

Själv

Huvudområde
Major Subject

Titel
Eventuell underrubrik

Förnamn Efternamn

--

MID SWEDEN UNIVERSITY

Name of department (division)

Examiner: First name Family name, xxxxx@miun.se

Supervisor: First name Family name, xxxxx@miun.se

Author: First name Family name, xxxxx0000@student.miun.se

Degree programme: XXXXXXXXXXXXXXXXXX, 180 credits

Main field of study: XXXXXXXXXXXXX

Semester, year: XX, 20XX

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 client program process the data that we collected. This report mainly focus 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 Flix's new routing algorithms program 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 new routing algorithm as required.

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

Acknowledgements / Foreword

Acknowledgements or Foreword (choose one of the heading alternatives) are not mandatory but can be applied if you as the writer wish to provide general information about your exam work or project work, educational programme, institution, business, tutors and personal comments, i.e. thanks to any persons that may have helped you. Acknowledgements are to be placed on a separate page.

Table of Contents

Abstract	iii
Acknowledgements / Foreword	iv
Terminology / Notation	vi
1Introduction	1
1.1Background and problem motivation	1
1.2Overall aim / High-level problem statement	2
1.3Scope	2
1.4Concrete and verifiable goals / Detailed problem statement	2
1.5Outline	3
1.6Contributions	3
2Theory / Related work	4
2.1Definition of terms and abbreviations	4
2.1.1Example of level 3 heading	4
2.2To review or quote	5
2.3References and source references	5
2.4Automatically numbered source references	6
2.5Illustrations	7
2.6Mathematical formulas	8
3Methodology / Model	9
4Design / Implementation	10
5Results	11
6Conclusions / Discussion	12
References	13
Appendix A: Documentation of own developed program code	14
Example of Appendix subheading	14

Terminology / Notation

(Choose one of the headline alternatives.) A possible list of terms, abbreviations and variable names with brief explanations may be placed after the table of contents, but is not required. Note that although a term is explained in the list of terms, it should also be explained in the chapter text where it is used the first time.

Acronyms/Abbreviations

ACK	Acknowledge. Verification of a correctly transferred message
AWGN	Additive White Gaussian Noise.

Mathematical notation

$G(x)$	CRC generator polynomial
k_{isi}	Degree of Inter-symbol-interference.

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, become 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 issues 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 / High-level problem statement**

The project overall aim is to implement a centralized, self-organized wireless sensor network which can finish following 3 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.

In this report, my system provides Flix's program with a link quality matrix based on Packet Deliver rate and uses Flix's 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 by using TinyOS. 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 Flix'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 software 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 being 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 that I devised.

3. Constrained by limited power resources and 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 a first 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^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 between TinyOS at the mote side and java application at the personal computer side.

1.5

Outline

Briefly describe the report's outline. "Chapter 2 describes..."

1.6

Contributions

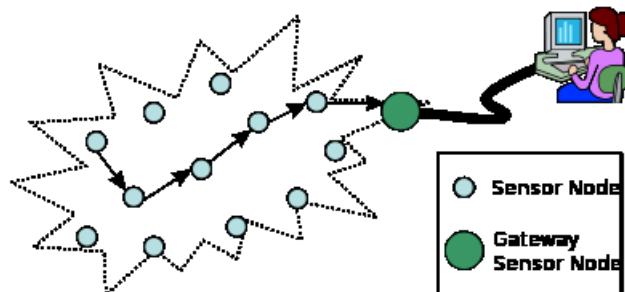
Describe which parts of the work that you have conducted yourself, and which parts that you had help with i.e. carried out by colleagues. If the work is carried out in a group the report should then explain how the tasks were divided between authors. All co-authors should be credited in the work as a whole.

2 Theory / Related work

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 in 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 put 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 data collected to a remote gateway sensor node through network.
- 3: deduct information by data fusion, which obtained from different locations.



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 **MAC Protocols for Wireless Sensor Network**

2.2.1 **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 [2].

2.2.2 **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 want to transmit a packet, before transmitting it, the node should sense the channel to determine where 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.

2.3 **Time division multiple access(TDMA) Scheduling**

In order to meet guaranteed access or bounded latency of industrial standards, scheduled protocols, such as TDMA, 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.

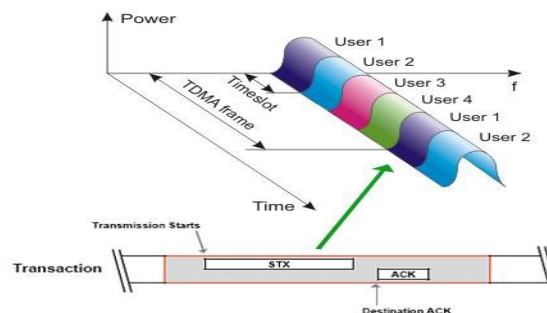


Figure 2.2

2.4

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.

2.4.1

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:

