

Delft University of Technology  
Master's Thesis in Computer Science

# A Comprehensive Study of Passive Wake-up Radio in Wireless Sensor Networks

Liang Huo





# A Comprehensive Study of Passive Wake-up Radio in Wireless Sensor Networks

Master's Thesis in Electrical Engineering

Embedded Software Section  
Faculty of Electrical Engineering, Mathematics and Computer Science  
Delft University of Technology  
Mekelweg 4, 2628 CD Delft, The Netherlands

Liang Huo  
L.Huo@student.tudelft.nl

10/07/2014

**Author**

Liang Huo (L.Huo@student.tudelft.nl)

**Title**

A Comprehensive Study of Passive Wake-up Radio in Wireless Sensor Networks

**MSc presentation**

17/07/2014

**Graduation Committee**

prof. dr. Koen Langendoen(chair)	Delft University of Technology
dr. Przemyslaw Pawelczak	Delft University of Technology
dr. Hong Li	NXP Semiconductors
dr. Wouter Serdijn	Delft University of Technology

## **Abstract**

Wake-up radio, as a secondary radio transceiver, is implemented to monitor the channel condition so that the main radio can be turned off when there is no communication activity. This thesis work focuses on the analysis of passive wake-up radio (PWUR), where the wake-up radio is entirely powered by the wake-up signal and does not need any additional battery supply. The work covers PWUR's both hardware implementation and performance analysis. In particular, we analyze some typical issues when PUWR is adopted into the ZigBee network. Our results suggest how PWUR can improve ZigBee network's performance in terms of latency, energy consumption and reliability.



# Preface

This master thesis is a cooperation project between Embedded Software group of TU Delft and NXP Semiconductors. In passive wake-up radio, wireless power is transferred in the format wake-up signals from the wake-up transmitter to the wake-up receiver. Embedded Software group in TU Delft has been working on the topic of wireless power transfer very successfully in recent years. Meanwhile, NXP Semiconductors has their focus on analyzing the overall ZigBee network performance and exploring new technologies and methods to improve network reliability. This thesis topic is formulated in order to combine both sides' best interests.

I would like to thank my supervisors Dr. Przemyslaw Pawelczak at TU Delft and Dr. Hong Li at NXP Semiconductors. It is their supports that guide me through this thesis work over the past 9 months. I came to the Netherlands two years ago and never expect to have such an amazing journey. Thanks for all my friends in Holland which makes this country my "home away from home". Thanks my mom. She is always there when I need her. It is her unconditional love and support make me who I am today.

Liang Huo

Delft, The Netherlands  
10th July 2014



# Contents

<b>Preface</b>	<b>v</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Overview of WSN . . . . .	1
1.2 WSN Supporting standards . . . . .	2
1.3 Wake-up radio . . . . .	3
1.4 Problem statement . . . . .	5
1.5 Structure of the report . . . . .	7
<b>2 Background</b>	<b>9</b>
2.1 Hardware design . . . . .	9
2.2 MAC layer performance . . . . .	11
<b>3 Theoretical Model</b>	<b>13</b>
3.1 Wake-up process . . . . .	13
3.2 Analytical model for duty cycling and PWUR . . . . .	15
3.2.1 Duty Cycling . . . . .	15
3.2.2 PWUR . . . . .	21
3.3 Multiple access control . . . . .	23
3.3.1 TDMA . . . . .	23
3.3.2 CSMA/CA . . . . .	24
3.3.3 Combination . . . . .	25
<b>4 Simulations</b>	<b>29</b>
4.1 Simulation setup . . . . .	29
4.1.1 ZigBee model in OPNET . . . . .	29
4.1.2 Wake-up radio implementation . . . . .	32
4.1.3 Simulation assumptions . . . . .	34
4.2 Results and analysis . . . . .	34
4.2.1 Duty cycling VS PWUR . . . . .	34
4.2.2 PWUR on CSMA/CA . . . . .	40

<b>5</b>	<b>Conclusions and Future Work</b>	<b>51</b>
5.1	Conclusions . . . . .	51
5.2	Future Work . . . . .	51

# Chapter 1

## Introduction

### 1.1 Overview of WSN

The emerging field of wireless sensor network (WSN) has attracted significant research interests in the recent years [10][11]. It is considered to be the most essential component of the Internet of Things (IoT) [25] since it supports numerous applications, such as environment and habitat monitoring, health monitoring, machine surveillance, and traffic control [12].

