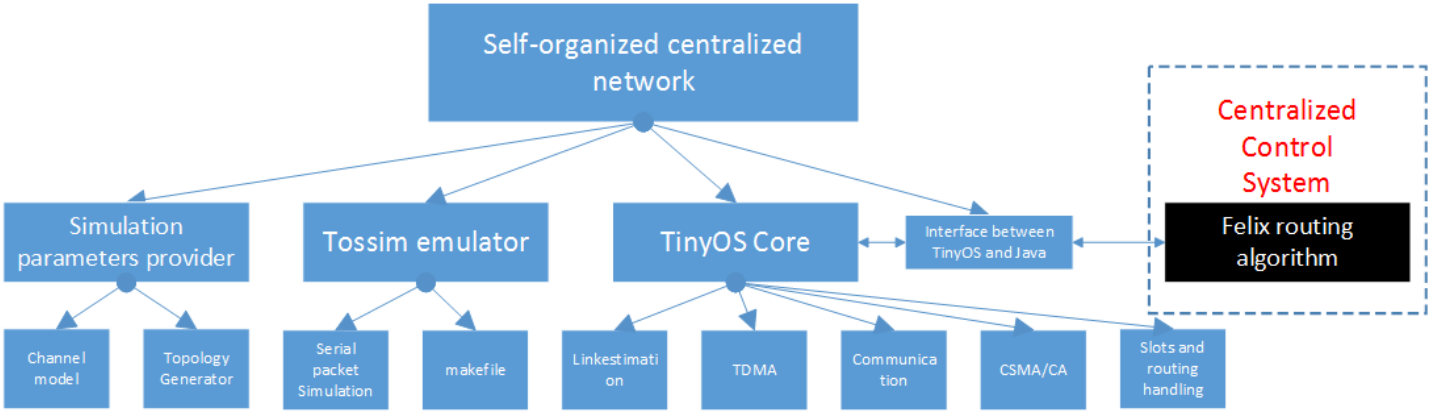

Figure 4.1 software layout

4.1 TinyOS Core

In this sub chapter, we will see the most important module TinyOS Core in detail to understand how I implemented a self-organized centralized network based on Tinyos.

TinyOS Core is consisted of 5 sub modules- Link estimation, TDMA, communication, CSMA/CA, and Slots and routing handling. Let's look at these 5 modules individually.

4.1.1 Link estimation

Link estimation module is mainly used for estimating the channel quality from a sensor to its neighbors by taking advantage of beacon message. This block can randomly get a slot at the beginning according to the size of the network, calculate the remaining time slot and transmit as many as beacon packets in the available slot. The procedure for motes to send beacon message is shown in figure 4.2

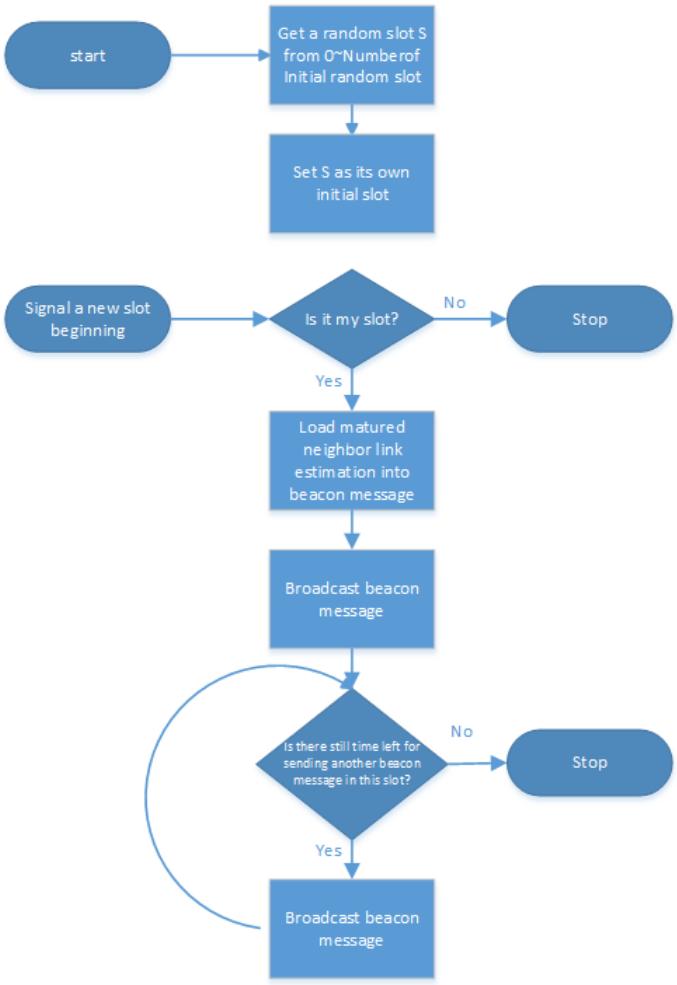

Figure 4.2 procedure to send beacon messages

When a mote receive beacon message they will use beacon message to calculate the channel quality according the sequence number stored in the header of beacon message. The structure of the beacon message is shown in figure 4.3.

| Header | Message_type | node_level | footer 1| footer 2| .... | footer n |

Header:

| flags | sequence number |

Footer:

| address | outbound link quality |

Figure 4.3 The structure of beacon message

When a mote receives a beacon message, it will update the link quality information according to the sequence number from which we can calculate the number of lost packets and the number of received packets and the inbound link quality in the beacon message shown in figure 4.4. The link quality information is based on the time average value which you can change the weight according to different situations and limited by the maximum payload of packet and the memory, I quantized the packet deliver rate from 0-255 which can be represented by 1 byte.