$$P_{RX}(d) = P_{RX}(1m) - 20\log(d_{breakpoint}m) - n10\log(d/d_{break})$$

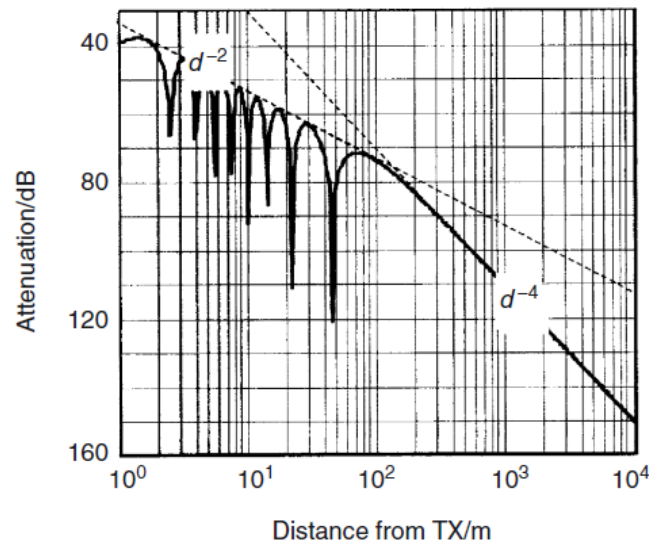


figure 2.3 propagation over an ideally reflecting ground

2.4.2 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

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

2.4.3 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

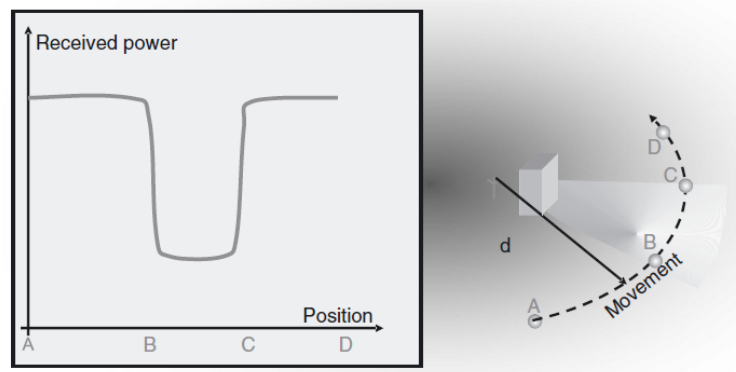
2.4.4 Noise floor

Noise floor is a parameters 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.

2.4.5 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 multipath propagation components. And 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 figure 2.4.

figure 2.4 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 very fast over the period of use.

2.5 ***Distributed system***

2.5.1 **Logic clock**

Wireless sensor network is a distributed system that has no physically synchronous global clock. 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.

2.5.2 **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

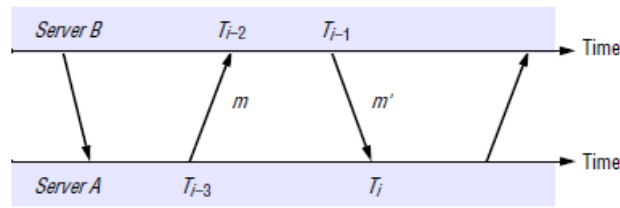


figure 2.5 message changes between two NTP peers

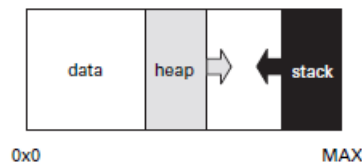
The estimated offset between these two is $o_i = (T_{i-2} - T_{i-3} + T_{i-1} - T_i)/2$

2.6 TinyOS and its extension

2.6.1 TinyOS

TinyOS is a lightweight, opensource operating system which is used to design wireless low-power sensors, having a very aggressive mechanisms 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

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. 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 envi-

Title – Subtitle

Author

2010-07-05

ronment. While, TOSSIM-live is an extension of TOSSIM, allowing users to forward their serial messages (PC to motes) to the virtual motes.

3 Methodology / Model

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. However, I 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 Dobsław.

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 [5] and [6] and build a channel that uses the d^{-4} power law to 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 collected from the library of Stanford. Then I set the parameters which is suitable for a static and low mobility building aisle environment.

In order to make goal (3) and (4) possible, I design an algorithm to let motes automatically build an initial topology 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 which will be explained at Chapter 4 in details. 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.

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 a 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.

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.

In order to achieve goal (7), I designed a data structure used for programs to communicate and create a mark file which being used to notice their counterpart to take some specified action.

4 Design / 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.

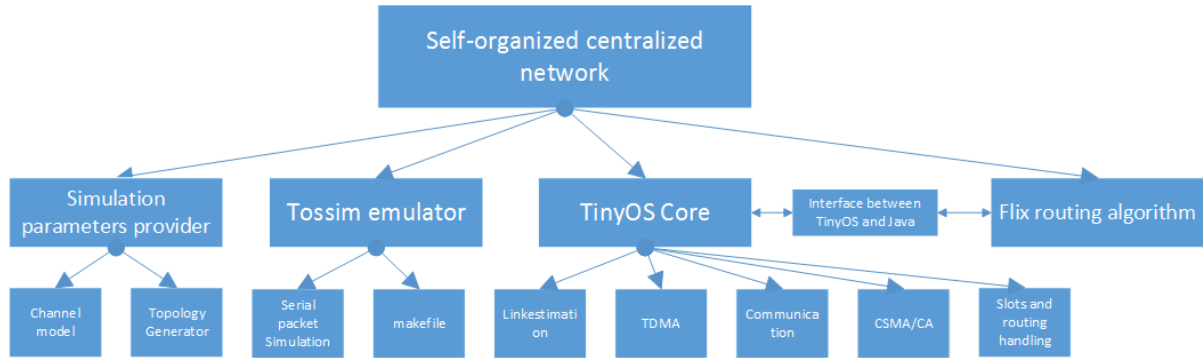


figure 4.1 software layout

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 this 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. The procedure for motes to send beacon message is shown in figure 4.2

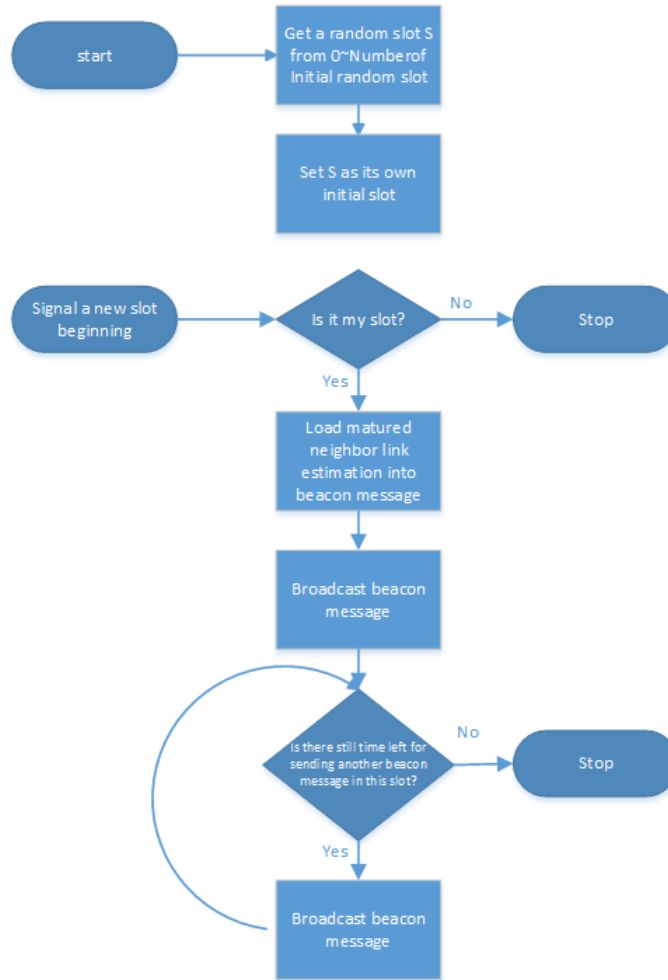


figure 4.2 procedure to send beacon messages

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.