A wireless sensor network is built from a few to hundreds or even thousands of nodes. Each node contains at least one sensor to monitor physical or environmental conditions, such as temperature, sound, pressure, etc. and to cooperatively pass their data through the network to a main location. Apart from sensor, each node has typically several parts: a radio transceiver with an internal antenna or connection to an external antenna, a microcontroller (MCU), an electronic circuit for interfacing with the sensors and an energy source, usually a battery or an on-board form of energy harvesting. Multiple nodes make up a sensor networks following a specific network topology such as star network or an advanced multi-hop wireless mesh network.[15][31]

Among all the issues that exist in WSN, there are three of them standing out the most, namely the energy consumption performance of the sensor node, latency performance for data transmission, and the overall reliability performance of the system.

#### *Energy*

First and foremost is the energy consumption issue. The sensor nodes are usually powered by batteries thus have very limited lifetime if no power management is performed. Plus, replacing sensors or batteries in many applications is not cost-efficient or sometimes even impossible. It is crucial to minimize energy consumption of wireless sensor network and extend its lifespan.

### ***Latency***

Latency is a measure of time delay which is defined as the time it takes from when a data packet is transmitted from the originating sensor to reaching its final destination. It is usually application-dependent. For example, messages to actuators should be communicated in real-time in our industrial automation system. Applications such as room temperature to detect a fire alarm often has a higher requirement than other more delay-tolerant systems. In WSN, information is exchanged between nodes in the format of data packet. Thus packet delay is another important issue.

### ***Reliability***

Last but not least, since wireless sensor network is often deployed in a very dynamic environment. Lots of factors can lead to performance variance. Collisions and data retransmissions may happen too often due to large number of nodes in the network. Parameters like received power or Signal Noise Ratio (SNR) vary from time to time due to the nature of wireless channel. To design a robust WSN system, we need to provide a certain level of reliability to handle all the uncertainties.

## **1.2 WSN Supporting standards**

### ***ZigBee***

There are quite a few competing WSN standards on the market. ZigBee[1], Bluetooth[2], WirelessHART[3], ISA100[4] are some of the most popular ones. Among them ZigBee is a specification for the higher protocol layers upon the physical (PHY) and medium-access control (MAC) layers defined in the IEEE 802.15.4 specification.

ZigBee operates in the industrial, scientific and medical (ISM) radio bands: 868 MHz in Europe, 915 MHz in the USA and 2.4 GHz in most jurisdictions worldwide. Data transmission rates vary from 20 kilobits/second in the 868 MHz frequency band to 250 kilobits/second in the 2.4 GHz frequency band. There are two versions of the CSMA/CA mechanisms at the MAC layer; slotted (beacon-enabled) and unslotted CSMA (non-beacon enabled). The ZigBee protocol is based on the ad-hoc on-demand distance vector (AODV) algorithm. This means, routing, discovery, and peer-to-peer communication is possible through this routing protocol. Mesh networking topologies are supported. All nodes share the same channel and frequency hopping is not available. There are two classes of network devices in ZigBee standards such as Full-Function Devices (FFD) and ReducedFunction Devices (RFD). FFD can form networks of any desired type such as mesh, star and hybrid whereas; RFD can only connect to a full function node [27].

## ***RFID***

RFID (radio frequency identification) is a means of storing and retrieving data through electromagnetic transmission using a radio frequency (RF)-compatible integrated circuit. Today, it is applied widely in supply-chain tracking, retail stock management, parking access control, library book tracking, marathon races, airline luggage tracking, electronic security keys, toll collection, theft prevention, and healthcare [23].

Efforts have been put into merging the RFID technology into WSN [23] [33]. The main idea is to combine the RFID's property (energy harvesting, identifying and positioning) into WSN's (sensing, identifying, positioning, and multihop communications). Another reason is that the application of RFID systems is much wider than that of wireless sensors. And RFID tags are much more economical than sensor nodes.

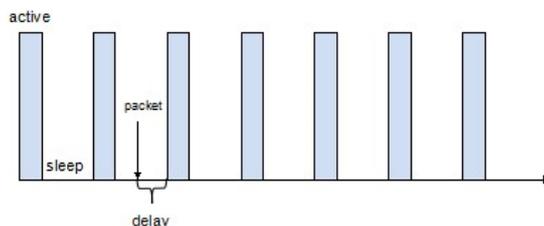
In this work, we choose ZigBee standard since it is the most widely used WSN standard and has been industrialized into numerous mature applications. Compared to its main rival Bluetooth, it provides significantly better mesh networking capability. In the following chapters, we will introduce how we integrate RFID technology into our ZigBee system to realize our passive wake-up radio implementation.

