Självständigt arbete på grundnivå

Independent degree project - first cycle

Computer Engineering, 15 credits

Design and implement a modified MAC protocol

Mengjun Qin

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.

MID SWEDEN UNIVERSITY
Department of Computer Engineering

Examiner: Ulf Jennehag, ulf.jennehag@miun.se
Supervisor: Tingting Zhang, tingting.zhang@miun.se
Author: Mengjun Qin, meqi1400@student.miun.se

Abstract

This paper present a modified MAC protocol which aims to give a better performance for critical message in industrial wireless sensor and actuator networks (IWSAN) than WirelessHART standard. With the development of information and technology, wireless sensor network (WSN) has been the replacement of the wired communication in recent years. By using wireless devices, people can easily install it at any time on anywhere, and it also has low power and low cost than the wired devices. Another important feature of wireless sensor network is that it is easier to configure the whole network. So, more and more WSN appears in industrial process control since it has so many benefits. However, there is big challenge in IWSAN which is that if some unpredictable and emergency traffic could not received and handled by the actuator or network manager in time, it maybe lead to economic losses or even lead to a threat to human safety. That is to say that these unpredictable and emergency message have a strict delay and high reliability in IWSN. Therefore, we design and implement this modified MAC protocol as well as evaluation at the end of the paper. The implementation was performed in TinyOS platform and the result shows that it actually can achieve a better performance than WirelessHART standard.

Keywords: MAC protocol, TDMA, Slotted Aloha, WirelessHART, TinyOS, WSN, IWSN

Acknowledgements

First of all, I would like to express my thanks to Prof. Tingting Zhang for her guidance and suggestions in weekly thesis seminars.

I am also want to thank Mehrzad Lavassani for her valuable guidance and without her help, this thesis would not be possible.

Finally, I am indebted to to my parents and my friends, whose support make it possible for me to have the confidence to overcome all the difficulties.

Table of Contents

Terminology

Acronyms

CSMA/CA	Carrier Sense Multiple Access / Collision Avoidance
CAP	Contention Access Period
FFD	Full Function Device
GTS	Guaranteed Time Slots
HART	Highway Addressable Remote Transducer Protocol
IWSN	Industrial Wireless Sensor Networks
ISA	International Society of Automation
LR-PAN	Low-Rate Personal Area Network
MAC	Media Access Control
NM	Network Manager
OSI	Open System Interconnection
RFD	Reduced Function Device
SM	Security Manager
TDMA	Time Division Multiple Access
WIA-PA	Wireless networks for Industrial Automation Process Control
WSN	Wireless Sensor Network

1 Introduction

In the past decades, lots of new technologies have been introduced into the field the process control to make it easier to realize[1]. Wireless is the most popular one. And one of the most important wireless technologies is Wireless Sensor Networks (WSN). Network consists of wireless sensors can bring us grate benefits, such as easy to deployment and installation, also it is quite simple to change the position of the devices to setup a new network. With time goes by, WSN will be more and more popular and more widely used in industrial environment.

Since we talk about industrial automation control, there are some necessary requirements that you must know if you want to control the transmission message in this environment[2]. Compare to other industrial applications of process control, industrial automation control has more demanding requirements, for example, lower delay and higher reliability among devices than any other kinds of applications of process control. What is more, one thing that you can not ignore is that there are more interference in industrial environment than others.

1.1 Background and problem motivation

In the recent years, a large amount of applications have been developed in low power and low cost wireless technology. Zigbee is the one of the most popular application in them. Zigbee [3] is based on IEEE 802.15.4 specification and using carrier sense multiple access with collision avoidance (CSMA/CA) in MAC layer as the multiple access control method, and in such method, it can provide low delay and adequate throughput when the traffic is not heavy. However, CSMA/CA can not provide guaranteed access to the wireless channel and it also can not give a good performance when the number of devices increase. So CSMA/CA is not suitable for industrial automation control which have a strict timing and high reliability as we discussed above.

In order to meet the requirements in industrial automation control, many wireless standard have been proposed, such as WirelessHART, ISA 100a, Wireless networks for Industrial Automation Process Control (WIA-PA) and IEEE 802.15.4e [4]. Generally speaking, there are two kinds of message in industrial automation network, which is the critical message and ordinary message. The critical message, for example, fire alarming must be sent in low latency and high reliability, otherwise it may lead to financial losses or even a threaten to human safety. On such condition, ordinary time division multiple access control method (TDMA) is not enough to give a good performance for those unpredictable and critical message, because unpredictable traffic can not be scheduled and assign a time slot for them to be sent. Therefore, in order to provide the critical traffic with low latency and high reliability, we design and implement this algorithm with respect to give a better performance for critical traffic.

1.2 Overall aim

In industrial wireless sensor and actuator networks, there are some critical message which need to be sent in strict required delay as well as high reliability. But the existing MAC protocol could not achieve this goal with a good performance. So this project's overall aim is to provide such a MAC protocol that can give a low latency and high reliability for those emergency message in industrial wireless sensor and actuator networks.

1.3 Scope

The study has its focus on design a MAC protocol and simulate it in TinyOS. We have not test it by real devices or in real industrial environment, and also, the noise that we use for the simulation is the system original file, this may have a big difference in different environment.

1.4 Concrete and verifiable goals

The main goals that will be achieved in this project are,

1) Study some existing standards and understand how the communication between devices works for each of them.

2) Study TinyOS and nesC and learn the basic concept of each of its components.

3) Be familiar with programming using nesC in TinyOS.

4) Design the algorithm of MAC protocol.

5) Implement the designed protocol in TinyOS.

6) Change some parameters and simulate it again to get more details about the modified MAC protocol.

7) Compare the result of different conditions with different parameters and analyze its performance.

1.5 Outline

Chapter 1 gives a brief introduction to the whole project. It also refer to the motivation of this thesis as well as overall aim and scope of the project. Chapter 2 mainly tells about some theory that related to the project, such as TinyOS, IEEE 802.15.4 standard, WirelessHART and so on, which will be simply introduced in this chapter, and the last part of this chapter is some related work. Chapter 3 will provide the methodology of choosing a better platform and programming language, implementation method, evaluation method and the work flow of the project will be added to the end of this part. For chapter 4, design and implementation of the modified MAC protocol will be stated. The simulation result and discussion will be presented in this chapter 5. The last chapter of this thesis, is going to give the conclusion of the paper and propose some future work.

1.6 Contributions

The algorithm of the protocol is designed by Mehrzad Lavassani, and this project is completed as a part of her master's thesis. The other work including implementation, write script to parse the raw data and plot as well as the evaluation and conclusion is finished by myself.

2 Theory

2.1 Wireless Sensor Network

Wireless sensor network consists of individual nodes, and these node can interact with their environment by sensing or controlling physical parameters. In order to fulfill their tasks, they have to collaborate with each other, wireless communication are used to finish their collaboration. In essence, there are some functionality that nodes must contain within such network, such as, computation, wireless communication, and sensing or control. Since the network also often include actuators, it is also have the name wireless sensor and actuator networks.

One of the most popular applications in WSN is industrial wireless sensor networks (IWSN). In industrial process monitoring and control, when sensor was triggered by environment variables, sensor generate a message and send it to the sink. And after the sink receive the message successfully, since there are so many interference in industrial environment, the transmission may be failed, but if success, the sink will send this message to the network manager to take further actions to handle the event. This is the simplest communication between field devices and network manager, in other case, the message maybe lost or corrupted and need retransmission, and there are some critical message which strict time requirement which is also another concern. Figure 2.1 shows the basic structure of the WSN.

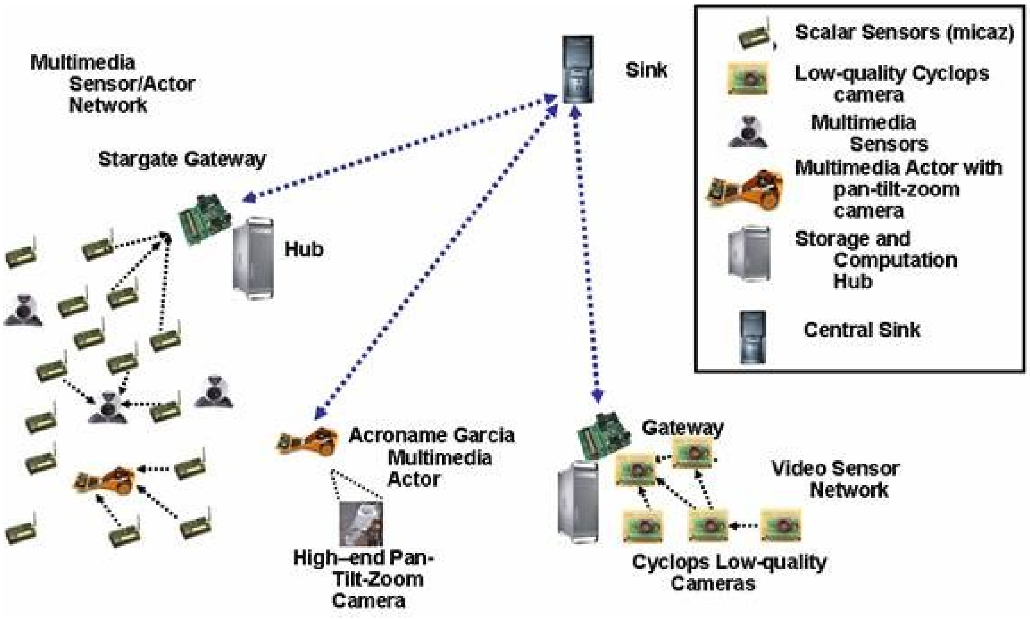

Multimedia Sensor/Actor Network
Stargate Gateway
Hub
Sink
Scalar Sensors (micaz)
Low-quality Cyclops camera
Multimedia Sensors
Multimedia Actor with pan-tilt-zoom camera
Storage and Computation Hub
Central Sink
Acroname Garcia Multimedia Actor
High–end Pan-Tilt-Zoom Camera
Gateway
Video Sensor Network
Cyclops Low-quality Cameras

Figure 2.1 Example of Wireless Sensor Network [5]

2.2 IEEE 802.15.4

IEEE 802.15.4 [6] is a standard that specify the MAC layer and physical layer in low-rate personal area network (LR-PAN). It was defined in 2003 and was maintained by the IEEE 802.15 working group. And many popular applications in wireless sensor network are based on this standard, such as Zigbee, ISA 100.11a, WIA-PA, and WirelessHART. Since IEEE 802.15.4 only define the MAC layer and physical layer, these new specifications further extend the standard by developing upper layers that is not specified in IEEE 802.15.4. Figure 2.2 shows the IEEE 802.15.4 protocol stack.