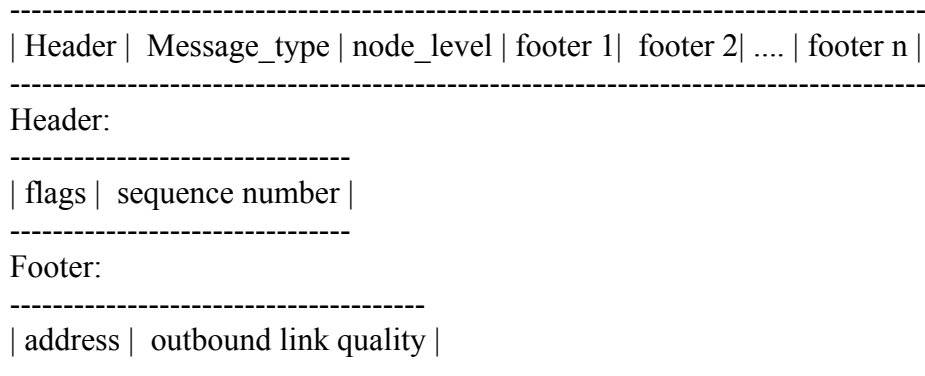


Figure 4.3 The structure of beacon message

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

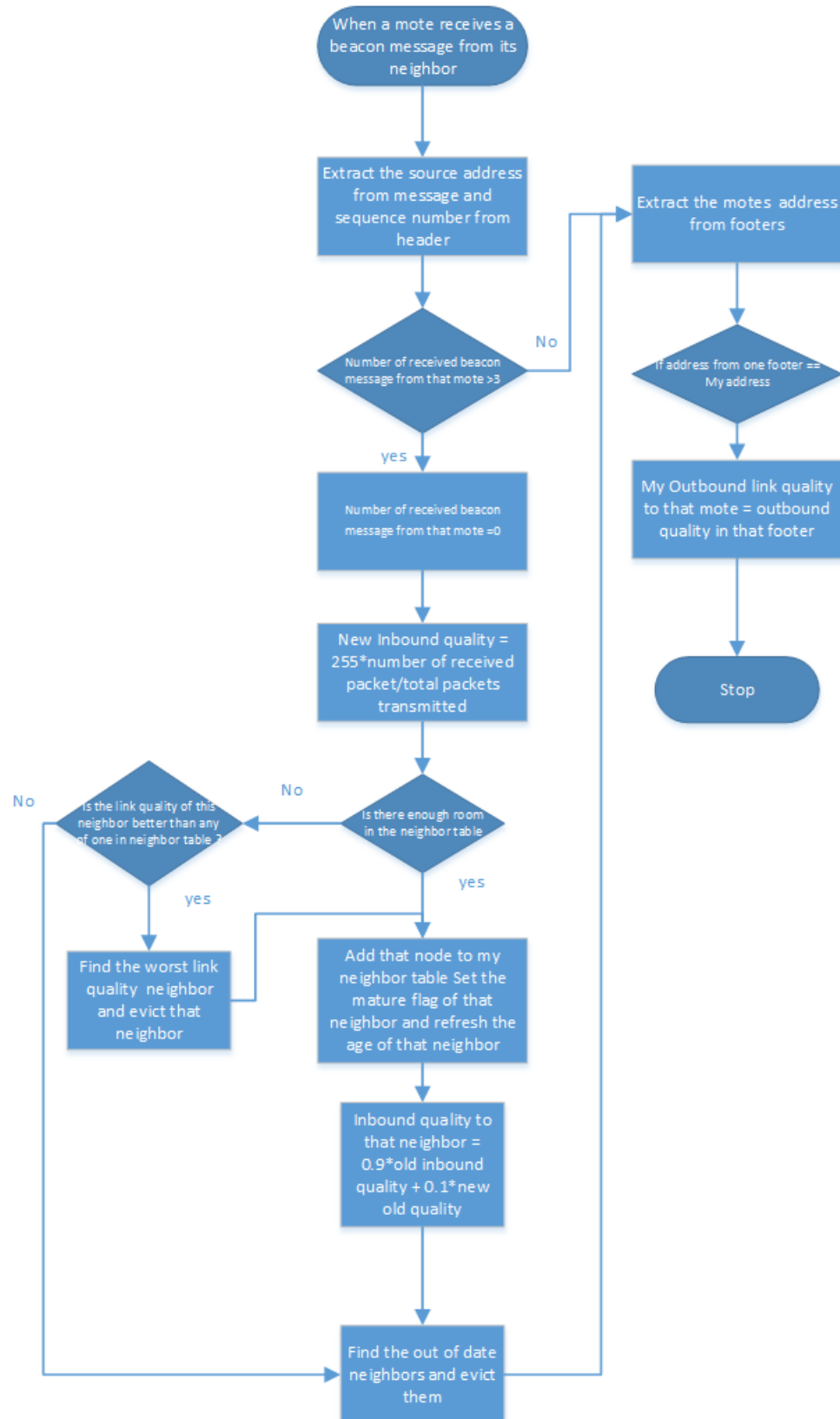


figure 4.4 procedure to estimate channel by using beacon message

4.2

TDMA

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

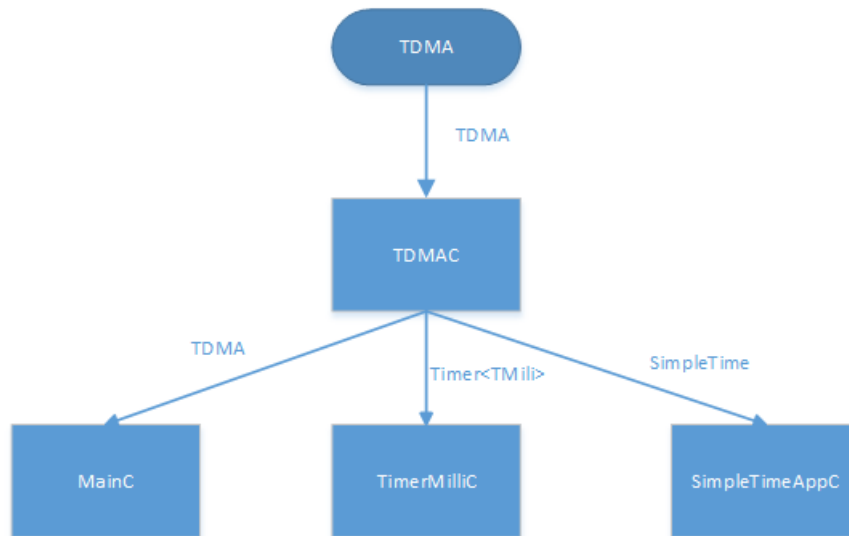


figure 4.5 component TDMAC's wiring

Time Division Multiple Access provides the mechanism where every mote maintains a logic clock which create 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 Flix'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.