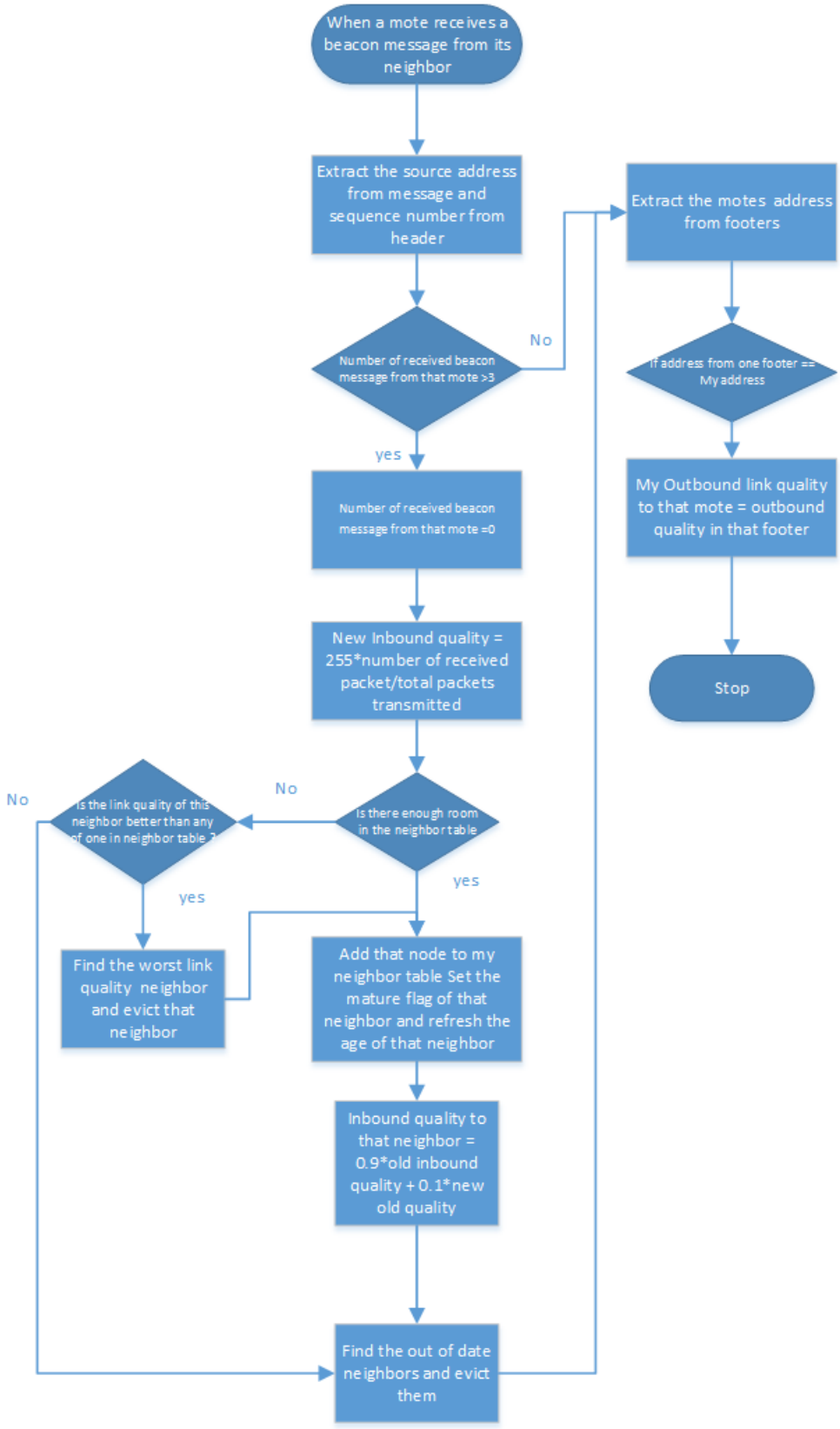

Figure 4.4 procedure to estimate channel by using beacon message

4.1.2 TDMA

The structure of software Wiring of TDMA is shown in figure 4.5

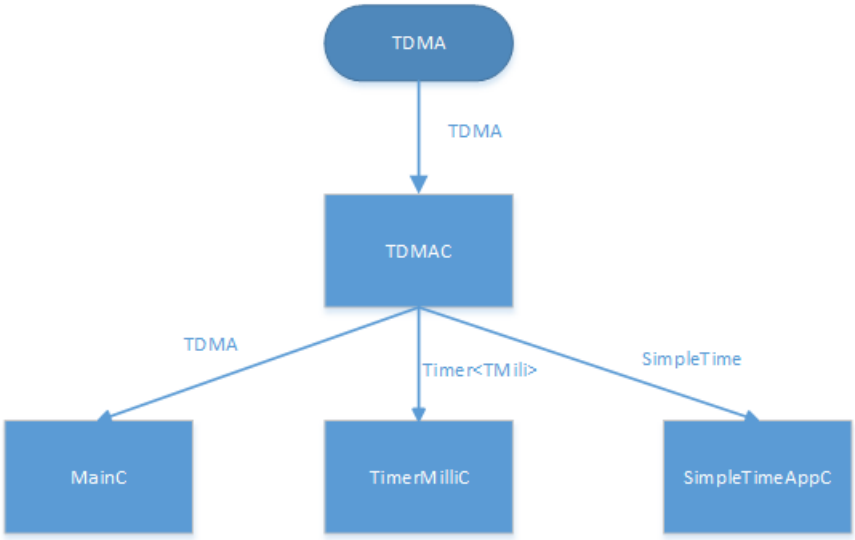

Figure 4.5 component TDMAC's wiring

Time Division Multiple Access provides the mechanism where every mote maintains a logic clock which creates slots, assigned by fixed interval. Slots consist a frame and the length of the frame can be adjusted according to the serial packets created by Felix Dobslaw's routing algorithm.

The procedure of a new slot event is shown in the figure 4.6. This Event not only signal the upper level of this software that a new slot is coming and also calibrate the commence of the slot time that eliminates the physical layer problem because of drift of physical clock.