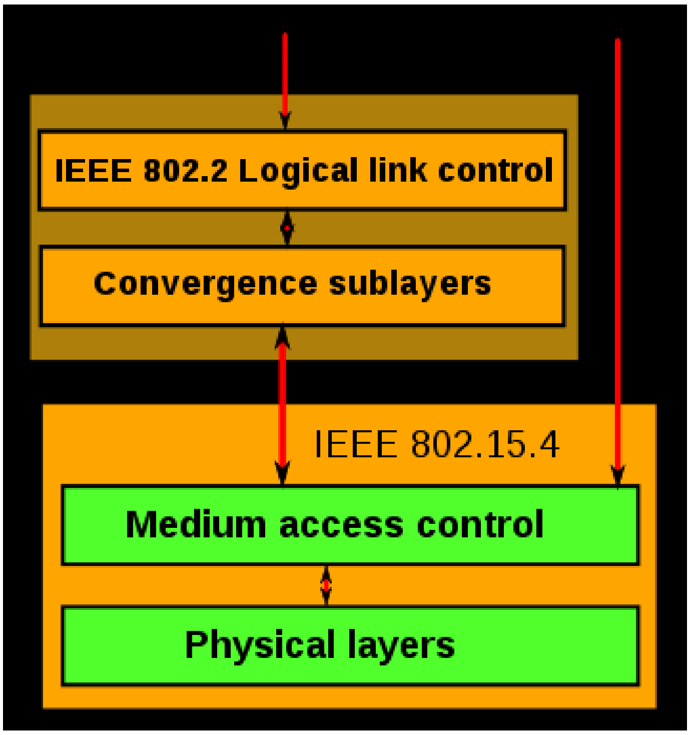


Figure 2.2 IEEE 802.15.4 protocol stack [8]

2.2.1 Physical layer

Physical layer is the first layer in the Open System Interconnection model (OSI model). It is used for data transmission and provide the interface to physical layer management entity, which have access to every layer management function. The main function of the physical layer is that it can perform channel selection and energy and signal management. And there are three unlicensed frequency band for it work on [6]:

(1) 868.0–868.6 MHz: Europe, allows one communication channel (2003, 2006, 2011)

(2) 902–928 MHz: North America, up to ten channels (2003), extended to thirty (2006)

(3) 2400–2483.5 MHz: worldwide use, up to sixteen channels (2003, 2006)

2.2.2 MAC layer

MAC layer is the second layer in the OSI model. It mainly specify when a node can transmit message. Thus, the main task of MAC layer are [7],

(1) slot synchronization

(2) identify the node which needs to access the medium

(3) propagation of message received from network layer

(4) listen for packet from neighbors

There are two kinds of nodes which the IEEE 802.15.4 standard specified on the MAC layer [9],

a. A Full Function Device (FFD): It can operate in three roles, a Personal Area Network (PAN) coordinator, a simple coordinator or a device.

b. A Reduced Function Device (RFD): It can only operate as a device.

A device must connect to a coordinator node (which must be a FFD) and only communicate with this coordinator. All coordinators make up of the PAN, and one of the coordinators is designed to be the PAN coordinator. There is a mode called beaconed mode that the coordinator operates in to manage channel access and data transmission with the help of a superframe structure displayed in Figure 2.3.

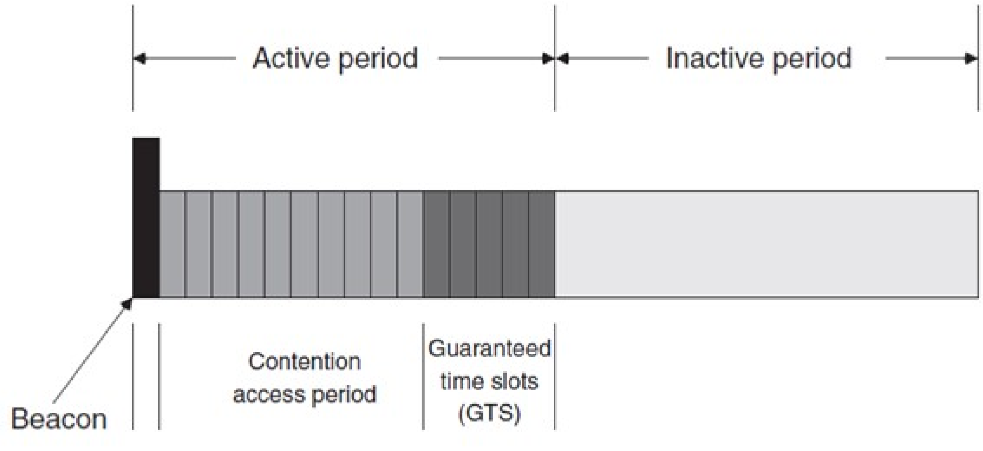


Figure 2.3 Superframe structure of IEEE 802.15.4 [10]

The length of all superframe are the same. The coordinator broadcast a frame beacon packet to start a superframe. The frame beacon contains a superframe specification which specify the length of the active period and inactive period. In inactive period, a node can go into sleep and must wake up before the inactive period ends. In active period, the first time slot is the beacon, and next follows a Contention Access Period (CAP) which means all nodes have the chance to transmit their message in this period, and next is a number of continuous Guaranteed Time Slots (GTS).

2.2.3 Zigbee

Zigbee [3] is a low power local area network protocol and it is based on IEEE 802.15.4 standard. According to the international specification, Zigbee is show-distance and low-power wireless communication technology. The main features of the Zigbee are short distance, less complexity, self-organization, low cost and low data transmission rate. And it is mainly used in automation control as well as the field of remote process control, also it can be embedded in many kinds of devices.

Zigbee is very similar with bluetooth, they are both emerging short-distance wireless communication technology. But for industrial automation control, bluetooth is too complicated and cost much as well as can not applied in large network. So people came up with Zigbee. Comparing to bluetooth, it has less complexity and less cost. Although Zigbee has so many advantages, it still can not meet the strict requirements for industrial applications. First of all, Zigbee can not guarantee a required time delay bound which is talked in the previous in the chapter. And the next problem is that it has no built-in channel hopping, and this may lead to lots of failure in transmission as there are so many interference in industrial environment.

2.2.4 ISA 100.11a

ISA 100.11a is a wireless networking technology standard developed by International Society of Automation (ISA) [11]. It is one of the most common four wireless sensor network standard. ISA 100.11a aims to support industrial applications with low complexity, reasonable cost, low power and suitable data rate for industrial wireless devices. What is more, ISA 100.11a is the first open and designed for a variety of industrial applications standard. It has the following features:

(1) Provide process industrial applications services, including factory automation.

(2) Global deployment.

(3) Provide level one to level five application.

(4) Guarantee the inter-cooperation of equipments from different manufactures.

(5) Provide simple, flexible and safe method for the main industrial threaten of IEEE 802.15.4-2006.

2.2.5 WirelessHART

WirelessHART is a wireless sensor network technology based on the Highway Addressable Remote Transducer Protocol (HART) [12]. It works on the frequency of 2.4 GHz in the ISM radio range and nodes in WirelessHART network communicate through Time Division Multiple Access (TDMA) manner. In this manner, time was divided into slots, and the duration of each slot is 10ms, which means that node inside the network has to finish a complete communication with other kinds of devices in 10ms. The time slot was scheduled by network manager, every node will get its own slot and transmit message in this slot without collisions from other nodes. A set of slots make up of superframe for data transfer. There are mainly three kinds of components in WirelessHART network:

(1) Field device: finish concrete functions and communicate with gateway

(2) Gateway (also called Access Point): connection between field device and network manager, manage the routing information as well.

(3) Network manager: configure the whole network, perform scheduling and synchronization and so on.

Figure 2.4 shows basic structure of WirelessHART network.