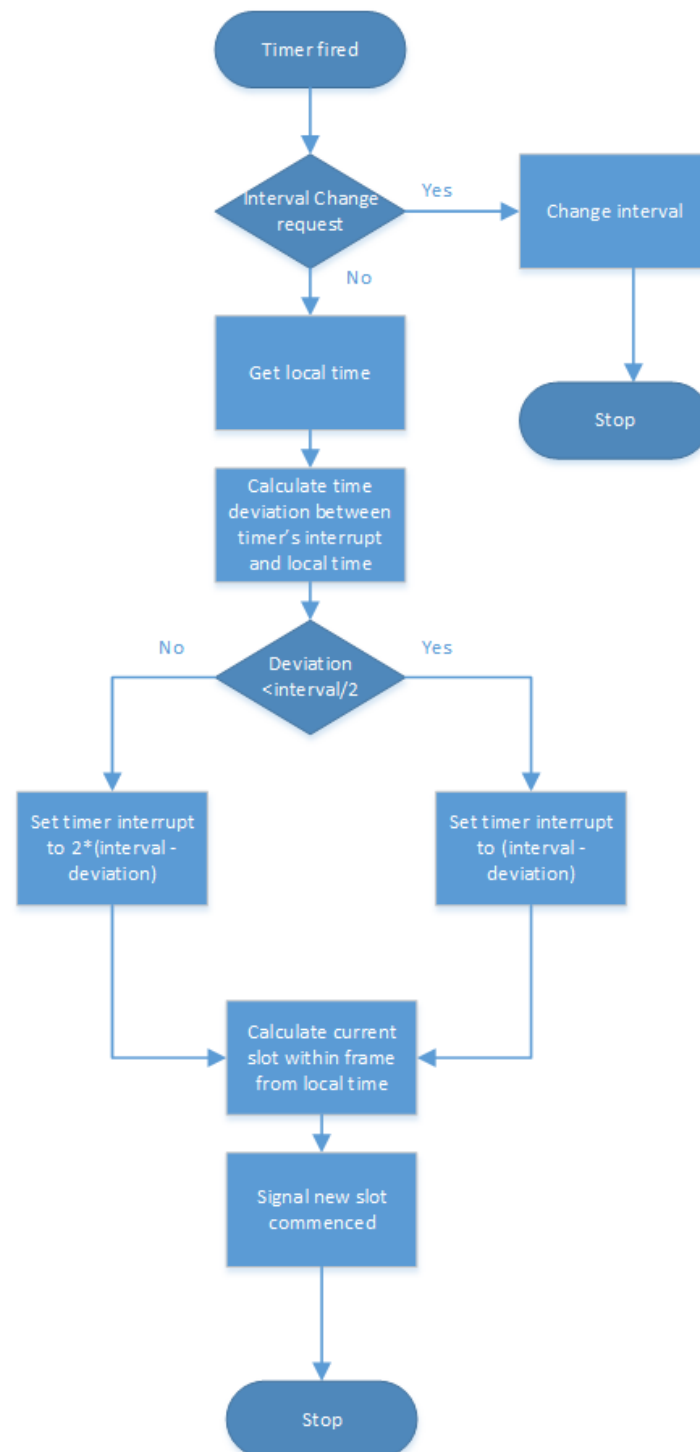


figure 4.6 new slot event procedure

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

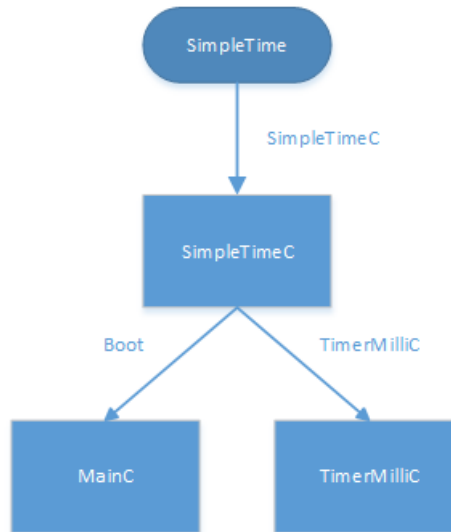


figure 4.7 structure of SimpletimeAPPC

4.3

TinyOSMain

Since the Communication, CSMA/CA and Slots and Routing handling are embedded in the TinyOSMain, 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.