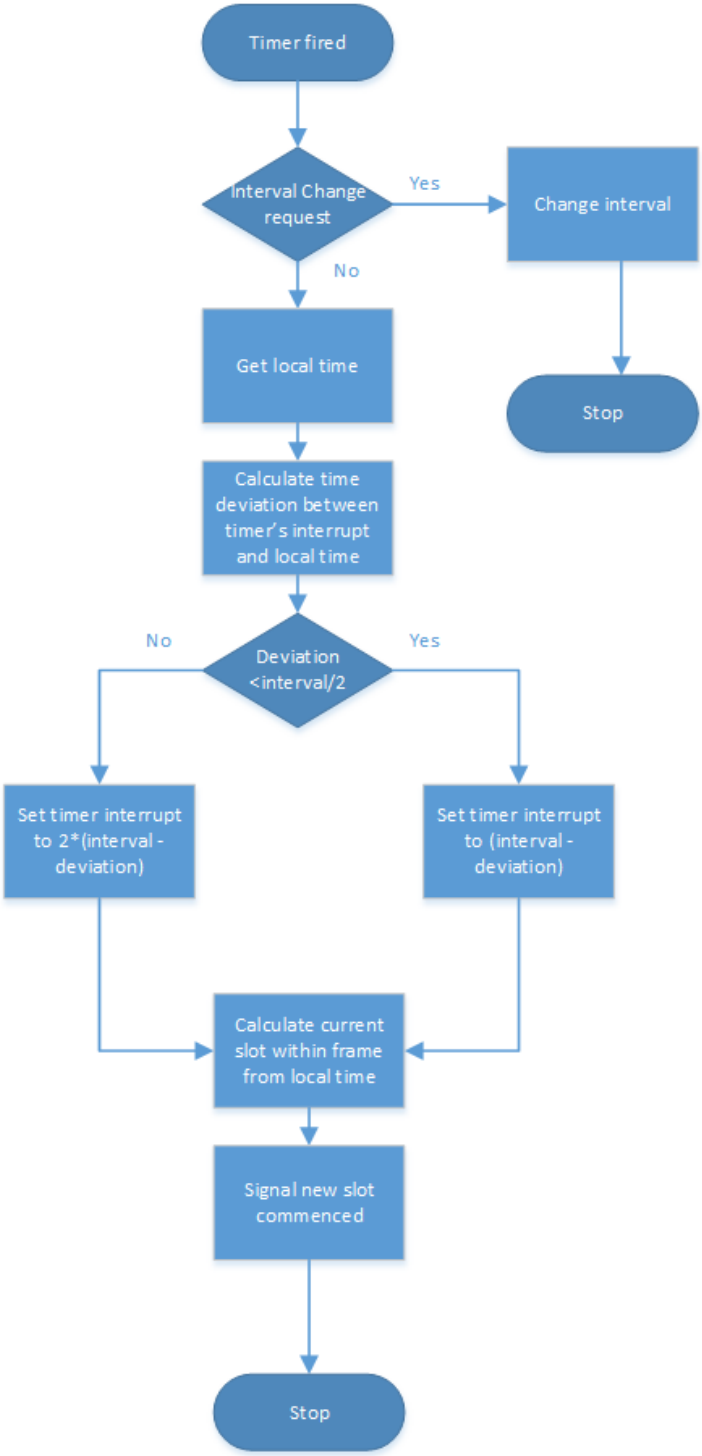

Figure 4.6 new slot event procedure

The structure of SimpletimeAPPC, a sub-block of the whole TDMA block, which provides manipulation and calculation module to TDMA block is shown in figure 4.7.

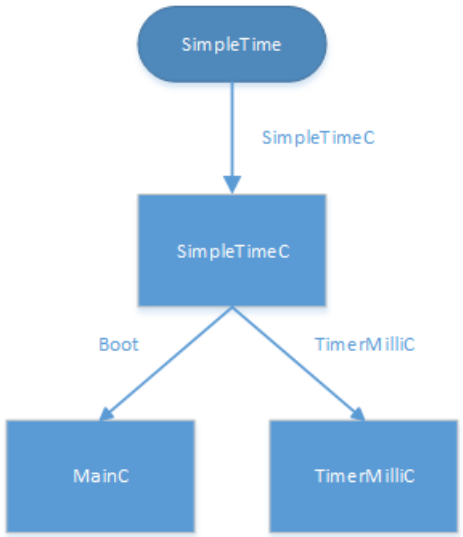

Figure 4.7 structure of SimpletimeAPPC

4.1.3 TinyOSCore

Since the Communication, CSMA/CA and Slots and Routing handling are embedded in the TinyOSCore, we will introduce how the system work and the process that forms a self-organized centralized network automatically.

In the Test, I first use topology generator randomly generate a topology containing 101 motes in a 40000 square meters square. According to the channel model in the building aisles, the program calculates noise floor, received power of each motes pairs. We will discuss how we model the channel and generate the topology in next chapter. Let's go back to our TinyOScore. Since the sink knows that its connected to the PC, it assigns its node level to level 1 and all other nodes assigns its node level to level 100 shown in figure 4.8.

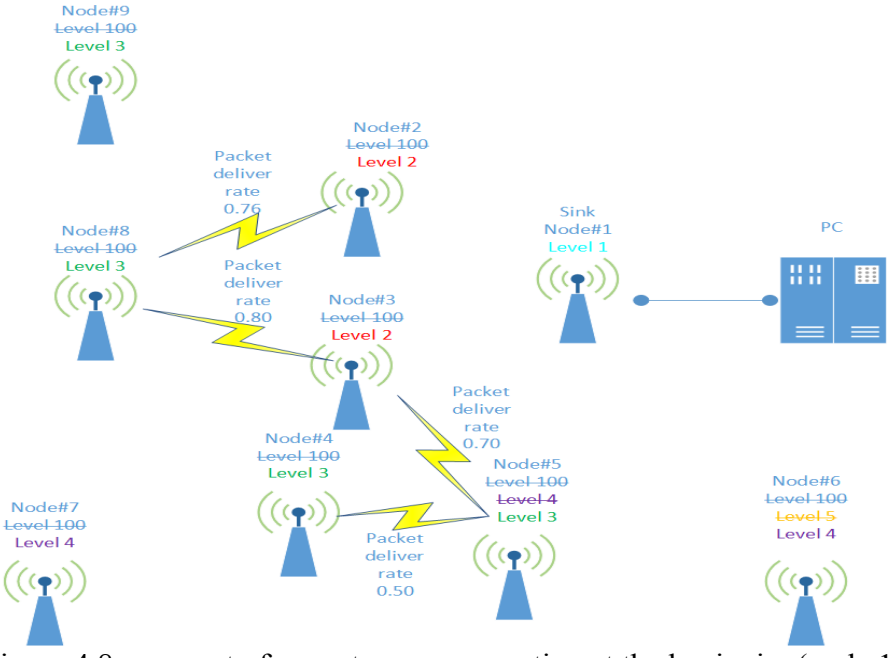

Figure 4.8 process to form a temporary routing at the beginning(node 1 is the sink)

As we can see from figure 4.8, every mote in the networks are booted in the same time and get a rand slot and start to broadcast their beacon packets for estimating the channel discussed in chapter 4.1.1 and forming the temporary routing shown in figure 4.2. The algorithm to form a temporary routing is shown in figure 4.9.