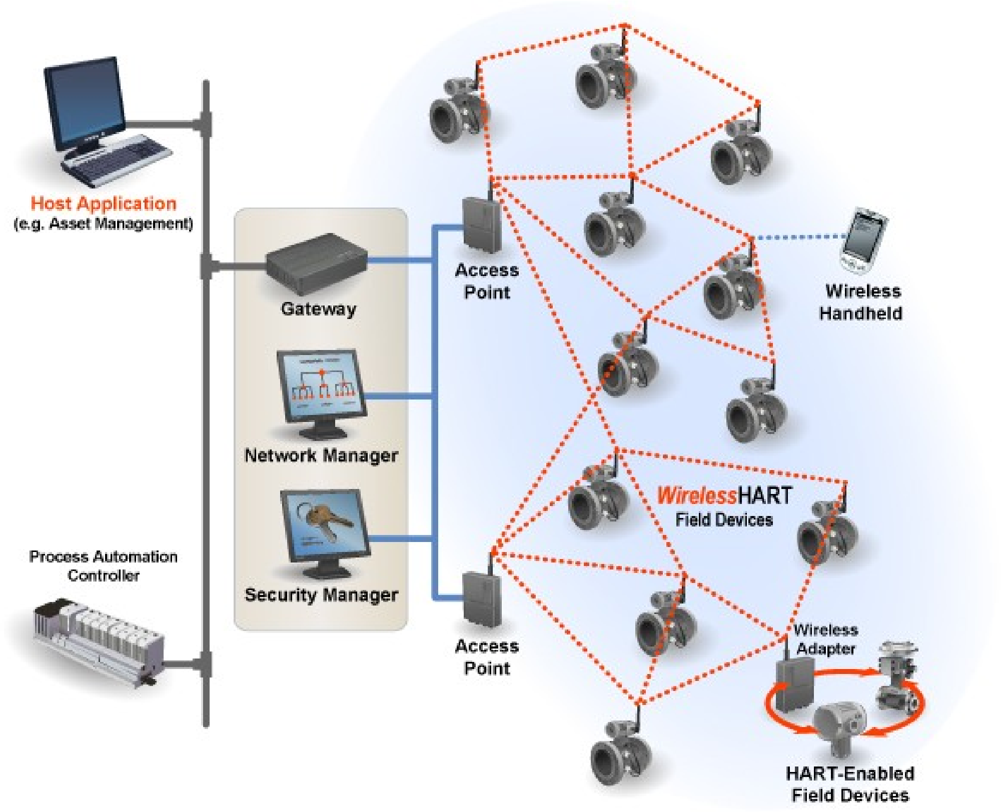


Figure 2.4 Basic structure of WirelessHART network [13]

2.2.6 WIA-PA

Wireless Networks for Industrial Automation Process Automation (WIA-PA) standard which is maintained by China Industrial Wireless Alliance is a substandard for the domain of process automation control. And it is based on IEEE 802.15.4 standard, mainly applied in industrial process measurement, monitoring and control. WIA-PA specifies five types of physical devices:

a. Host computer: An interface through which users and maintenance/management personnel perform transactions to the WIA-PA network and the management networks.

b. Gateway device (GW): A device connecting the WIA-PA network and other plant networks with the functions of protocol translation and data mapping.

c. Routing device: A device forwarding packets from one network device to another in the WIA-PA network.

d. Field device: A device installed in the industrial field, which is connected to or controls processes such as sensors, actuators, etc.

e. Handheld device: A portable device that is responsible for configuring network devices and monitoring network performance.

WIA-PA also specifies two types of uppermost logical devices:

a. Network manager (NM): Responsible for configuring the network, scheduling communication between routing devices, managing the routing table, and monitoring the performance of the whole network.

b. Security manager (SM): Responsible for configuring the security mechanism, managing the security key, and authenticating routing devices and field devices.

Figure 2.5 shows a simple example of WIA-PA network.

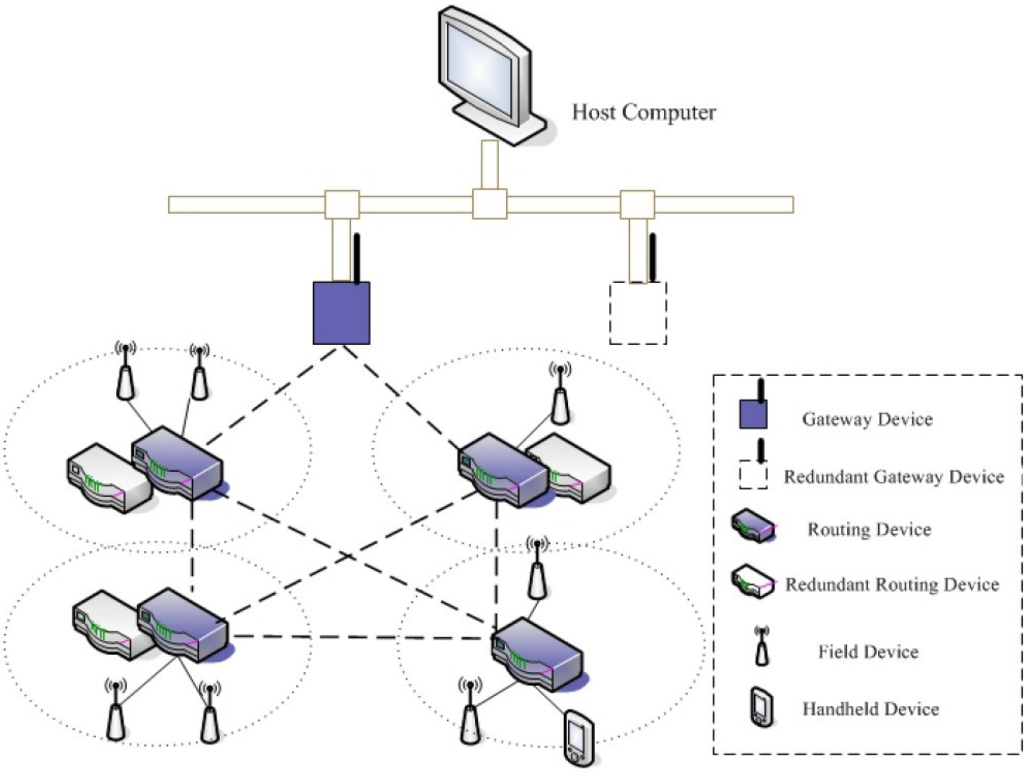


Figure 2.5 Example of WIA-PA network [14]

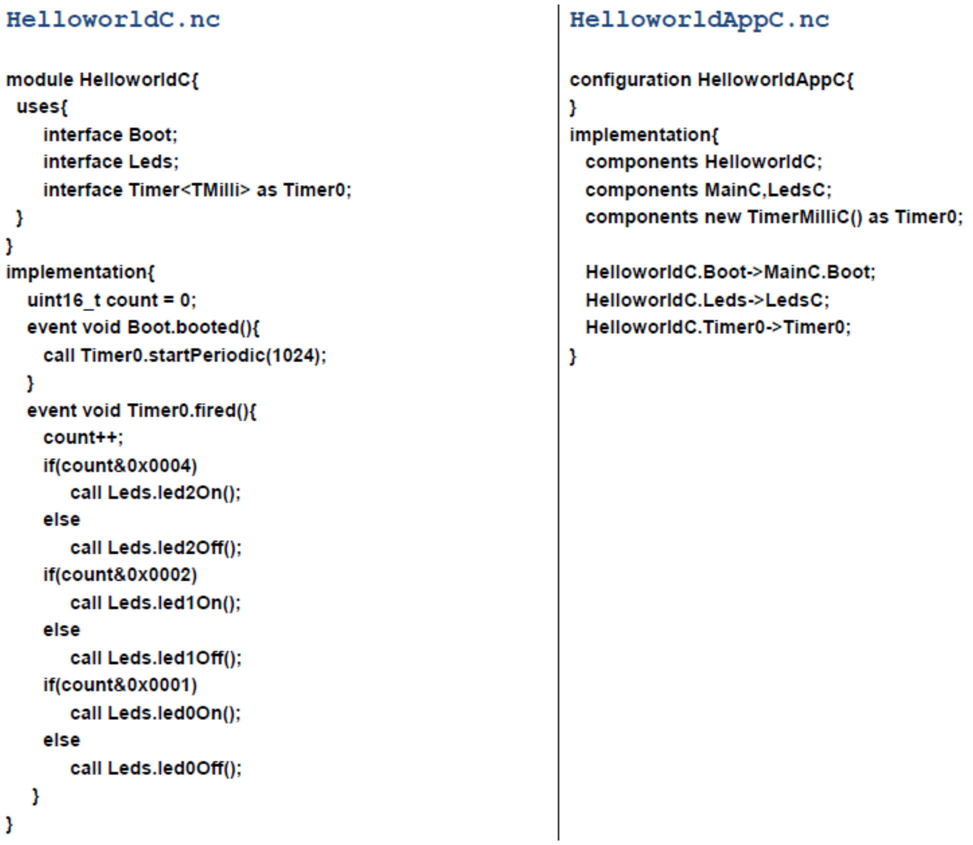
2.3 TinyOS and nesC

TinyOS is an open source, BSD-licensed operating system, and it is designed for low power wireless devices, for example, those used in sensor networks, ubiquitous computing, personal area network, smart buildings and smart meters [15]. It is the result of combination of a large amount of projects which are developed by nesC. TinyOS applications are written in nesC which is a dialect of C language optimized for the memory limits of sensor networks.

In Wireless Sensor Networks, since energy consumption determines sensor node lifetime, sensor nodes tend to have very limited computational and communication resources. As a result, software needs to be very efficient, both in terms of CPU cycles and in terms of memory use. TinyOS is a lightweight operating system and has very aggressive system and mechanisms for saving power and memory. TinyOS makes building sensor networks applications easier. It provides a set of important services and abstractions and defines a concurrent execution model, so that developers can build applications out of reusable services and components. What's more, TinyOS supports a dozen generic platforms and its structure makes it reasonable to port and to new platform.

2.3.1 NesC

NesC is a component-based C dialect programming language [16]. In nesC, programs are built out of a set of cooperating components. Components are the basic program units. NesC compiler connects and compile this components as a single unit. There are two kind of components in nesC, modules and configurations, both of which are composed of specification and implementation. In module components, the first step is define all the interfaces that is going to use in the whole program, and then create variables and write functions to deal with all events that generated in the program running time. Configurations connect components into larger abstractions. For one component to be able to call another, we have to map a set of names in one component's specification to a set of names in another component specification. And this is called wiring. The configuration part is made up of these wirings. Figure 2.6 show a simple example in nesC.



```
HelloworldC.nc

module HelloworldC{
 uses{
    interface Boot;
    interface Leds;
    interface Timer<TMilli> as Timer0;
 }
}
implementation{
   uint16_t count = 0;
   event void Boot.booted(){
     call Timer0.startPeriodic(1024);
   }
   event void Timer0.fired(){
     count++;
     if(count&0x0004)
        call Leds.led2On();
     else
        call Leds.led2Off();
     if(count&0x0002)
        call Leds.led1On();
     else
        call Leds.led1Off();
     if(count&0x0001)
        call Leds.led0On();
     else
        call Leds.led0Off();
   }
}
```

```
HelloworldAppC.nc

configuration HelloworldAppC{
}
implementation{
  components HelloworldC;
  components MainC,LedsC;
  components new TimerMilliC() as Timer0;

  HelloworldC.Boot->MainC.Boot;
  HelloworldC.Leds->LedsC;
  HelloworldC.Timer0->Timer0;
}
```

Figure 2.6 Hello world example in nesC [17]

2.4 Related work

Since WSN has been more and more popular in industrial automation control, plenty of protocols have been explored for critical message in industrial environment. The author in [19] design a work-conserving scheduling algorithm for GTS in order to meet the strict delay requirements. But the algorithm is based on CSMA/CA and may cause high collision which is not suitable for critical message in IWSN. In [20], the author provide a Time-Division Cluster Scheduling mechanism to meet all end-to-end deadlines as well as minimizing the energy consumption. And in [21], the author propose a distance-constrained real-time offline message-scheduling algorithm which both make scheduling for periodic messages and specify parameters for superframe to meet the timing constrains. However, this two papers do not distinguish between ordinary message and critical message. The modified MAC utilizes these protocols as the baseline and provides enhanced access method for emergency message in industrial automation control.