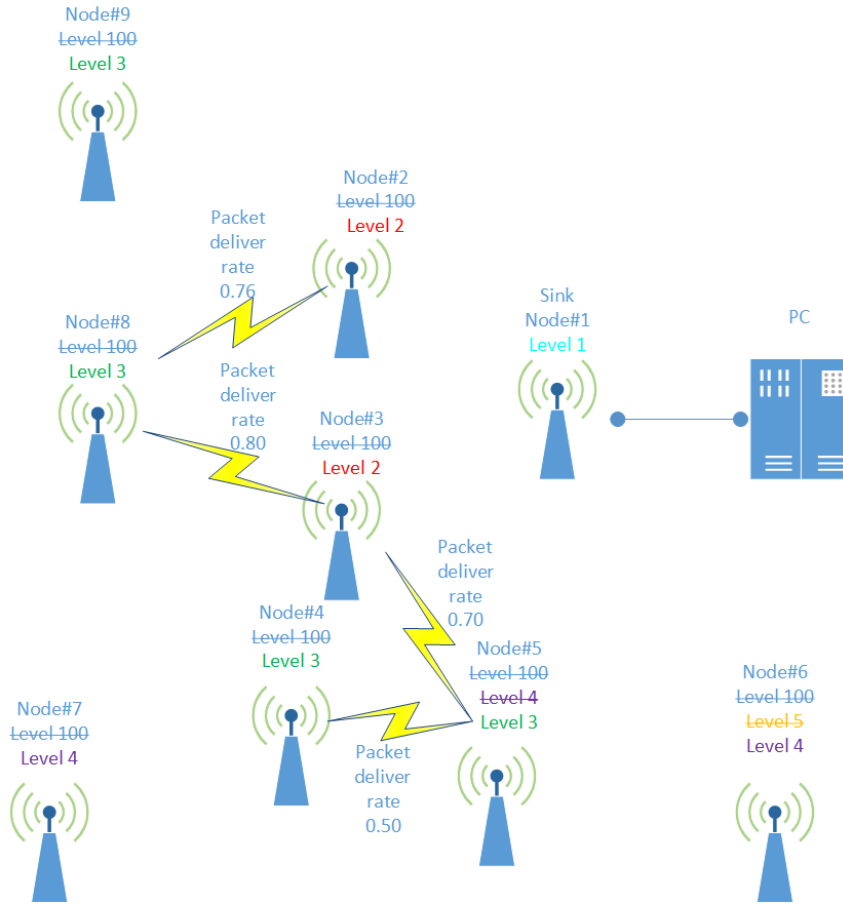


figure 4.8 process to form a temporary routing at the beginning(node 1 is the sink)

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.

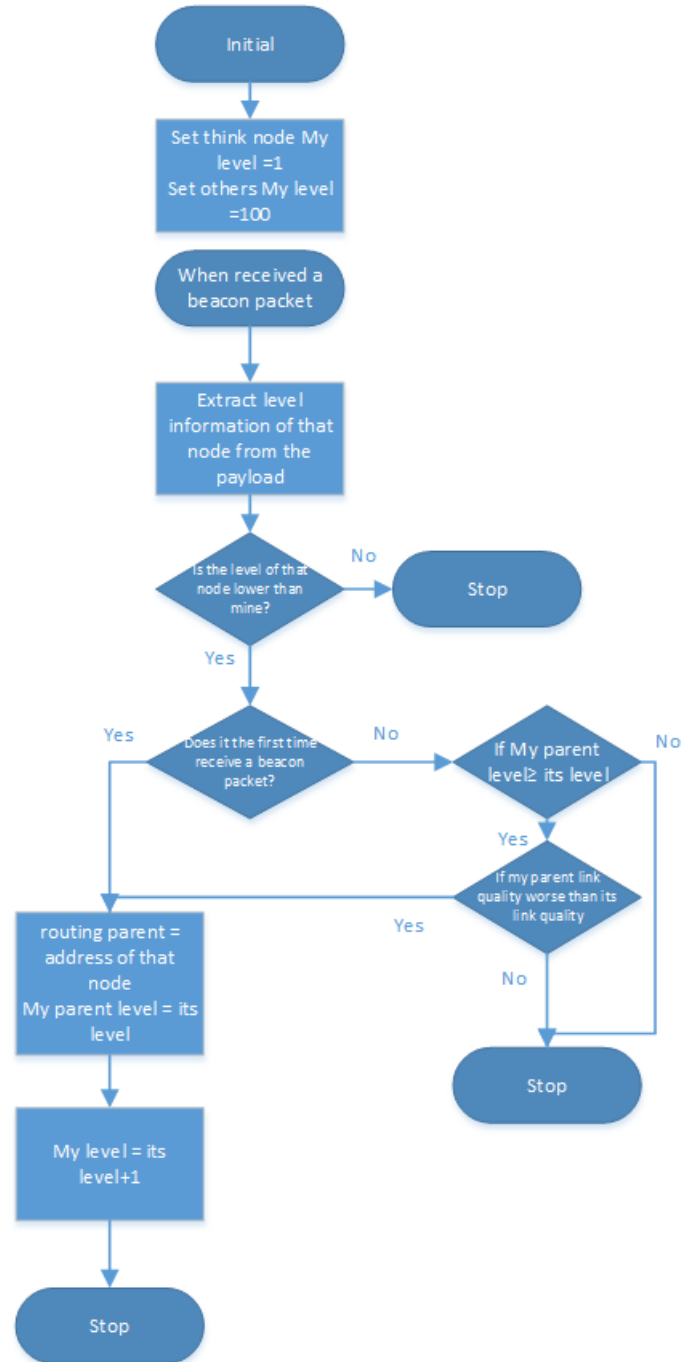


figure 4.9 algorithm to form a temporary routing

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

When the network successfully form 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.