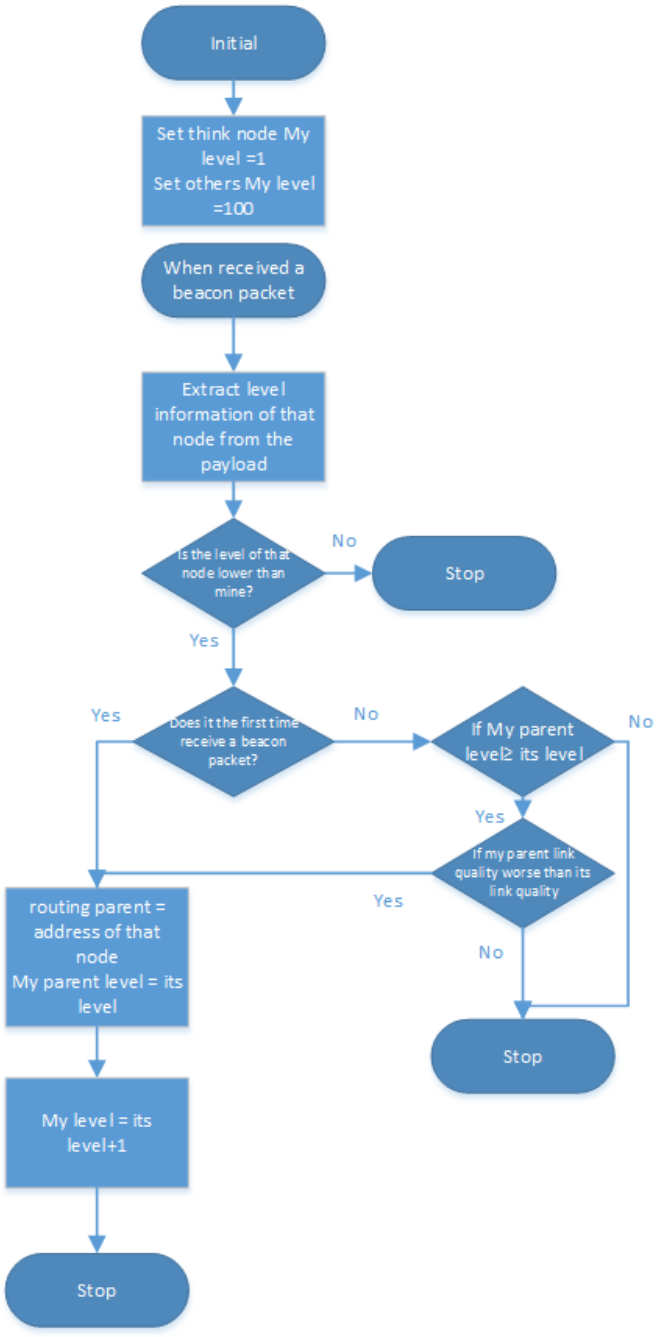

Figure 4.9 algorithm to form a temporary routing

Figure 4.8 explains this algorithm vividly. you can see this process from the tag on the motes clearly.

When the network successfully forms a routing, I set 5 minutes to let all the motes in the network to estimate the channel completely. After 5 minutes, all the motes in the network stop sending beacon message and begin to forward their channel quality packets to the sink. Limited by the length of payload that a packet can support, The maximum number of channel quality information is eight per packet. The structure of quality packet is shown in figure 4.10 and the algorithm of how to send the channel quality packets to the sink is shown in figure 4.11.

| Header | footer 1| footer 2| .... | footer n |

Header:

| My node address| message type | flags |

Footer:

| address | outbound link quality |

Figure 4.10 The structure of link quality message

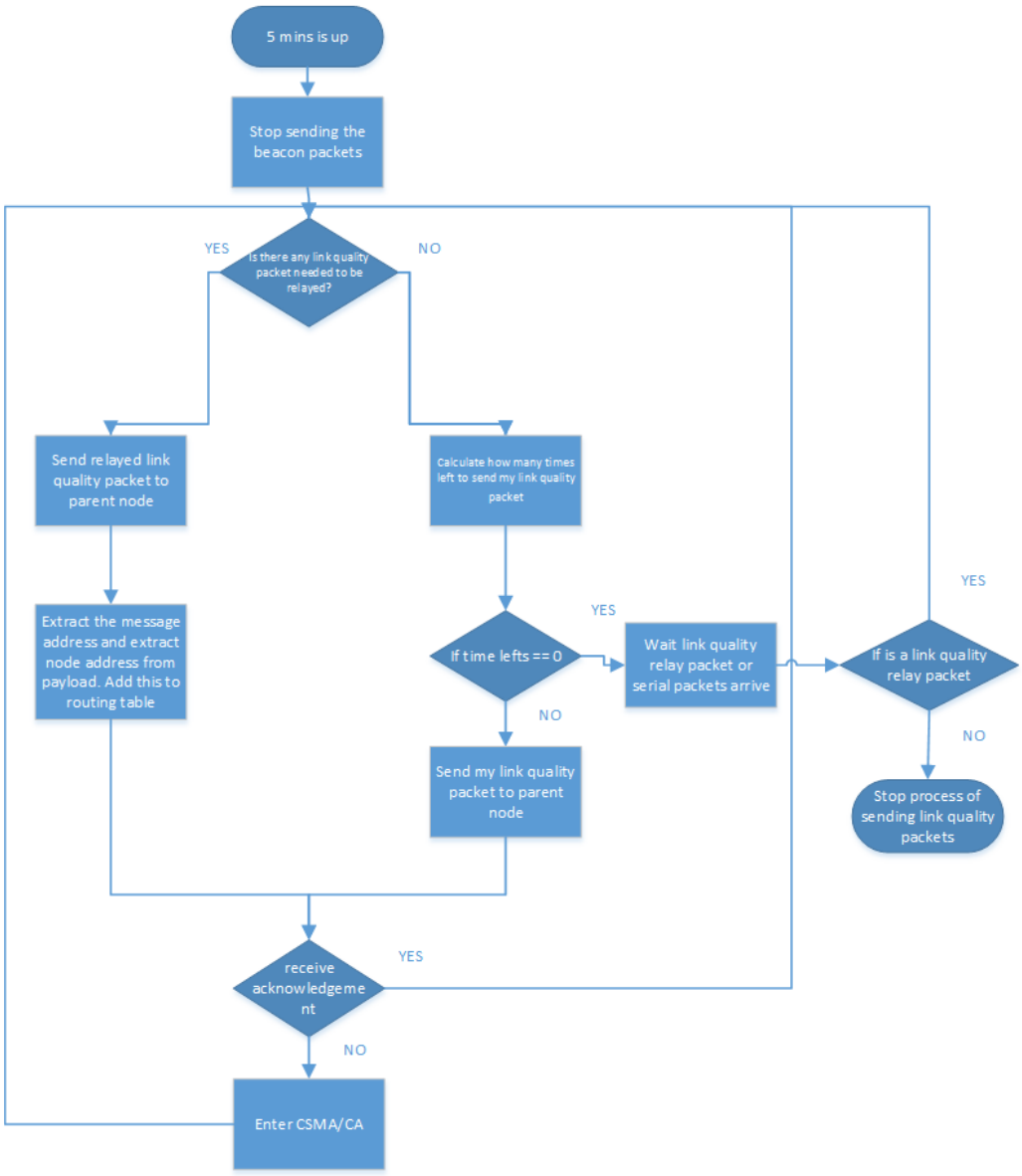

Figure 4.11 algorithm to send the channel quality packets

From 4.11 we can see that we form a routing table by recording which node can be reached through which node. An example of routing table formed by node #1 in figure 4.8 is shown in table 4.1.