### **1.3 Wake-up radio**

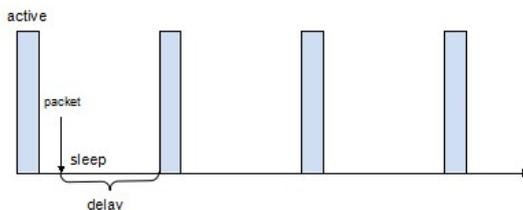
Mentioned in earlier section, energy and latency are two crucial factors in the design of WSNs. When a node is active and waiting to receive data, it wastes energy on idle listening. Since traffic loads are usually low in WSNs, such idle listening can waste enormous amounts of energy unless efficient communication mechanisms are employed.

To extend the lifetime of a sensor node, its radio component is turned off, its microcontroller (MCU) is set into a sleep mode and a timer is used to turn the node active periodically. This scheduled approach is called duty cycling. Duty cycling has been widely used in WSN. However, it suffers an energy-latency trade-off issue (Figure 1.1).

Figure 1.1 describes the process of a sensor node receiving a packet. The node tries to manage the trade-off between energy consumption and data latency, by setting the duty cycle accordingly. With a higher duty cycle like Figure 1.1a, the packet delay is shortened because even if packet was generated when the receive node is asleep, it does not take too long until the next active period. However, in this case, more energy is consumed since the node is turned to active more frequently which leads to more idle-listenings. On the other hand, with a lower duty cycle like Figure 1.1b, the node will consume less energy at the cost of higher latency for data delivery since sleeping period in each cycle is now longer. There is no way



(a) shorter delay, higher energy consumption



(b) longer delay, lower energy consumption

Figure 1.1: Energy-latency trade-off.

to improve the energy and latency performance at the same time by the duty cycling approach. The best we can do is to try to synchronize the whole system so that receiver is exactly at its active period when the packet is generated at the transmitter side. But there are two major disadvantages of synchronization.

First, usually system events are not so periodical or predictable. This makes the synchronization process more difficult. Second, complexities and overhead to the MAC protocol introduced by synchronization process explodes exponentially in a network with a large number of nodes.

Wake-up radio provides a solution to break the energy-latency tradeoff. It is an on-demand approach where the node is woken up by a radio signal, namely wake-up signal (WuS). A WuS triggers a node to wake up from the sleep mode and start reception/transmission activities. The wake-up signal is sent or received by a secondary radio transceiver, which consumes extremely low power. The energy benefit of using radio wake-up in comparison with duty-cycling is that nodes do not waste energy on idle listening of the main radio, since they are only awakened when there is a request for communication. The latency benefit is that since wake-up radio is a purely on-demand approach, the only delay is the one introduced by the wake-up process itself.

Figure 1.2 summarized multiple ways to categorize different wake-up receiver schemes [16]. In our work we follow the classification of what is the power source of the Wake-up Receiver (WuRx).

Based on their energy sources, wake-up receiver proposals can be classified

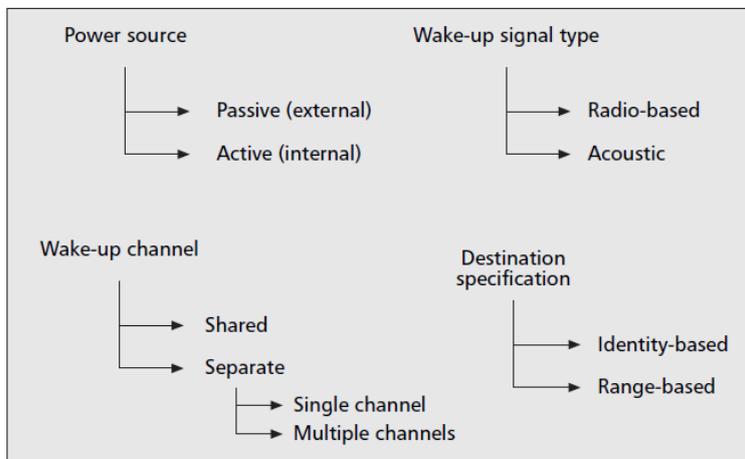


Figure 1.2: Classification of wake-up receiver schemes [16].

as passive wake-up receivers in which the wake-up circuitry is triggered by an external energy source and as active wake-up receivers in which the internal battery is used such that the wake-up receiver monitors for the possible wake-up signal. In our work we are more interested in the performance of Passive Wake-up Radio (PWUR) which harvests energy from the WuS transmitted by the sender to power themselves. To be more specific, we achieve the wake-up radio using a passive RFID tag as the wake-up signal receiver and an RFID reader as the wake-up radio transmitter. More details on the hardware implementation will be discussed later.