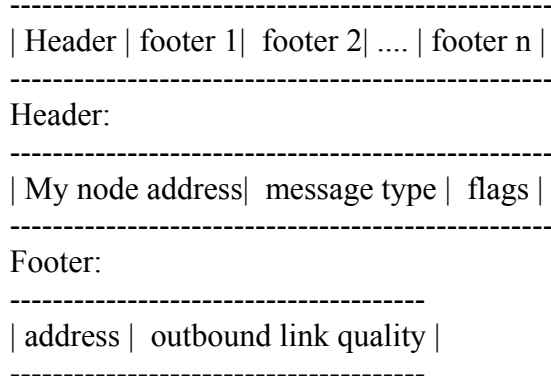


Figure 4.10 The structure of link quality message

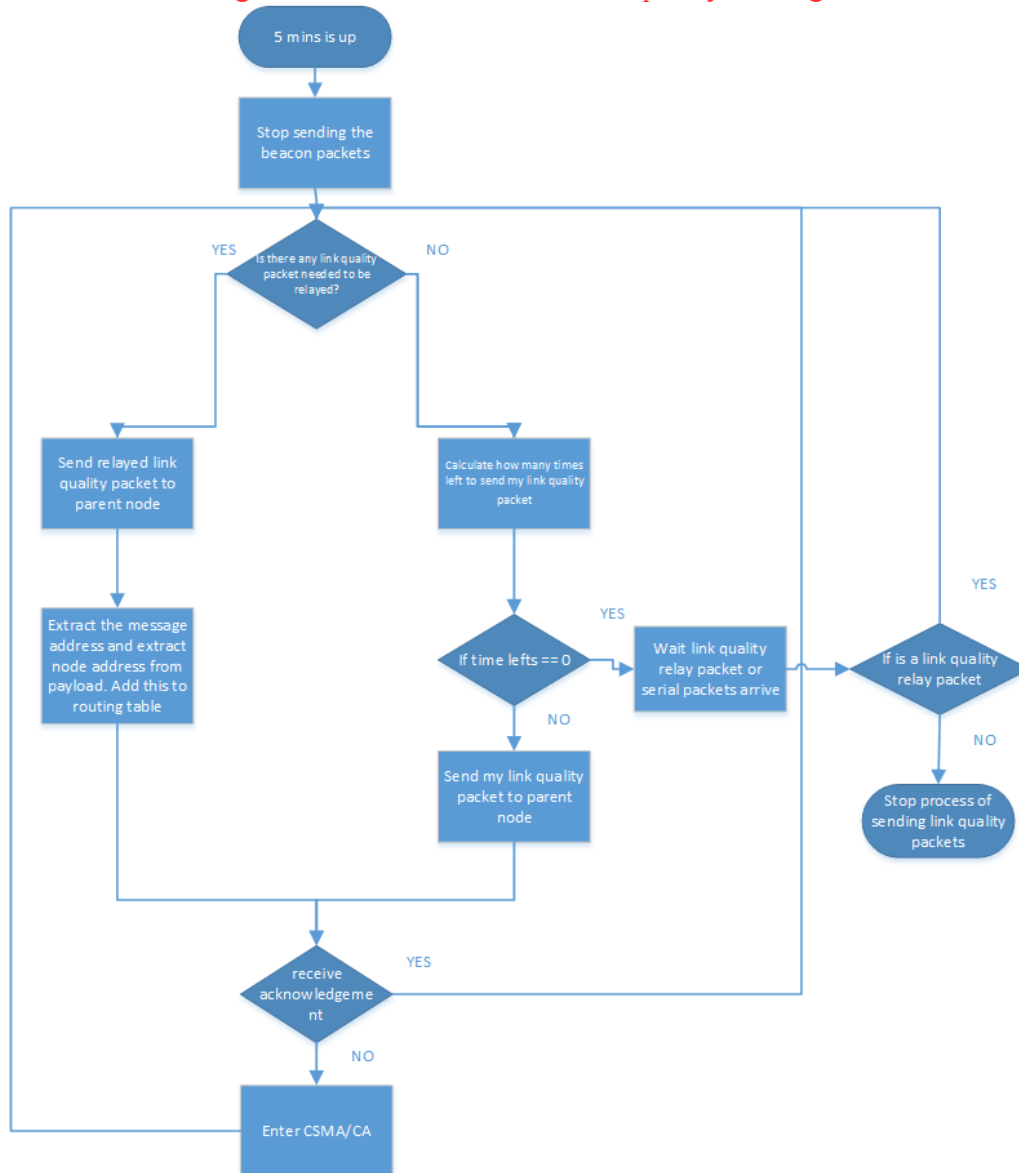


Figure 4.11 algorithm to send the channel quality packets

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.

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

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

When all the link quality packets from all of the nodes are received at the sink, The interface between the Flix 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

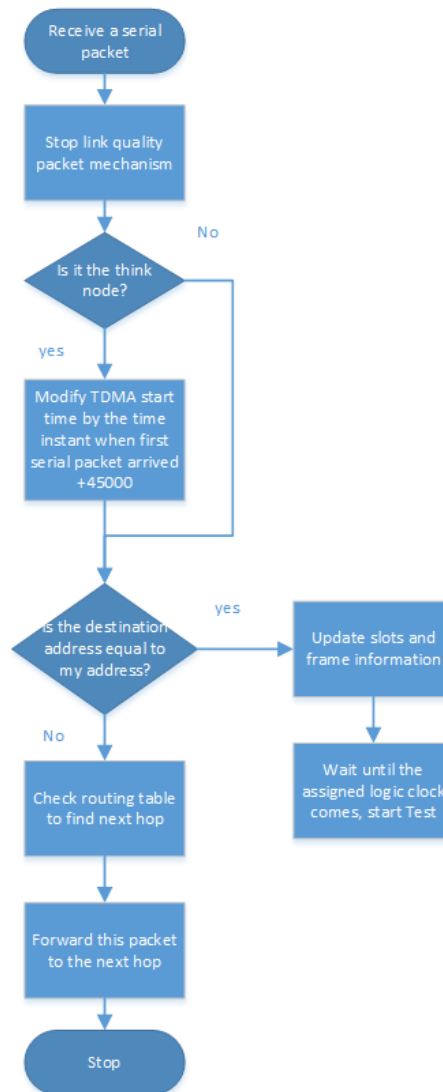


figure 4.13 TinyOS core handles the serial packets and start the test
When all of the nodes in the networks have successfully being scheduled, they enter a state that is ready for testing. The procedure of testing is shown in figure 4.14