Table 4.1 routing table formed by node#1 in figure 4.8

destination	Next hop
2	2
3	3
4	3
5	3
6	3
7	3
8	3
9	2

When all the link quality packets from all of the nodes are received at the sink, The interface between the Felix Dobslaw routing algorithm and TinyOS will send the serial packets containing schedule information such as the length of the frame, the slots that a node will occupy shown in figure 4.12. We will discuss how interface work in the next chapter. Let's focus on how TinyOS core handles the serial packets and begins to test routing algorithm.

| routing destination | parent address | TDMA start time|frame length | length of

---

the slots contained in this packet | slot[0] | slot[1] | …….| slot[length]|

Figure 4.12 structure of serial packets where slot[length], length ≤8

The Procure of how TinyOS core handles the serial packets and start to test the routing algorithm is shown in figure 4.13

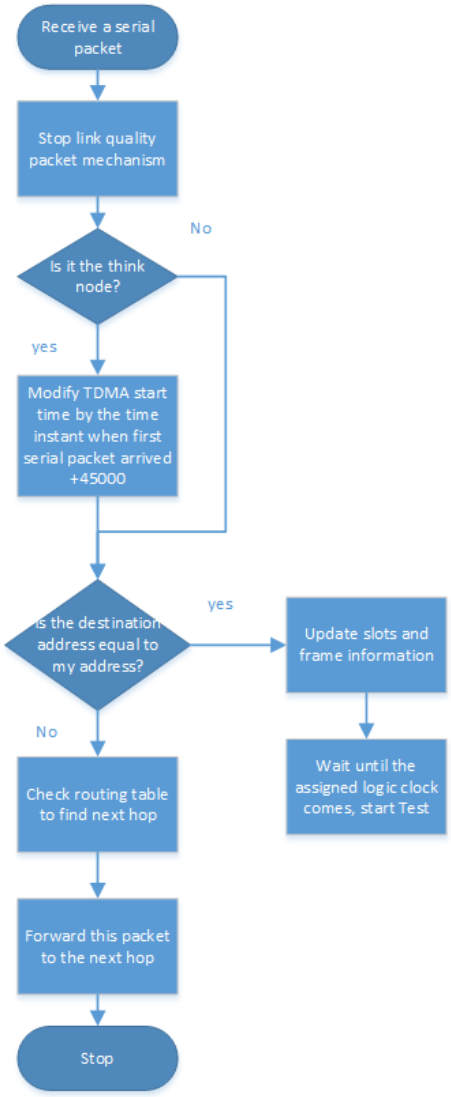

Figure 4.13 TinyOS core handles the serial packets and start the test

When all of the nodes in the networks have successfully being scheduled, they enter a state that is ready for testing. The procedure of testing is shown in figure 4.14

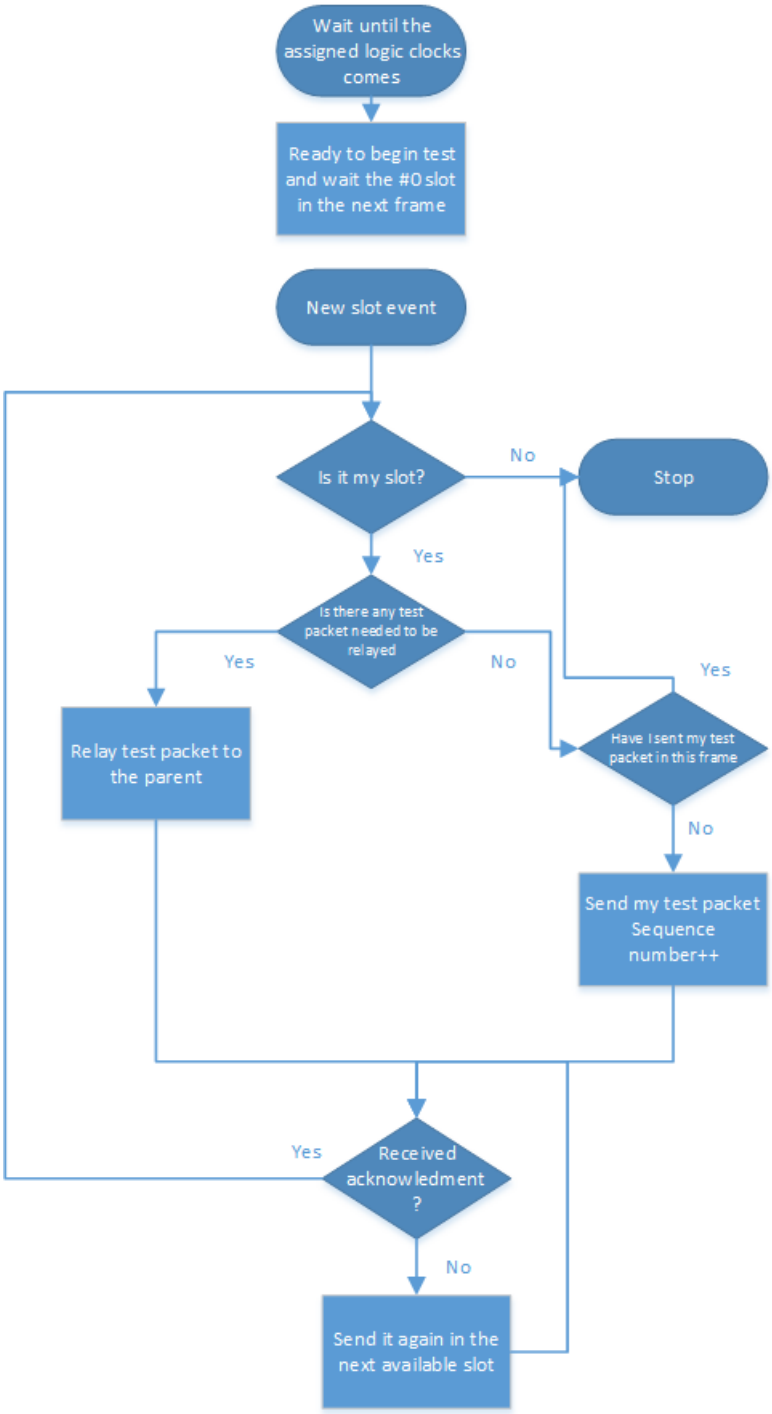

Figure 4.14 The Procedure of testing the routing algorithm

The structure of the testing packet which can be collected from PC is shown in figure 4.15

| mote address | sequence number |

Figure 4.15 the structure of the testing packet

4.2 Simulation parameters provider

4.2.1 Link layer model/channel model

The simulation parameters that Tossim needed is based on the received power from node x to node y. However, in the real situation we only have the topology based on the geological coordinates of each node. So I use the LinkLayerModel developed by University of Southern Canifornia, setting parameters to simulate a building aisle condition and convert the geological coordinates representation into received power representation refer to [17]. The Algorithm of converting process is shown in figure 4.16.