## 1.4 Problem statement

So far we have discussed WSN (ZigBee in particular) and PWUR. Our main focus is to investigate,

**How can PWUR improve the overall performance, including energy, latency, and reliability, in a ZigBee network?**

To answer this research question, we first propose a PWUR-enabled ZigBee network architecture in Figure 1.3. To our knowledge, this is the first-ever effort to combine PWUR with ZigBee network specifically.

Figure 1.3 shows a cluster-tree topology ZigBee network. Due to the power cost and size of a RFID reader, it is not realistic to equip an RFID reader onto every ZigBee end-device. Thus, ZigBee end-device cannot function as a wake-up transmitter. As an alternative, we equip the RFID reader only on the ZigBee routers. In a start network, it would be on the ZigBee

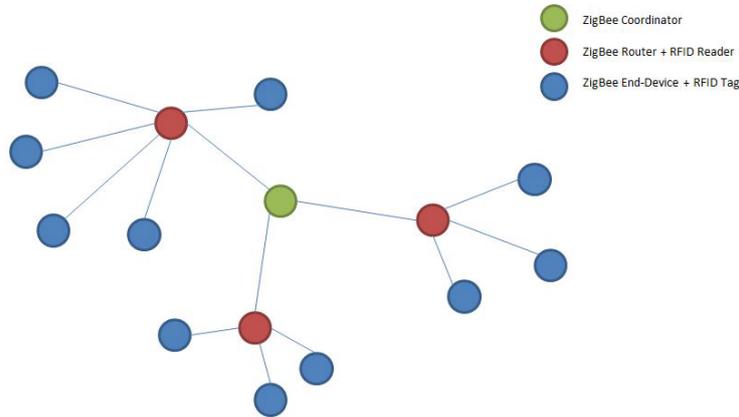


Figure 1.3: PWUR-enabled ZigBee network.

coordinator. A ZigBee end-device is equipped with an RFID tag to harvest energy from WuS as a WuRx. Noticing that in this network architecture, end-devices do not have the capability of waking up each other. Any communication between two end-devices has to go through routers/coordinator.

Having our network architecture defined, we focus on two specific problems to answer the research question.

*Problem (1): The router/coordinator needs to send a data packet to an end-device. It first sends a WuS to wake up the end-device. After being woken up, the end-device knows that there is a packet for him and start to receive the packet from coordinator. What would the energy and latency performance be compared to the traditional duty cycling approach?*

*Problem (2) The router/coordinator needs to receive a data packet from an end-device. It sends a WuS to wake up the end-device. After being woken up, the end-device knows that the coordinator asks for a packet and start to transmit the packet to coordinator. What would the overall MAC layer performance be when such wake-up mechanism is introduced?*

These two approaches cover two sorts of WSN applications. Scenario (1) is that the coordinator is going to distribute information to the end-device. For instance, a central control in a smart home network informs the sensor on the air-conditioner what is the correct temperature to be set at the moment. Scenario (2) is that the coordinator is going to collect information from the end-device. Typical use cases are the health monitoring, coordinator receives all the data from sensors on different parts of human body.

## 1.5 Structure of the report

The structure of the report is as follows, Chapter 2 gives a background introduction of PWUR study, including how we refer to the previous works and improve them. Chapter 3 introduces the theoretical analysis on PWUR compared to duty cycling and PWUR's influence on CSMA/CA protocol. In Chapter 4, simulations of the two scenarios mentioned above, including the simulation setup, simulation results and analysis are given. Finally, in Chapter 5, we conclude our work and give a suggestion of the future on this topic.



## Chapter 2

# Background

The related work of passive wake-up radio can be categorized into two groups, PWUR hardware design and its MAC layer performance analysis. In this chapter, we will introduce the background including how we refer to and improve them in our work.