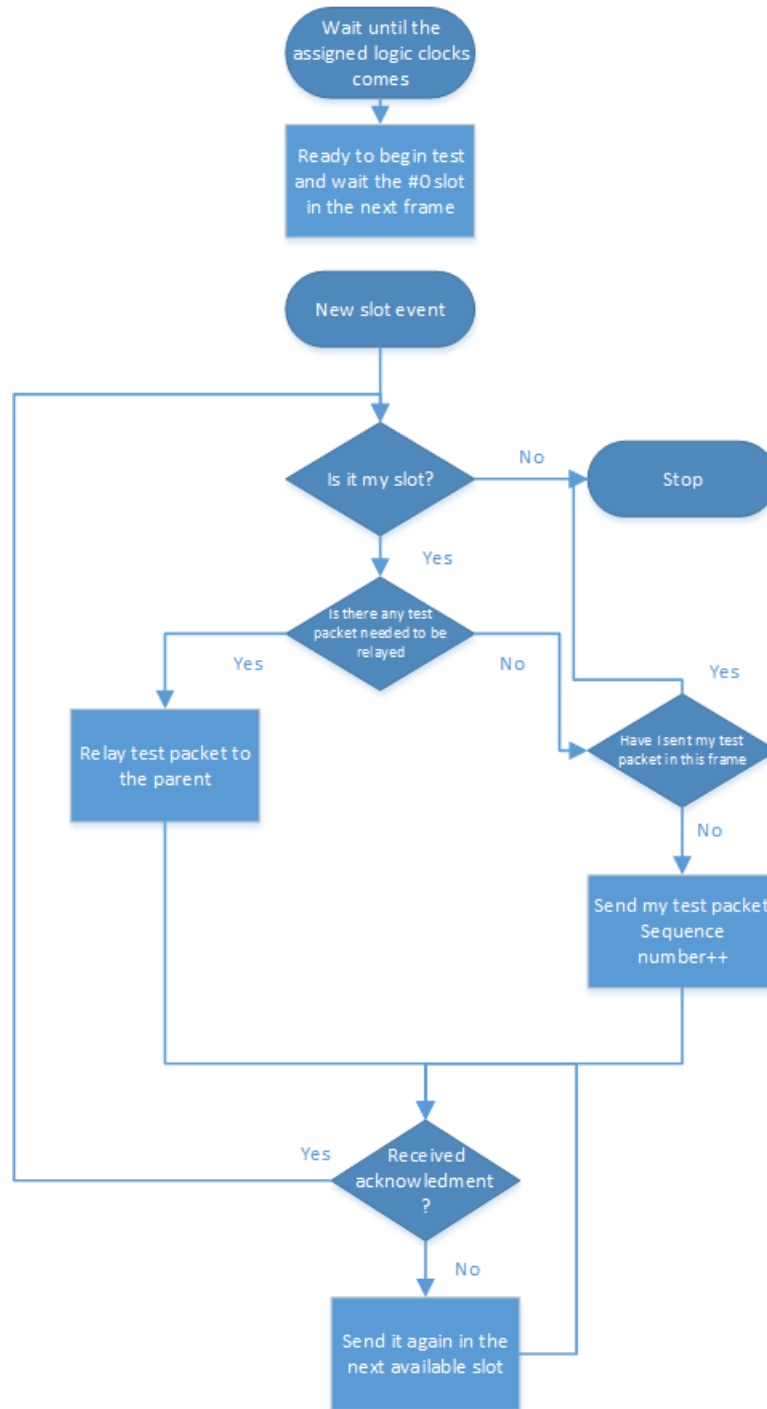


Figure 4.14 The Procedure of testing the routing algorithm

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

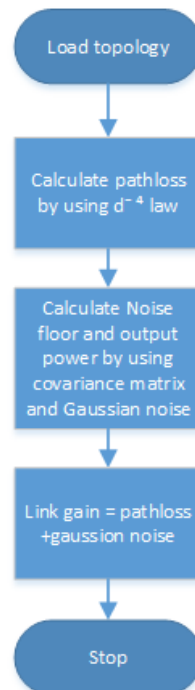
mote address sequence number

figure 4.15 the structure of the testing packet

4.4 **Simulation parameters provider**

4.4.1 **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 California, setting parameters to simulate a building aisle condition and convert the geological coordinates representation into received power representation. The Algorithm of converting process is



shown in figure 4.16.

figure 4.16 the process to covert topology into Tossim parameters

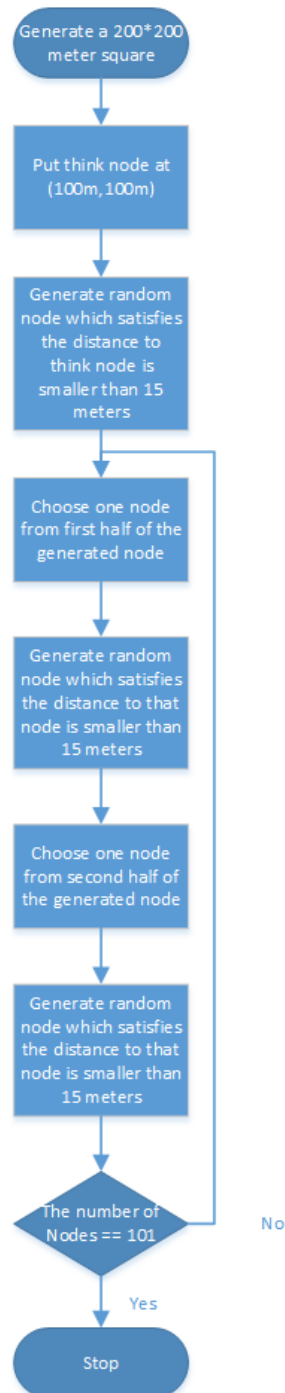
The parameter I used to simulate a building aisle is listed in the Table 4.1

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

Table 4.1 parameters used for Channel Model

4.4.2 **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 node 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

4.5

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 interference around -86dbm. Then I use Tossim live extension to simulate the serial packets (packets that communicates between think node and PC). By writing a Make-File 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

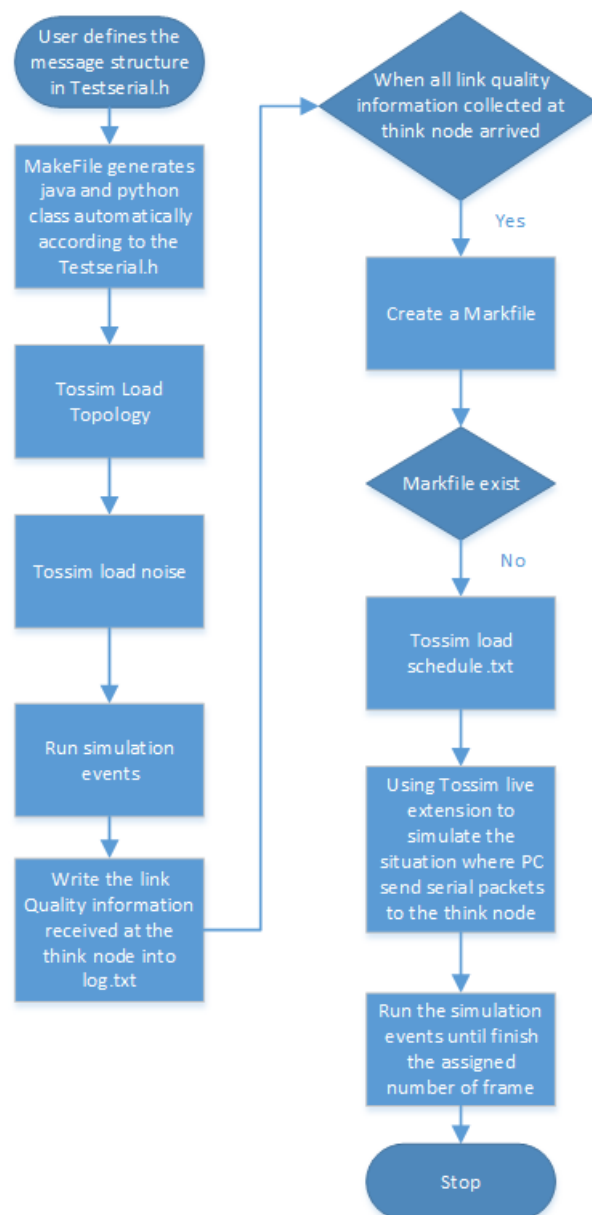


figure 4.18 the procedure of Tossim Emulator

Interface between my wireless sensor network system and Flix 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 Flix routing algorithm in which Flix 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.