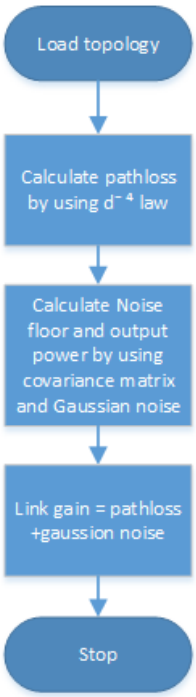

Figure 4.16 the process to covert topology into Tossim parameters

The parameter I used to simulate a building aisle is listed in the Table 4.1

Table 4.1 parameters used for Channel Model to simulate a building aisle environment

rate at which signal decays	PATH_LOSS_EXPONENT =3.3
randomness of received signal due to multipath	SHADOWING_STANDARD_DEVIATION=5.5
reference distance	D0 =1m
power decay in dB for the reference distance D0	PL_D0 = 52.1
radio noise floor in dBm	NOISE_FLOOR = -106.0
covariance matrix S = [S11 S12; S21 S22]	S11 = 3.7; S12 = -3.3; S21 = -3.3; S22 = 6.0;
standard deviation of additive white Gaussian noise	WHITE_GAUSSIAN_NOISE = 4

4.2.2 Topology random generator

Topology random generator is responsible for generating a suitable topology generator that do not have orphan node in the network. The maximum distance between two nodes using the parameters in table 4.1 is 30 meters. So, in order to satisfy a not bad connection between nodes (packet deliver ate >0.3), the good distance between nodes is set to 15 meters. The algorithm to generate a suitable topology is shown in figure 4.17

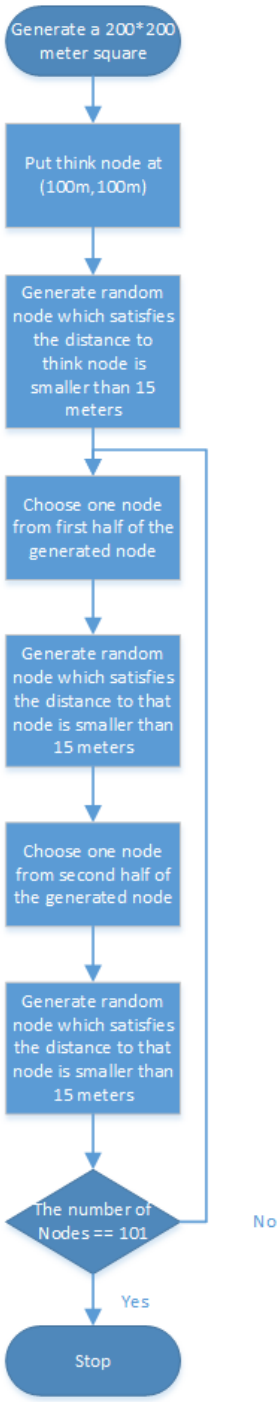

Figure 4.17 topology random generator

4.3 Tossim Emulator

I used the Tossim to simulate a real environment by importing the noise collected from Stanford library. If you look the file, noise.txt in my code, you can see the hardware noise floor is around -105dbm, but there are spikes of interfer-

ence around -86dbm. Then I use Tossim live extension to simulate the serial packets (packets that communicate between think node and PC). By writing a MakeFile that uses mig(Message Interface generator), the program can automatically build a Java, C, python interface to the message structure. The algorithm of Tossim and Tossim-live extension is shown in figure 4.18

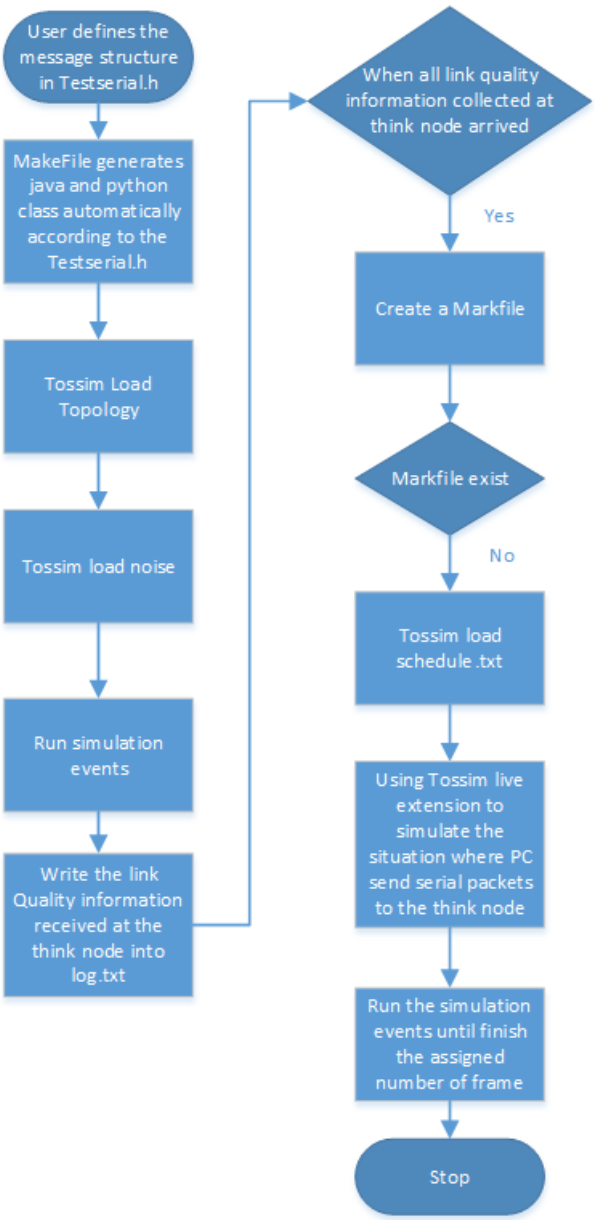

Figure 4.18 the procedure of Tossim Emulator

4.4 Interface between my wireless sensor network system and Felix Dobslaw routing algorithm

As discussed above, my wireless sensor network system can provide some useful information to the sink such as the link quality information, sensor data and so on. The user can take advantage of these useful information and process it in

their the powerful personal computer at the sink and distribute the result to the whole distributed wireless sensor network and command them to perform some activities as expected. In this scenario, I create an interface between my wireless sensor network system and Felix Dobslaw routing algorithm in which Felix Dobslaw routing algorithm can achieve useful link quality information measured in packet deliver rate and command all the sensors in the network to be scheduled with assigned time slots and frame length. The structure of this interface is shown in the figure 4.19.