3 Methodology

To achieve the main goal of this thesis, we use several different methods. The main three methods we used to accomplish the goal is choosing a suitable platform, implementation method and evaluation method. At last, the workflow of the project is added to the methodology.

3.1 Platform

There are so many embedded operating system were developed for Wireless Sensor Networks (WSN), for example, TinyOS, Contiki, MANTIS, Nano-RK, LiteOS [18] and so on. But we choose TinyOS at last, because it is not only the most popular operating system for WSN, but also for the reasons we referred to in the theory chapter.

3.2 Implementation method

In order to implement the modified MAC protocol and analyze its performance in different situations, we design four scenarios. This first two is designed for the modified MAC protocol, and the main difference is that it has different number of nodes for transmitting emergency message. The rest two is designed for WirelessHART which transmit message in slotted aloha manner, and their difference is the same with the two scenarios above.

3.3 Evaluation method

Choosing proper evaluation parameters is a very important part in the whole project. First of all, it gives you a criterion to evaluate algorithm that you have designed. And next, by these parameters you can comparing the result of different scenarios.

After get the result from simulation, we choose the following parameters as our evaluation methods:

(1) End-to-End delay: time between a node receives event and send it successfully to the gateway.

(2) The maximum, minimum and average value of End-to-End delay.

(3) Percentage of received critical packet: the percentage of successful critical packets among all critical packets.

(4) Percentage of received events to the sink: the percentage of successful packets (include critical packets and ordinary packets) among all packets generated.

Because delay and reliability are the most important point in the whole protocol and these parameter can show what the delay and reliability actually is after the simulation.

3.4 Workflow

Figure 3.1 illustrate the workflow of our project.

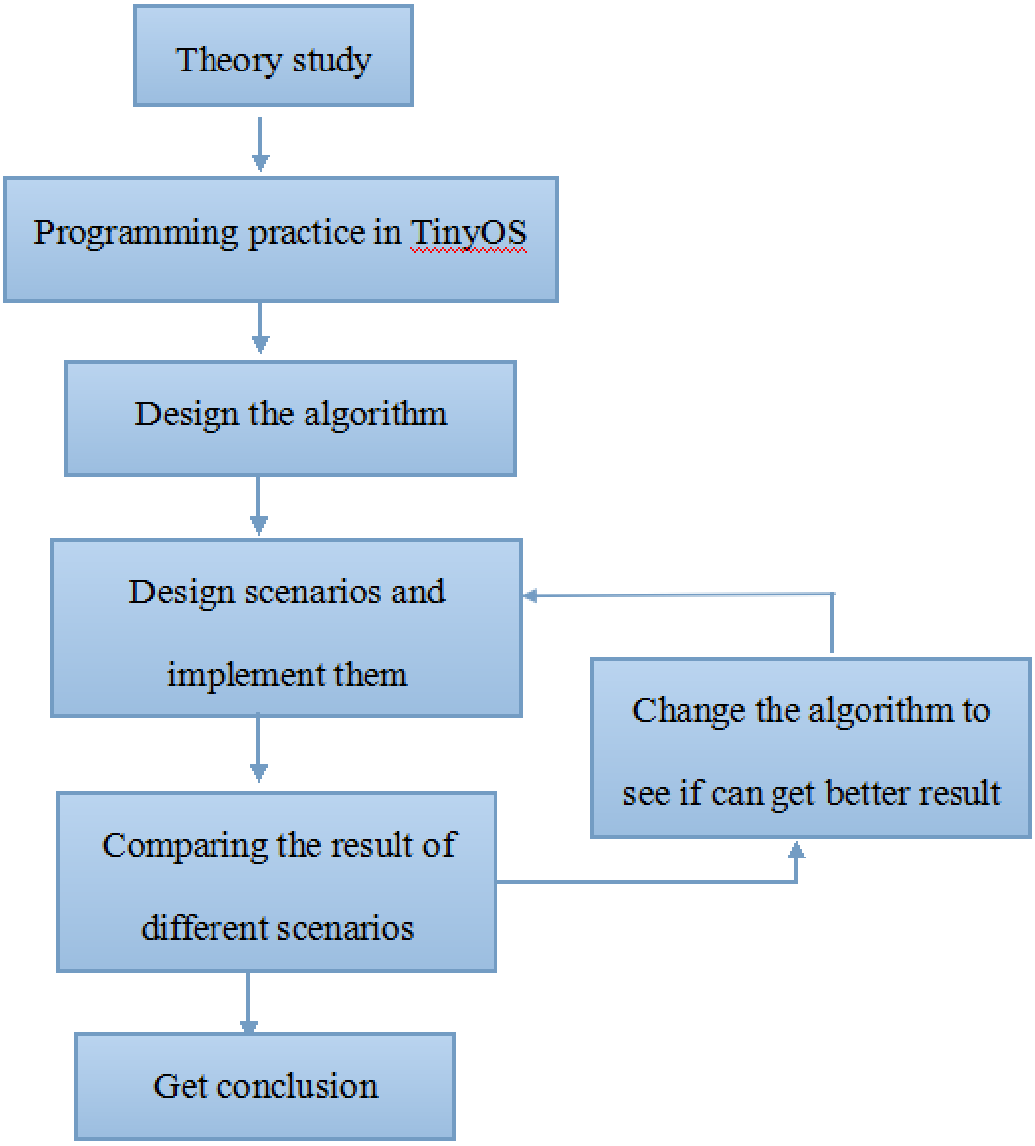

Theory study
Programming practice in TinyOS
Design the algorithm
Design scenarios and implement them
Comparing the result of different scenarios
Change the algorithm to see if can get better result
Get conclusion

Figure 3.1 Workflow of the project

4 Design

4.1 Design

This part mainly describes the basic idea of the modified MAC protocol and how it works.

4.1.1 Basic idea of the protocol

As we have discussed the MAC layer of the IEEE 802.15.4 standard in the theory chapter, there are contention access period and guaranteed time slots in active period. In WirelessHART standard, slotted aloha is used as the method for node to transmit message in contention access period, while TDMA is used in guaranteed time slots. But in industrial automation control environment, this may lead to a high delay as well as low reliability for those unpredictable critical message since there existed collision in CAP and the long waiting time in GTS.

Because of the unpredictable of the critical message, TDMA is not enough. The basic idea of the modified MAC protocol is that, instead of arranging only one slot for every node in GTS, we put subslot between slots as showed in Figure 4.1.

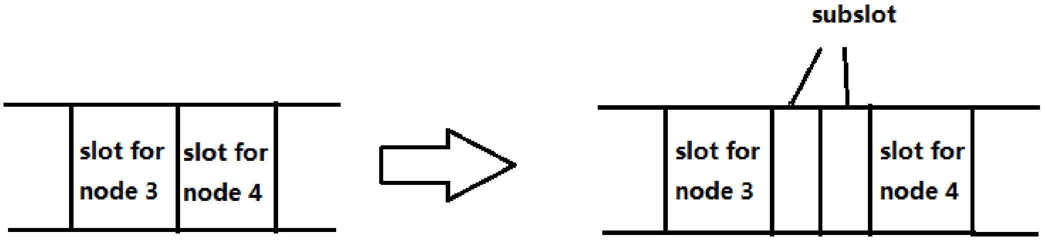


Figure 4.1 Slot and subslot

The subslot is also scheduled for node to transmit message, but only critical message can be transmitted at this time. Another change that we make is that we add a relay node at the end of the GTS. When a critical message sent from node is not received successfully by the coordinator, this message will be sent to the relay node but only once, then when it is time for relay node to transmit message, relay node will aggregate all packets it has received into a single one and send it to coordinator. These measures seems to increase the length of superframe, but it gives more chance for critical message to transmit at any time. And finally one more important thing is that critical message has higher priority than ordinary message in the protocol. If there existed a critical message and a new ordinary message comes, this ordinary message will be lost. But if there existed a critical message and a new critical message comes, the ordinary will also be covered which means lost.