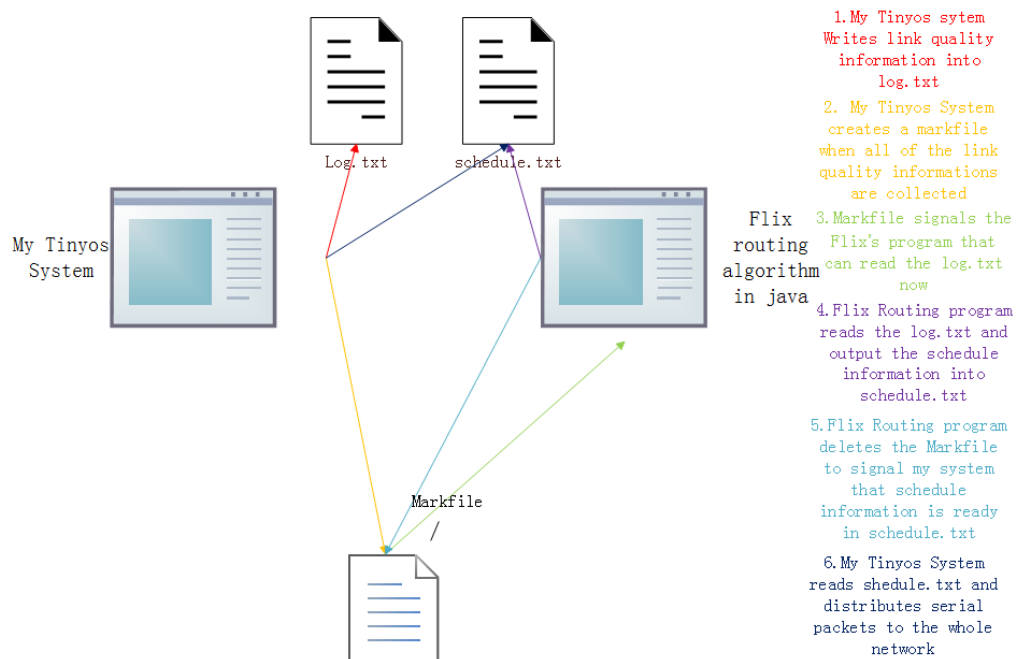


figure 4.19 the interface between My wireless sensor network system and Flix 's Routing algorithm

5 Results

The results chapter is included when you have produced a systematic study, i.e. an evaluation of a program that you have developed, which is required for C - and D-level diploma work. In the results chapter objective results of the empirical study are presented. Keep in mind that possible comments in this chapter should only be used for clarification. Your own views and subjective (personal) comments belong in the chapter conclusion/discussion.

Strive to present the results, for example measurement-, calculations- and/or the simulation result, in a form that is as lucid and easily understandable as possible. The results are preferably presented in diagrams or tables. Accounts of interviews can be summarised, but may include concrete examples supporting your work.

Extensive results, for example complete summaries of survey results, large tables and long mathematical deductions, are placed in the appendices.

6 Conclusions / Discussion

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 program - Flix's new Routing algorithm 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 Flix's new routing algorithm. The worst packet deliver rate is above 0.99 that not only proves the efficiency of Flix's new routing algorithm, but also makes this system seems like more reliable and trustworthy. Since, I have finished all of my overall aims 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, Flix 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 extendability, 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.

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.

Title – Subtitle

Author

2010-07-05

If this chapter is extensive it can be divided up into more chapters or sub-chapters i.e. one analysis or discussion chapter with explanations of and critical assessment of the results, a concluding chapter where the most important results and well supported conclusions are discussed and to sum it up a chapter with suggestions for further research in the same area.

In this chapter it is of vital importance that a connection back to the aim of the survey is made and thus the purpose is pointed out in a summary and analysis of the results.

In this chapter you should also include answers to the following questions: What is the project's news value and its most vital contribution to the research or technology development? Have the project's goals been achieved? Has the task been accomplished? What is the answer to the opening problem formula? Was the result as expected? Are the conclusions general, or do they only apply during certain conditions? Discuss the importance of the choice of method and model for the results. Have new questions arisen due to the result?

The last question invites the possibility to offer proposals to others relevant research, i.e. proposal points for measures and recommendations, points for continued research or development for those wishing to build upon your work.

In technical reports on behalf of companies, the recommended solution to a problem is presented at this stage and it is possible to offer a consequence analysis of the solution from both a technical and layman perspective, for example regarding environment, economy and changed work procedures. The chapter then contains recommended measures and proposals for further development or research, and thus to function as a basis for decision-making for the employer or client.

References

below is an example of an automatically numbered list of references according to the numbered list and cross references method, as described in chapters 2.4:

- [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 Access Control(MAC) and Physical Layer(PHY) Specification, IEEE Std 802.11, 2007.
- [5] communication system, 4th edition, Simon Haykin
- [6] Wireless Communications, 2nd Edition,Andreas F. Molisch, ISBN: 978-0-470-74186-3
- [7] AB Benzlers. *Växelboken. Kuggväxlar, kuggremmar, kilremmar etc.* Mecman. Hydraulik, SKF. Huvudkatalog. (Example of reference to a catalogue)
- [8] Microsoft Corporation, 1992: *Microsoft Word Användarhandbok. Ordbehandlingsprogram för Macintosh Version 5.0.* (Example of ref-erence to a manual)
- [9] Wikipedia, "H.264/MPEG-4 AVC",
http://en.wikipedia.org/wiki/H.264/MPEG-4_AVC
Retrieved 2010-04-21. (Example of reference to a webpage)

Appendix A: Documentation of own developed program code

Example of Appendix subheading