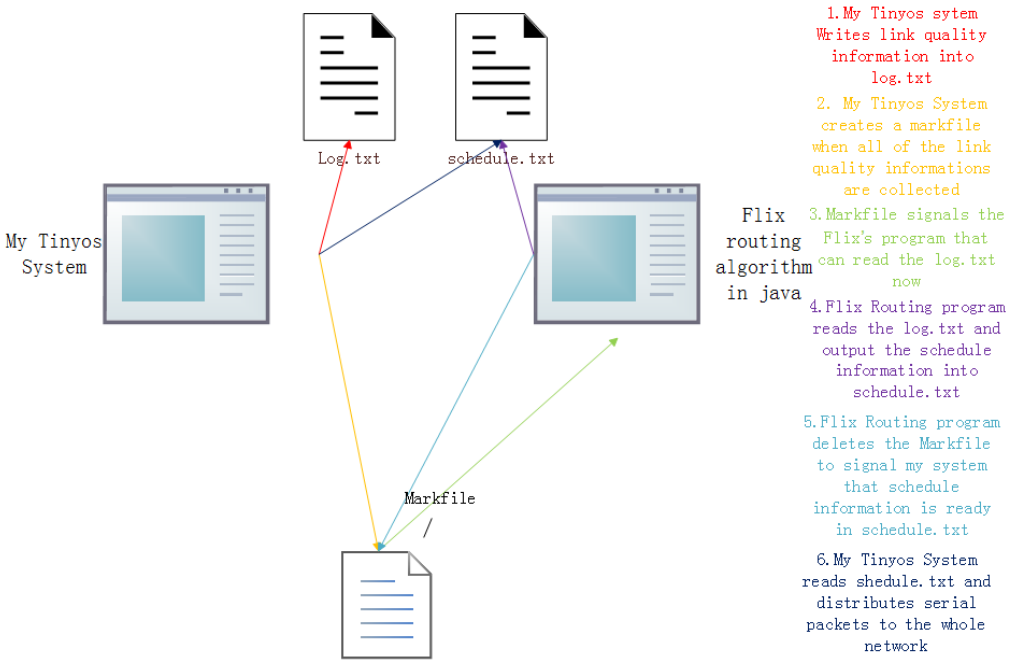

Figure 4.19 the interface between My wireless sensor network system and Felix Dobslaw 's Routing algorithm

5 Results

This Chapter shows the results of this project.

5.1 The link quality information collected at the sink

The structure of the link quality information at the sink from motes is stored in log.txt shown in figure 5.1

```
50 0 3 8 0 39 f4 0 1 e3 0 5b db 0 51 f6 0 5c e6 0 2d d5 0 20 e9 0 17 f4
50 0 2 8 0 43 fb 0 3e fb 0 36 e3 0 11 e9 0 c fb 0 7 f5 0 3 e4 0 5d fb
50 0 1 8 0 5e fb 0 39 f4 0 1 e3 0 5b db 0 51 f6 0 5c e6 0 2d d5 0 20 e9
61 0 2 8 0 39 fb 0 2f cf 0 5f f3 0 5b e7 0 17 fb 0 c e0 0 7 f5 0 3 e5
61 0 1 8 0 5d b3 0 38 77 0 5e d0 0 39 fb 0 2f cf 0 5f f3 0 5b e7 0 17 fb
5d 0 2 8 0 5c e0 0 2d ee 0 50 f7 0 17 f0 0 3e f7 0 14 e2 0 11 f2 0 7 fb
```

Figure 5.1 link quality information samples in Hexadecimal from log.txt

The structure of this link quality information is explained in figure4.10. We take the last packet as an example, 5d 0 means that the link quality packet address is from number 93 mote. Because the limited length of the payload, the link quality information might need many packets to deliver. 2 means this packet is the penultimate packet from number 93 mote. 8 means there are 8 footers in this packet which contains the link quality information. 0 5c e0 means the link quality from number 93 mote to number 92 is 224. Because I have quantized the link quality information from 0 to 255 which can be represented by 1 byte, we should use 224 divided by 255 which means the packet deliver rate from number 93 mote to number 92 is 0.878.

5.2 The schedule information and routing tree created by Felix Dobslaw's program

After Felix.Dobslaw's program processed all of the link quality information from the log.txt, it will generate the schedule information which contains the routing schedule and TDMA structure of the motes in the wireless sensor network shown in figure 5.2.

```
4 34 10 274 8 1 39 7 42 8 9 40 41
4 34 10 274 8 15 16 21 22 25 24 27 26
5 15 10 274 6 2 6 7 8 9 10
6 1 10 274 8 103 2 82 22 93 42 111 62
7 1 10 274 4 3 23 43 63
```

Figure 5.2 schedule packet stored in schedule.txt

The structure of schedule packet is explained in Figure 4.12. We also take the last packet as an example. 7 means this packet's destination is number 7 mote. 1 means number 7 mote's routing parent is number 1 mote. 10 is the place to hold the TDMA start time which will be changed by think node according to time instant of the first serial packet that arrives in order to let all the motes start test simultaneously. 274 means that the frame length is 274 from slot 0 to slot 273. 4 means there are 4 slots assigned to that slot which are slot 3, slot 23, slot 43, slot 63. Also you can see that there are many schedule packets which have the same destination because the maximum payload length is 28, each of the slot takes 2 byte. There is not enough room to hold so many slots assigned to that mote.

The routing tree is a visualized tools to view the network, you can see how the network is formed directly. However, Since there are 101 motes in the network, we cannot put that picture in the paper because its too large. You can see it in the Appendix of this project with the code implementation.

5.3 Packet deliver rate in 20 frames with different user demand using Felix Dobslaw's Routing algorithm tested in my system

Table 5.1 the test result of different user demands

Topology/user demand		0.9	0.999	0.99999
topology1	PDR	0.995	1.0	1.0
	TRP	1990	2000	2000
	DP	464	441	388
topology2	PDR	0.9985	0.9995	1.0
	TRP	1997	1999	2000
	DP	305	432	398
topology3	PDR	0.9895	1.0	0.9995
	TRP	1979	2000	1999
	DP	399	470	482
topology4	PDR	0.9985	1.0	1.0
	TRP	1997	2000	2000
	DP	400	376	352
topology5	PDR	0.9965	0.9995	1.0
	TRP	1993	1999	2000

	DP	337	352	318
topology6	PDR	0.995	1.0	1.0
	TRP	1990	2000	2000
	DP	241	238	263
topology7	PDR	0.998	0.999	0.9995
	TRP	1996	1998	1999
	DP	354	428	336
topology8	PDR	0.997	1.0	1.0
	TRP	1994	2000	2000
	DP	421	444	450
topology9	PDR	0.999	1.0	0.9985
	TRP	1998	2000	1997
	DP	394	455	464
topology10	PDR	0.9975	0.9995	0.9995
	TRP	1995	1999	1999
	DP	358	320	336
Topology 11	PDR	0.9985	1.0	0.9995
	TRP	1997	2000	1999
	DP	289	334	303
Topology 12	PDR	0.997	1.0	1.0
	TRP	1994	2000	2000
	DP	352	293	328
Topology 13	PDR	0.9985	1.0	0.999
	TRP	1997	2000	1998
	DP	439	437	474
Topology 14	PDR	0.999	0.9995	1.0
	TRP	1998	1999	2000
	DP	304	293	342
Topology 15	PDR	0.9975	1.0	0.9995
	TRP	1995	2000	1999
	DP	306	271	357
Topology 16	PDR	0.9975	1.0	1.0
	TRP	1995	2000	2000
	DP	392	475	430

Topology 17	PDR	0.998	1.0	0.9995
	TRP	1996	2000	1999
	DP	494	454	528
Topology 18	PDR	1.0	0.999	1.0
	TRP	2000	1998	2000
	DP	290	299	302
Topology 19	PDR	0.996	1.0	1.0
	TRP	1992	2000	2000
	DP	307	287	286
Topology 20	PDR	0.997	1.0	1.0
	TRP	1994	2000	2000
	DP	202	209	229
Topology 21	PDR	0.995	1.0	1.0
	TRP	1999	2000	2000
	DP	392	451	431

This test result only tests 20 frame per user demands per topology. There are 101 one motes in the wireless sensor network. Each of them except the think node want to send one packet to the routing parent in one frame. So the total number of packets expecting to be received is 2000. Total Received Packets(TRP) represents the number of packets received in 20 frames. You can use this value divided by 2000 to get the Packet Deliver Rate(PDR). There are 3 different user demands which is 0.9, 0.99, 0.999. The frame length of the schedule is smaller when the user demand is smaller which also means a low latency. You an also see that the lower user demands cause packet lost in the test represented by Packet deliver rate. The Duplicated packets(DP) is because when a mote send a packet to its parent, its parent receive this packet. However, this node do not receive a ACK back from its parent. So it will transmit this test message again in the next available slot. Thus, there are a lot of same packets being delivered to the sink.