4.1.2 Flow chart of the algorithm

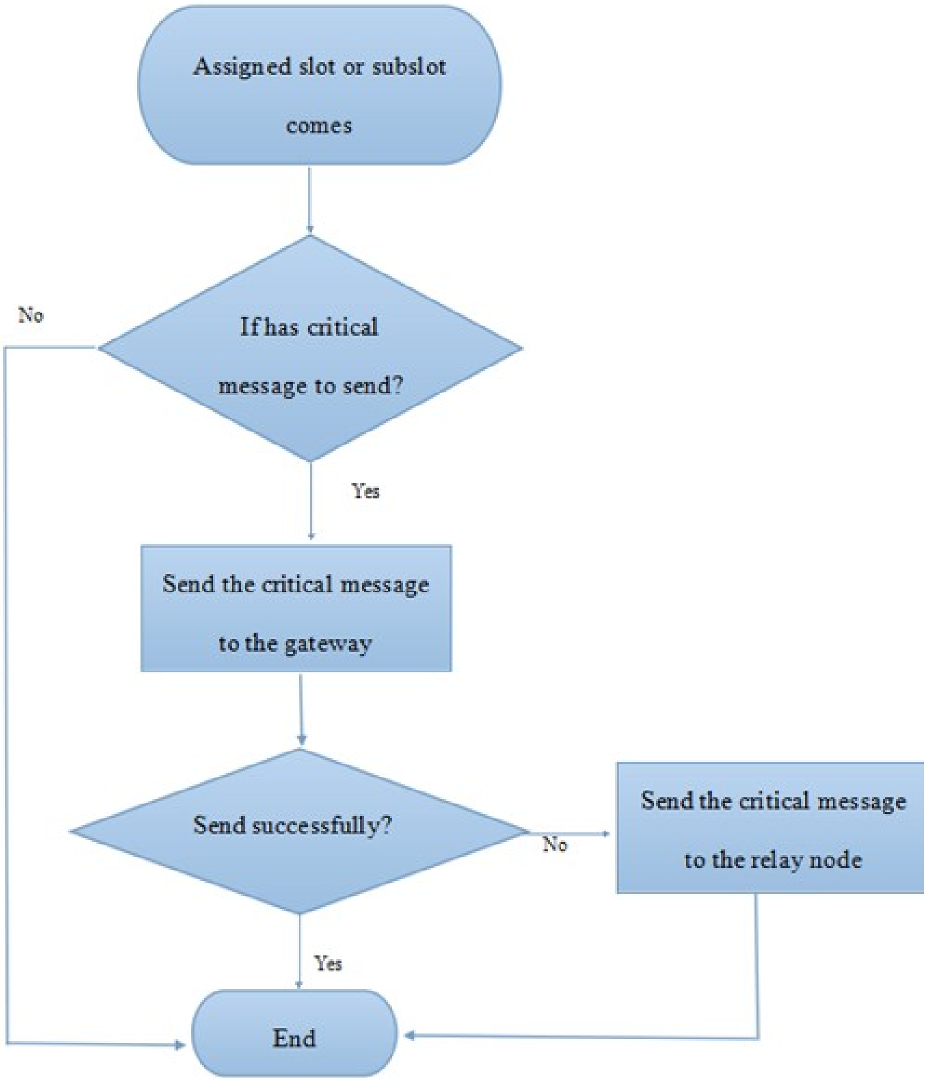

Assigned slot or subslot comes
If has critical message to send?
No
Yes
Send the critical message to the gateway
Send successfully?
No
Send the critical message to the relay node
Yes
End

Figure 4.2 Flow chart for ordinary node

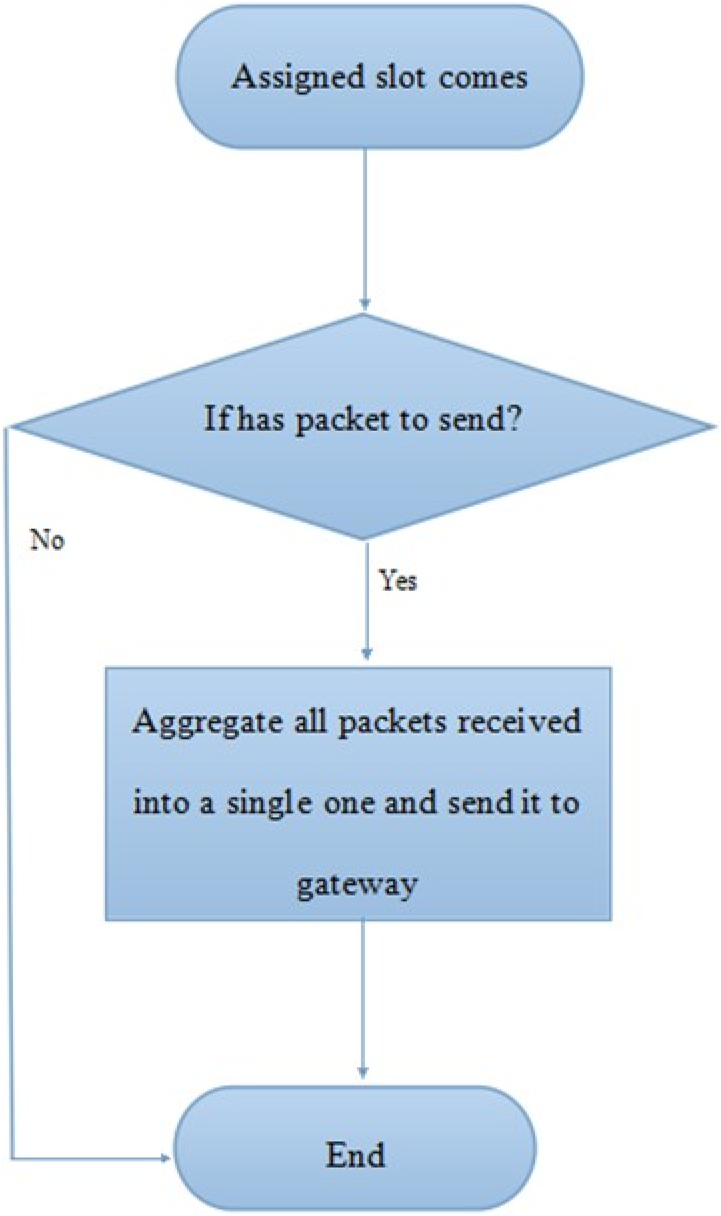

Assigned slot comes
If has packet to send?
No
Yes
Aggregate all packets received into a single one and send it to gateway
End

Figure 4.3 Flow chart for relay node

4.2 Implementation

There are some work which need to be done before implementation. First of all, a simple network is needed for simulation. Figure 4.4 shows a basic topology of the simulation. Here are some illustrations about the topology.

a. Node# 0: Sensor, generate emergency event randomly and broadcast it to all other nodes except relay node and gateway.

b. Node# 1: Gateway, receive message from all other nodes in the network except sensor, and communicate with network manager to organize the network.

c. Node# 2-29: Node, receive emergency event from sensor and generate ordinary event by themselves. They can communicate with both relay node and the gateway.

d. Node# 30: Relay node, receive message from ordinary node (Node# 2-29), aggregate them into a single packet and send it to gateway. The position of its slot is at the end of the superframe.

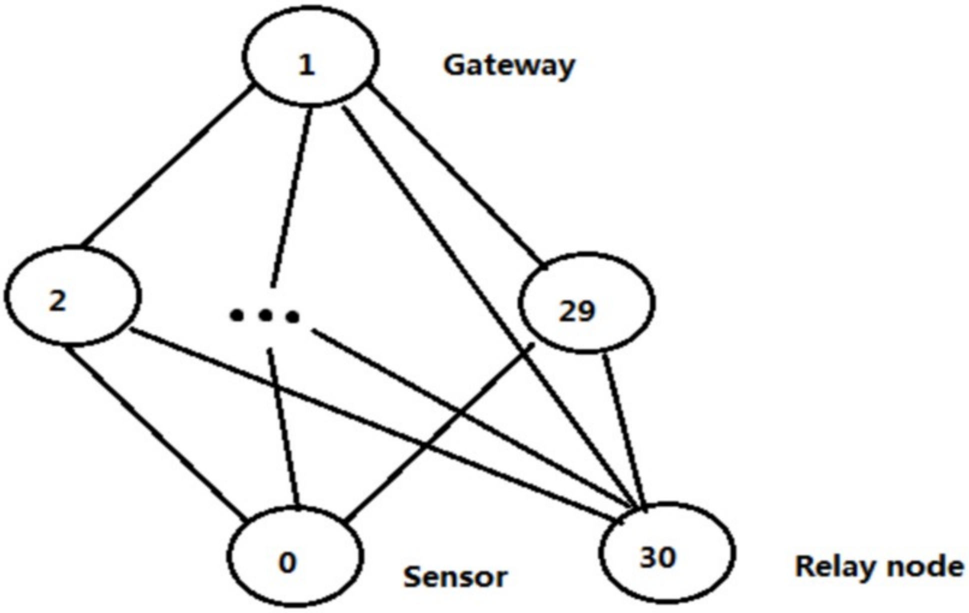


Figure 4.4 Basic topology of the simulation network

In order to reduce complexity, we do not care about the CAP and inactive period since we can not make scheduling for these two part. So, as Figure 4.5 illustrated, our superframe structure is just like this.

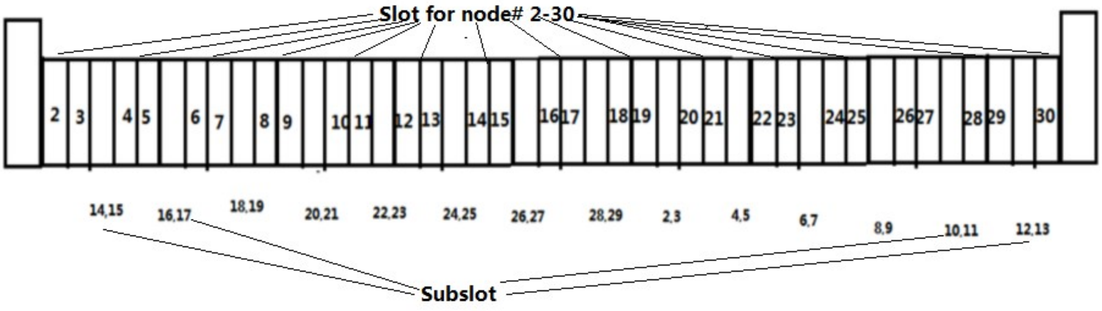


Figure 4.5 Superframe structure of modified MAC protocol

The first slot of the superframe is a beacon frame, which announcing the PAN identifier, a list of outstanding frames and other parameters, but in our simulation, we do not specify its content and just leave it there to remind us that a new superframe comes. Next following the allocation of slot and subslot for every node.