### 2.1 Hardware design

While there has been a lot of research on active wake-up radio [21][26][17], there has not been many on PWUR. Based on the wake-up capability, PWUR can be categorized into range-based PUWR and ID-based PWUR. Using a range-based wake-up radio, all sensors within range of the transmitted wakeup signal will be woken up, while an ID-based wake-up radio transmits a wake-up signal that contains the intended destination's address and thus only wakes up the node with a matching address. ID-based wake-up provides the identifying capability, therefore its hardware is more sophisticated. A demodulation component should be added in order to obtain the WuS's destination address.

#### *Range-based*

One of the earliest and most cited work [18] laid the groundwork for PWUR study. The circuit of its passive WuRx is shown in Figure 2.1. Although some more recent works have more sophisticated circuit design, the basic concept is still the same.

In Figure 2.1, the antenna reacts to the EM wave of the WuS and generates an input voltage. The transformer increases the peak voltage of the sinusoidal signal curve to a certain level. Then a diode is used as a rectifier to produce the output voltage. It uses a capacitor to store energy.

When the antenna receives the radio signal, the flow of current accumulates energy on Cse and the voltage across the capacitor increases accordingly. This process is described by equation Equation 2.12.2.

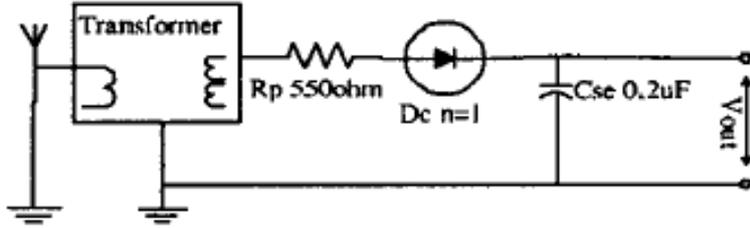


Figure 2.1: Passive WuRx circuit [23].

$$E = P_r \times t \times ef \quad (2.1)$$

$$V_{out} = \sqrt{\frac{2 \times E}{C_{se}}} \quad (2.2)$$

where  $E$  is the total amount of RF energy the circuit collected.  $P_r$  is the receive power of WuS.  $t$  is the time required to accumulate energy.  $ef$  is the efficiency of energy harvesting.  $C_{se}$  is the capacitance of the capacitor  $C_{se}$ , which is responsible of storing the energy and generate the output voltage. When sufficient energy has been stored on  $C_{se}$ ,  $V_{out}$  becomes higher than the threshold of interrupt voltage of sensor's MCU. A successful wake-up is completed. [18] claims that its PWUR circuit could operate at the range of 10 ft with 5 ms latency. If a comparator and an amplifier are added, which respectively consume negligible currents of 350 nA and 880 nA, the radio could theoretically reach up to 100 ft with 55 ms latency. However, there is no hardware implementation in [18].

Later [14] provides their PWUR circuit design. According to [14], they achieve a sensitivity of -29.3 dBm to obtain an output voltage of more than 0.7 V at the frequency of 868 MHz. Figure 2.2 is their wake-up circuit diagram. Unlike [18], their CMOS IC is fabricated. Our later analysis is based on this hardware implementation.

### ***ID-based***

ID-based wake-up requires the identifying capability on top of basic range-based wake-up. RFID technology is a natural candidate to meet the need. In fact, WISP (Wireless Identification and Sensing Platform) [29], developed by Intel Research is exactly such a hardware designed to explore sensor-enhanced RFID applications (Figure 2.3).

WISP is initially designed to conduct research which combines RFID and WSN. However, because it has the demodulation module shown in Figure 2.4, it can be directly applied to realize ID-based PWUR with an external sensor attached to WISP. Related works are published in [12]. Work

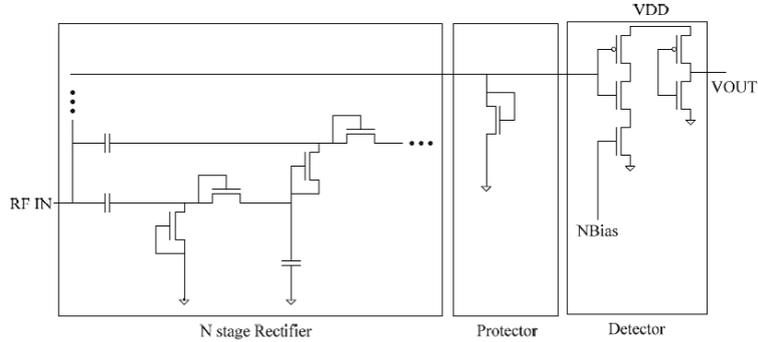


Figure 2.2: Block diagram of the wake-up circuit [14].