6 Conclusions

As you can see from the chapter 5, I tested my system with 20 randomly generated topologies with different user demands. It serves our client's centralized control system – Felix Dobslaw's Weighted Shortest Path new Routing algorithm using ETX very well. From the result, different user demands and different topology structure caused different packet deliver rate and different number of duplicate packets in the network. The link quality information that stored in log.txt shows that my system successfully fulfill the first and second overall aims - builds an initial routing tree and collects the link quality information from mote to its neighbors. We can also see from the debug messages in the simulation that every mote in the wireless sensor network has successfully formed TDMA schedule and Routing schedule according to the schedule.txt created by Felix Dobslaw's new routing algorithm. The worst packet deliver rate is above 0.99 that not only proves the efficiency of Felix Dobslaw's new routing algorithm, but also makes this system seems like more reliable and trustworthy. An efficient routing has been achieved which lets motes saves the precious power and improves the life of the wireless sensor network. Also, by using the safety coding criteria, this system does not have any segmentation fault or kernel corruption problem and run smoothly. TinyOS provide a good simulation environment and resourceful tools to simulate wireless sensor network. However, the simulation need a fast computer that has a good CPU. It runs very slow in my laptop, taking about 1 day to run in a higher user demand. The channel model provides a real industrial environment that calculates the channel condition based on the received power well and stimulates a high asymmetric channel in the building aisle condition. Algorithm to build the initial topology is also very efficient and intelligent which always choose a parent with a good link quality. The CSMA/CA and TDMA successfully avoid the collision problem at the beginning when there is no schedule information which uses the channel at its full potential. Interface serves the communication between programs very well and hide the physical layer's details from centralized control system. Since, I have finished all of my overall aims and problem statement stated in the chapter 1 successfully.

The most vital contribution of this project is providing an incredible platform for the users who want to use wireless sensor network either to test their own algorithms or uses my system to do some real applications such as environment monitoring. Also, My system provide a considerable flexibility for those who want to use my system. I wrote an makefile that only needs user to write their own packet structure into a head file(.h file) and the Message Interface generator will automatically generate the classes and methods which can be used in my system. What's more, My system is also very user-friendly, like I stated in this report, Felix Dobslaw used my system to test his new routing algorithm who only needs to consider upper level because my system has implemented all the things of the physical level and my interface between user's program and my system has hidden all of the physical details from my clients. Another big

advantage of my system is its extensibility, the channel model used to generate Tossim parameters which can be adapted into different scenarios like building aisle, football field and so on. Users can simulate different scenarios according to the specified environment they used by changing the channel parameters in the channel model.

6.1 Ethical issues

Wireless sensor network can serves our life in many different ways. For example, in healthcare application, it can helps us to monitor the situation of the patients with a very cheap price instead of using wired, cumbersome monitor devices. My system can make the implementation for these applications easier and improves the excepted life of wireless sensor network. However, If some evil people can easily use this application into some bad condition because of the user-friendly and adaptability of my system, the results could be disastrous e.g. surveying others privacy or uses to break some crucial systems. Thus, this technology is a double-blade. It can benefit our human beings or destroy the whole world which depends on how we use it.

6.2 Future works

Since I only test my system in the TOSSIM simulation which guarantees a perfect time synchronization between motes in the wireless sensor network, Thus, Considering the clock drift in the real situation, the first thing need to be done in the future work is to implement a time synchronization between motes which can be implemented like network time protocol. Since this system is only valid in static and low mobility condition, how to extend this system to a high mobility condition will be another challenge to implement.

References

[1] Approved IEEE Draft Amendment to IEEE Standard for Information Technology-Telecommunications and Information Exchange Between Systems-Part 15.4:Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANS): Amendment to Add Alternate Phy (Amendment of IEEE Std 802.15.4)

[2] Karl, Holger, and Andreas Willig. Protocols and architectures for wireless sensor networks. John Wiley & Sons, 2007.

[3] Cali, Federico, Marco Conti, and Enrico Gregori. "IEEE 802.11 protocol: design and performance evaluation of an adaptive backoff mechanism." Selected Areas in Communications, IEEE Journal on 18.9 (2000): 1774-1786.

[4] IEEE Standard for Wireless LAN Medium Aceess Control(MAC) and Physical Layer(PHY) Specification, IEEE Std 802.11, 2007.

[5] Nelson, Randolph, and Leonard Kleinrock. "Spatial TDMA: A collision-free multihop channel access protocol." Communications, IEEE Transactions on 33.9 (1985): 934-944.

[6] Andrew S. Tanenbaum, Maarten van Steen, Distributed Systems: Principles and Paradigms. 2ed. Pearson Prentice Hall, 2007

[7] Sukumar Ghosh, Distributed Systems: An Algorithmic Approach, 2ed.CRC Press, Jul 14, 2014

[8] Philip Levis, David Gay, TinyOS Programming. 1ed. Cambridge University Press, Mar 12, 2009

[9] Wikipedia,http://tinyos.stanford.edu/tinyos-wiki/index.php/TinyOS_Tutorials, Retrieved 2010-05-11.

[10] Wikipedia,http://tinyos.stanford.edu/tinyos-wiki/index.php/TOSSIM_Live, Retrieved 2010-05-11.

[11] http://www.contiki-os.org/. Retrieved 2005-06-13

[12] Reusing, Tobias. "Comparison of operating systems TinyOS and Contiki." Sens. Nodes-Oper. Netw. Appl.(SN) 7, 2012.

[13] https://www.millennium.berkeley.edu/pipermail/tinyos-help/2010-November/048751.html Retrieved 2005-05-13

[14] Simon Haykin, communication system, 4th edition, 2000

[15] Andreas F. Molisch, Wireless Communications, 2nd Edition.2010

[16] http://tinyos.stanford.edu/tinyos-wiki/index.php/TOSSIM. Retrieved 2005-05-13

[17] http://www.tinyos.net/tinyos-2.x/doc/html/tutorial/usc-topologies.html. Retrieved 2005-05-13