4.2.1 Implementation in TinyOS

In order to implement the modified MAC protocol, four scenarios are deigned as it has been discussed in the implementation method of methodology. Table 1 shows the setting for each scenario.

SF: duration of superframe TS: duration of time slot SS: duration of subslot ST: duration of shared slot

Table 1. Scenario setting

Scenario #	Number of node	Number of node receive emergency packet	Algorithm	SF (ms)	TS (ms)	SS (ms)	ST (ms)
1	31 (0-30)	28 (2-29)	Modified MAC	440	10	5	No
2	31 (0-30)	9 (2-10)	Modified MAC	440	10	5	No
3	30 (0-29)	28 (2-29)	WirelessHA RT	440	10	No	10
4	30 (0-29)	9 (2-10)	WirelessHA RT	440	10	No	10

For scenario 1, Figure 4.4 shows the topology and Figure 4.5 shows the superframe structure. The only difference between scenario 1 and scenario 2 is that sensor only connect to node# 2-10 for scenario 2. For scenario 3, Figure 4.6 shows the topology and Figure 4.7 shows the superframe structure. And the difference between scenario 3 and scenario 4 is the same as difference between scenario 1 and scenario 2.

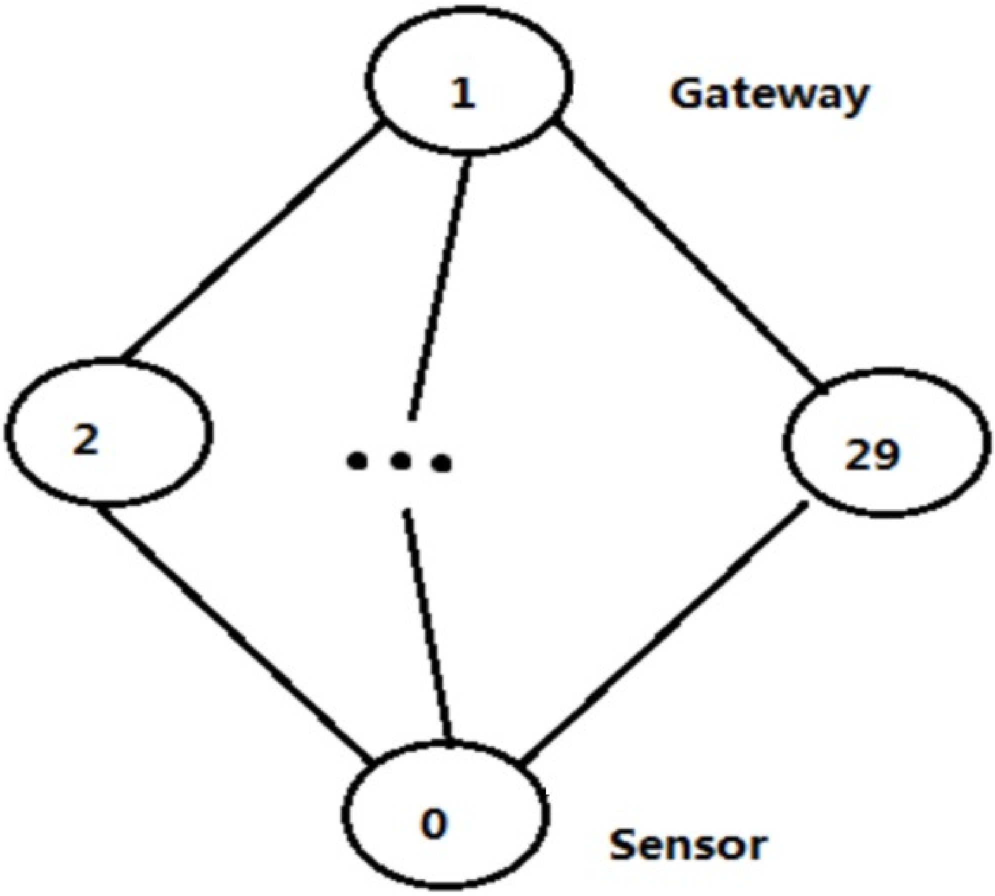


Figure 4.6 Topology of Scenario 3

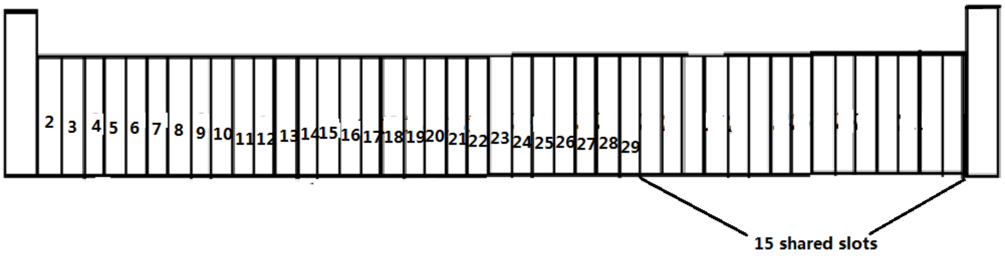
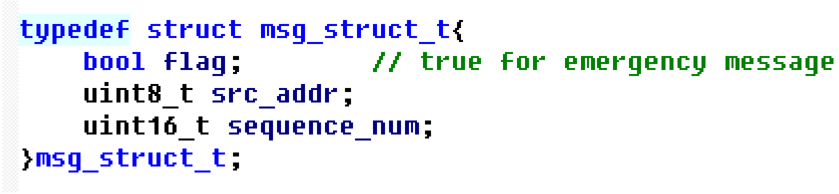


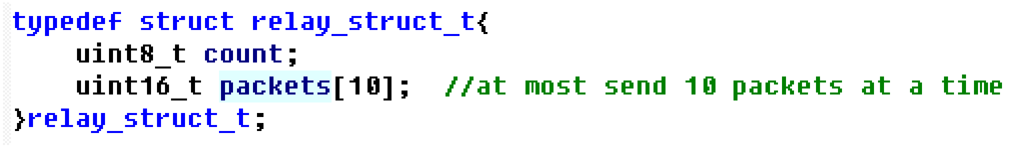
Figure 4.7 Superframe structure of Scenario 3 and 4

There are two kinds of data structure defined for the algorithm, and they are illustrated by Figure 4.8 and Figure 4.9.



```
typedef struct msg_struct_t{
    bool flag;        // true for emergency message
    uint8_t src_addr;
    uint16_t sequence_num;
}msg_struct_t;
```

Figure 4.8 Structure of message

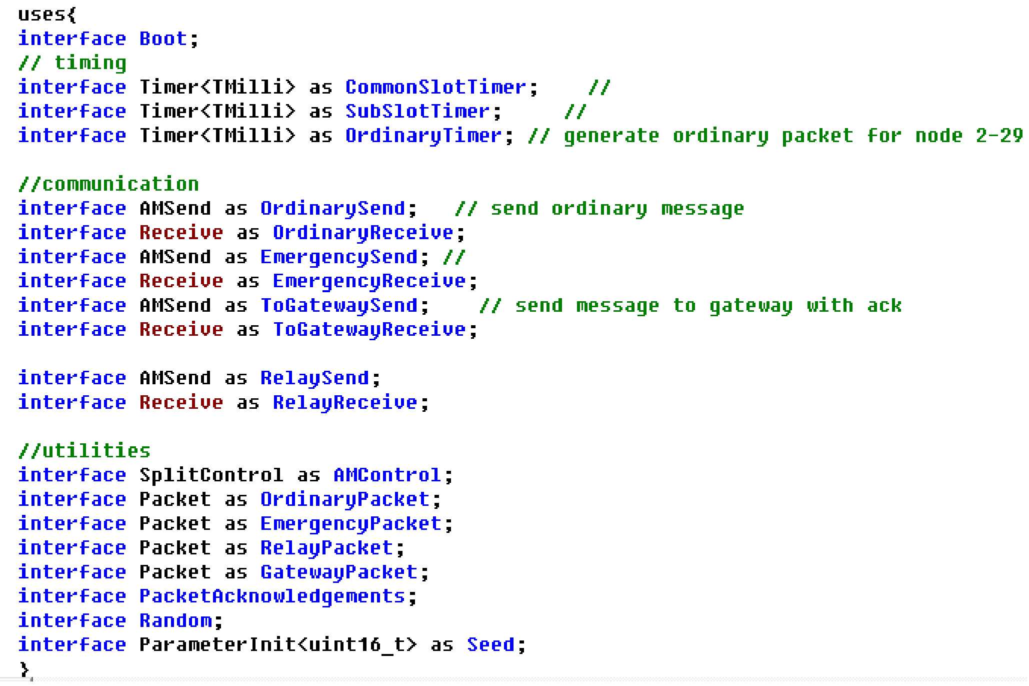


```
typedef struct relay_struct_t{
    uint8_t count;
    uint16_t packets[10];  //at most send 10 packets at a time
}relay_struct_t;
```

Figure 4.9 Structure of message sent from relay node

Figure 4.8 is the structure of both emergency packet and ordinary packet, and they are distinguished by the flag variable. And another two variables specify the source address and the sequence number of the packet. Figure 4.9 shows the packet structure sent from relay node to the gateway, the first variable count define how many packet will be send and the content of the array is the sequence number of each packet.

Since nesC is a component-based language, in order to make the scheduling for each node, we use the component Timer and the interface startPeriodicAt() to specify when a node can start to transmit message. And for communication, we use the component AMSender to send message and component Receiver to receive message. A node must know whether the receiver has receive the message successfully, TinyOS provide us the interface PacketAcknowledgements and the function requestAck() and wasAcked() to tell the sender if the packet was acknowledged by the receiver. And we use the acknowledgements between gateway and nodes including the relay node. But in order not to add extra delay, we do not use acknowledgements between sensor and node, as well as between ordinary node and relay node. All interfaces used are illustrated in Figure 4.10.



```
uses{
interface Boot;
// timing
interface Timer<TMilli> as CommonSlotTimer;    //
interface Timer<TMilli> as SubSlotTimer;     //
interface Timer<TMilli> as OrdinaryTimer; // generate ordinary packet for node 2-29

//communication
interface AMSend as OrdinarySend;   // send ordinary message
interface Receive as OrdinaryReceive;
interface AMSend as EmergencySend; //
interface Receive as EmergencyReceive;
interface AMSend as ToGatewaySend;    // send message to gateway with ack
interface Receive as ToGatewayReceive;

interface AMSend as RelaySend;
interface Receive as RelayReceive;

//utilities
interface SplitControl as AMControl;
interface Packet as OrdinaryPacket;
interface Packet as EmergencyPacket;
interface Packet as RelayPacket;
interface Packet as GatewayPacket;
interface PacketAcknowledgements;
interface Random;
interface ParameterInit<uint16_t> as Seed;
}
```

Figure 4.10 All interfaces

5 Results

In this part, the simulation result of all four scenarios will be presented. And the discussion of result will also be presented.

5.1 Scenario# 1 & Scenario# 3

Scenario 1 and Scenario 3 are both all nodes can send emergency packet, so it is reasonable to put their results together and compare with each other. Figure 5.1 shows the distribution of end-to-end delay for scenario 1 and scenario 3. From Figure 5.1, we can see that before about 290 ms in x axis, the percentage of received packets in modified MAC is much higher than the case in WirelessHART. And after that, they are almost the same.

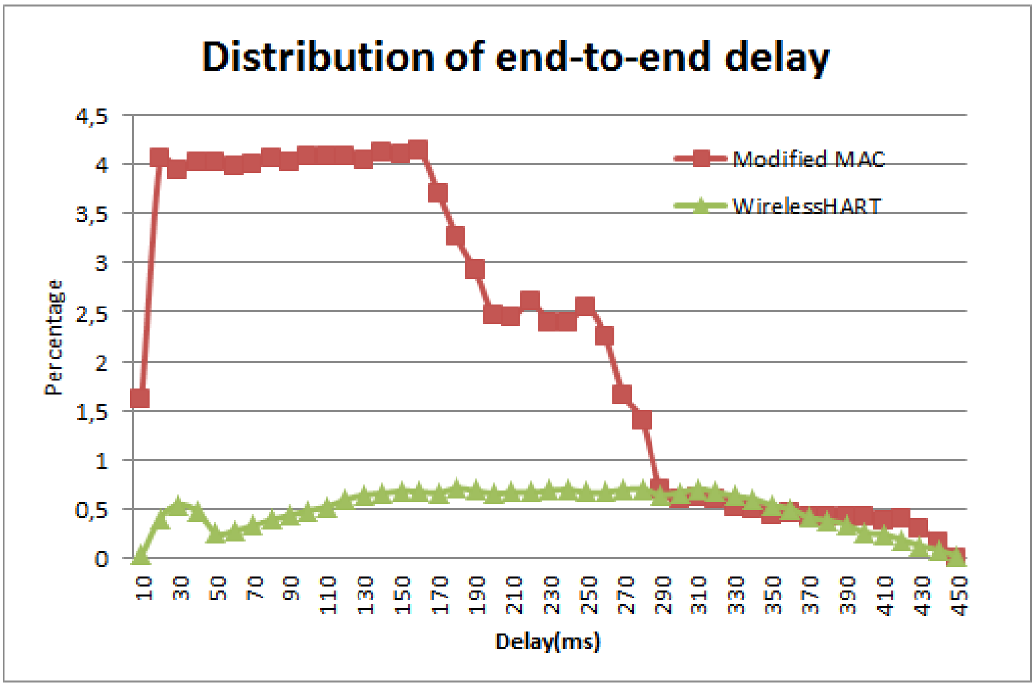


Figure 5.1 Scenario 1 & 3 – Distribution of end to end delay

Figure 5.2 shows the maximum, minimum and average value of end-to-end delay. From this figure we can see that the minimum delay is the same, which is the case that after the node receive a message it send the message immediately successfully. For the maximum delay, it is 439 ms in modified MAC and 451 ms in WirelessHART which is higher than that in our protocol. And as we can see from the figure, the average delay of modified MAC is 74 ms less than WirelessHART, which shows a better performance.

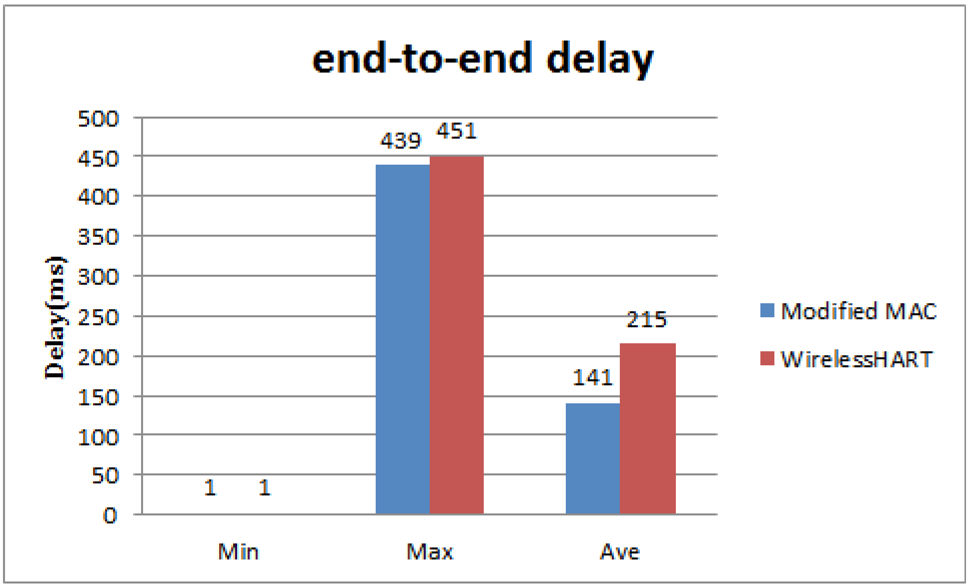


Figure 5.2 Scenario 1 & 3 – Max Min Ave delay

Figure 5.3 shows that the critical packet delivery rate is 99.54% for modified MAC which is quite good but only 22.86% for WirelessHART. The main reason of low delivery rate for WirelessHART is that slotted aloha method which lead to high collision is used to transmit the critical message. And the all packet including ordinary packet delivery rate is 86.16% for modified MAC, this is relatively lower than critical packet delivery rate, because some ordinary messages are covered by the critical message. And for WirelessHART, the all packet delivery rate is 84.68%, which is similar to modified MAC.

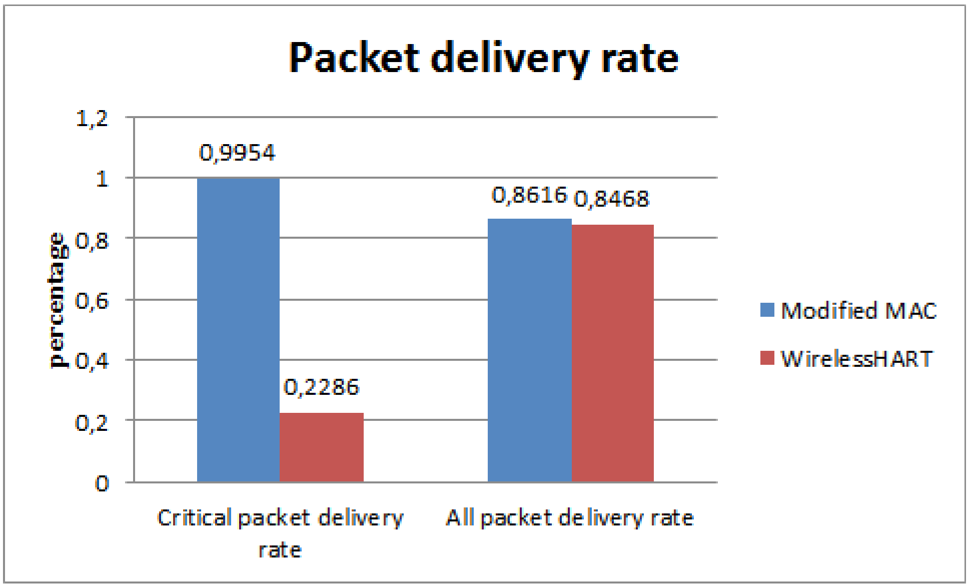


Figure 5.3 Scenario 1 & 3 – Packet delivery rate

5.2 Scenario# 2 & Scenario# 4

This two scenarios are both only 9 nodes sending emergency packet to the gateway. From figure 5.4, we can see that there are more packets received between 20 ms and 170 ms for modified MAC. And the percentage of each delay period is also increased for WirelessHART comparing to the scenario 2. But the difference between modified MAC and WirelessHART does not change much.

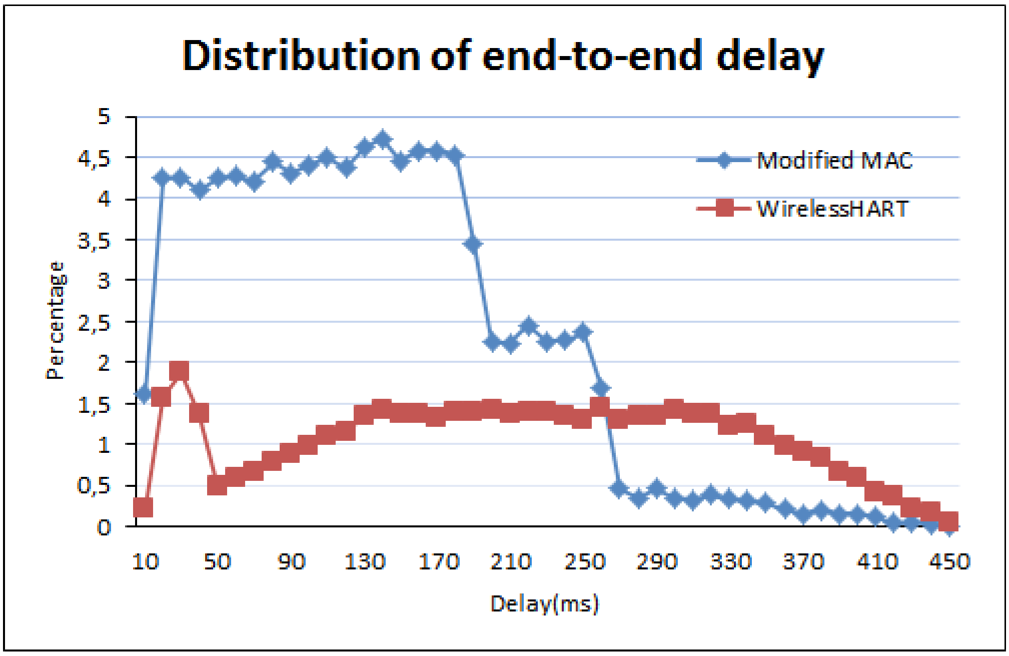


Figure 5.4 Scenario 2 & 4 – Distribution of end to end delay

As we can see from Figure 5.5, the minimum, maximum and average delay are 1ms, 439 ms and 128 ms for modified MAC, while for WirelessHART they are 1 ms, 448 ms and 206ms. According the result, we can say that the modified MAC has a better performance than WirelessHART in the simulation environment when 9 nodes sending emergency packet.

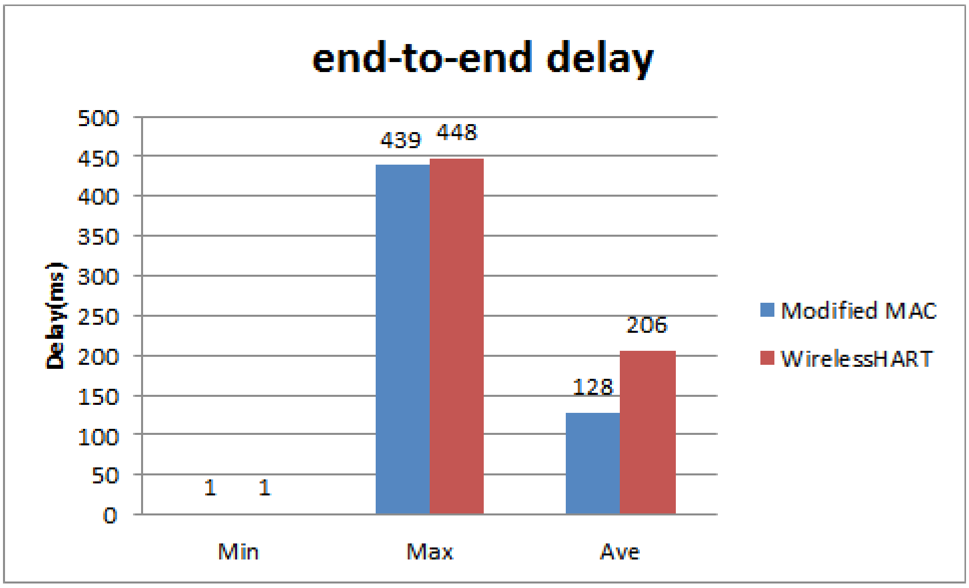


Figure 5.5 Scenario 2 & 4 – Max Min Ave delay

From Figure 5.6, we can see that both of the percentage are increased in both condition comparing with all nodes sending emergency packet. But for modified MAC the critical packet delivery rate has no big difference comparing to all nodes sending emergency packet, and this shows that modified MAC can perform well for critical message even when the number of transmitting emergency packet changed. For WirelessHART, both percentage are increased because the number of sending emergency packet decreased so that there are less collision in the random backoff period since slotted aloha method is used for transmission. Although both percentage are increased for WirelessHART, they are still less than the case in modified MAC.

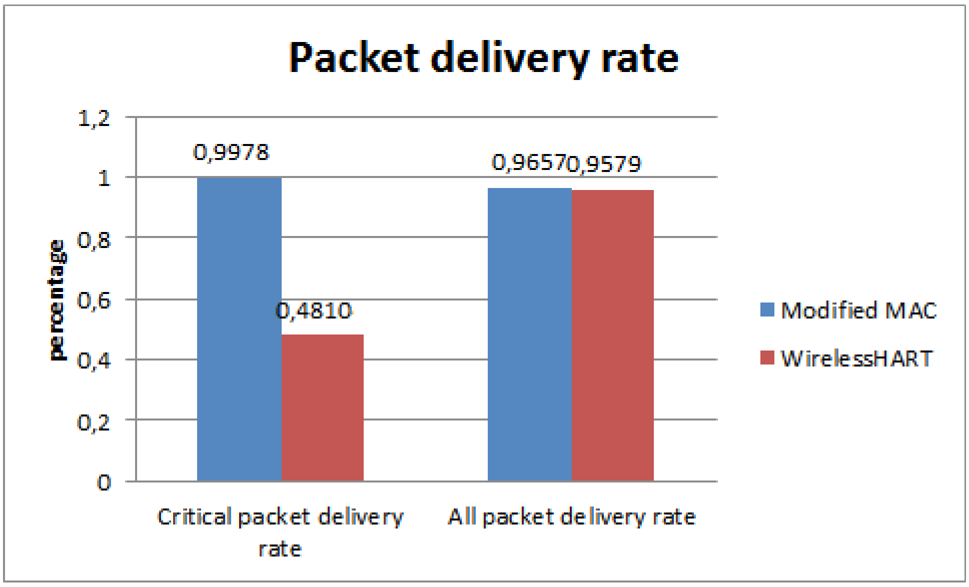


Figure 5.6 Scenario 2 & 4 – Packet delivery rate

6 Conclusions

This paper proposes a modified MAC protocol for emergency message in IWSAN. From the result and discussions we can see that the modified MAC can actually achieve a better performance from the point of both average delay and packet delivery rate for critical message in IWSAN than WirelessHART. And the main contributions of this thesis are:

(1) Implement a modified MAC for critical message in industrial automation control in TinyOS.

(2) Provide a new idea of setting the position of slot in superframe to adapt to different kinds of message which need specific requirements.

(3) A performance comparison between modified MAC and WirelessHART was provided.

Future work will add channel hopping in the simulation, because TinyOS do not provide related interface for generic MAC protocol design (but provide for specific chips), and thus have no interface for implementation of channel hopping. And this paper focus on comparing the performance of the algorithm with WirelessHART on the same condition, and for simplify the simulation, we do not add the multichannel part. After adding that, it may gives a better performance than the present result.

References

[1] Andreas Willig,"Recent and Emerging Topics in Wireless Industrial Communications: A Selection",IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, VOL. 4, NO. 2, MAY 2008.

[2] Gungor, V.C. Hancke, G.P."Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Approaches", IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, VOL. 56, NO. 10, OCT 2009.

[3] Wikipedia,"Zigbee",
http://en.wikipedia.org/wiki/ZigBee

[4] Suriyachai, P. Roedig, U.;Scott, A."A Survey of MAC Protocols for Mission-Critical Applications in Wireless Sensor Networks", IEEE COMMUNICATIONS SUVERYS TUTORIALS, VOL. 14, NO. 2, 2012.

[5] Broadband Wireless Networking Lab, "Wireless Multimedia Sensor Networks",
http://www.ece.gatech.edu/research/labs/bwn/WMSN/testbed.html

[6] Wikipedia,"IEEE 802.15.4",
http://en.wikipedia.org/wiki/IEEE_802.15.4

[7] Zlatko Bundalo,Miroslav K. "TECHNICAL CHARACTERISTICS OF WIRELESSHART NETWORK", 13th International Research/Expert Conference, Hammamet, Tunisia, 16-21 October 2009

[8] http://en.wikipedia.org/wiki/IEEE_802.15.4#/media/File:IEEE_802.15.4_protocol_stack.svg

[9] Holger Karl, Andreas Willig,"Protocols and Architectures for Wireless Sensor Networks", pp. 140, 2007

[10] Holger Karl, Andreas Willig,"Protocols and Architectures for Wireless Sensor Networks", pp. 141, 2007

[11] Wikipedia,"ISA 100.11a",
http://en.wikipedia.org/wiki/ISA100.11a

[12] Wikipedia,"WirelessHART", http://en.wikipedia.org/wiki/WirelessHART

[13] http://en.hartcomm.org/hcp/tech/wihart/wireless_how_it_works.html

[14] LIANG Wei, "An example of WIA-PA network", http://english.sia.cas.cn/rh/rp/201404/t20140422_119815.html

[15] http://www.tinyos.net/

[16] http://tinyos.stanford.edu/tinyoswiki/index.php/TinyOS_Overview

[17] Wei Shen, "TinyOS Programming PartI"

[18] Muhammad Omer Farooq and Thomas Kunz, "Operating Systems for Wireless Sensor Networks: A Survey", Sensors 2011, 11, 5900-5930;

[19] Chewoo Na, Yaling Yang, Amitabh Mishra, "An optimal GTS scheduling algorithm for time-sensitive transactions in IEEE 802.15.4 networks", Computer Networks 52 (2008) 2543–2557

[20] Zdenek Hanzálek ˇ , and Petr Jurcík , “Energy Efficient Scheduling for Cluster-Tree Wireless Sensor Networks With Time-Bounded Data Flows: Application to IEEE 802.15.4/ZigBee ”, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, VOL. 6, NO. 3, AUGUST 2010

[21] Seong-eun Yoo, Poh Kit Chong, Daeyoung Kim, Yoonmee Doh, Minh-Long Pham, Eunchang Choi, and Jaedoo Huh, “Guaranteeing Real-Time Services for Industrial Wireless Sensor Networks With IEEE 802.15.4 ”, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 57, NO. 11, NOVEMBER 2010