Figure 2.3: WISP 4.1 DL.

of [13] has extended WISPs wake-up range from 13 ft to 17 ft. This is, however, still noticeably shorter than the range-based wake up, since part of the harvested energy needs to be spent on the demodulation process. Detailed analysis of wake-up range calculation will be provided in Section 3.1.

## 2.2 MAC layer performance

Apart from hardware implementation, previous work also tried to answer how the PWUR can improve WSN's MAC layer performance, what new issues and requirements will be introduced PWUR is applied. [30] [22] [32] [20] proposed their PWUR MAC protocols and/or compare its performance with other existing protocols, while [24] [34] focused more on the analytical model of PWUR MAC. In terms of using RFID to realize PUWR, [30] proposed their so-called "PRFW" (passive RF wake up) scheme and conducted a very simple energy consumption analysis. [19] proposed a MAC protocol called RFIDImpulse and compared its energy efficiency and transmission rate with the BMAC protocol and the IEEE 802.15.4 standard.

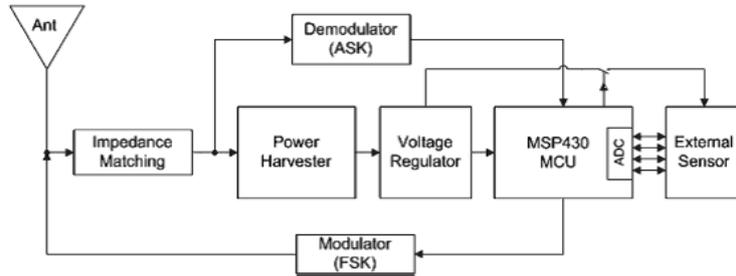


Figure 2.4: Block diagram of the WISP [29].

However, with all the previous work that have been done, there are still some limitations that have not been addressed before.

1) When designing MAC protocols, hardware feasibility is often ignored. Factors like sensitivity of circuit component, antenna gain, influence of wireless channel property, etc. should be considered into the MAC layer performance analysis. Although these are all PHY layer factors, MAC layer performance analysis should rely on valid PHY layer assumptions.

2) Most previous RFID-based PWUR analyses assume a commercial RFID reader and a passive RFID tag are attached to each sensor node. The analysis is based on an important assumption that all nodes have the capability to wake up their neighbors. This is unrealistic due to the considerate amount of energy required by RFID reader and its large size.

3) No previous work has conducted intensive PWUR energy and latency performance analysis in a ZigBee network before. ZigBee, using IEEE 802.15.4 standard to support its PHY and MAC, has its own characteristics. They should all be taken into consideration.

In our work, we have considered the limitations listed above. Improved upon the previous work, we believe that we provide more all-around, more intensive and more realistic results and analysis.

## Chapter 3

# Theoretical Model

In this chapter, we discuss the theoretical models which are related to our analysis. Firstly, we introduce the wake-up process of the PWUR-enabled sensor node that we proposed. Then in order to compare the energy and latency performance of duty cycling and PWUR approach, we describe our analytical models for these two approaches. Lastly, in order to make our model more applicable to the real-world multiple end-nodes scenario, two multiple access control methods that are widely used in WSN are introduced, namely TDMA and CSMA/CA. We also proposed a possible solution to combine TDMA and CSMA/CA in our analytical model.

### 3.1 Wake-up process

In a 2-channel wake-up radio enabled sensor node, wake-up radio and the main radio are deployed at different frequencies. Therefore, two sets of transceivers are needed. The block diagram below shows the work flow of the wake-up process.

As shown in Figure 3.1, wake-up radio is deployed on the 868 MHz band while main radio operates on the frequency of 2.4 GHz [9]. When a wake-up signal from Wake-up Transmitter (WuTx) is successfully received by the RFID WuRx, WuRx collects the RF energy from the wake-up signal until it is able to generate a voltage  $V_{out}$  that is higher than the threshold voltage to produce an interrupt signal to MCU. We have discussed the hardware implementation of this process in Chapter 2.

Once the MCU is interrupted by WuRx, it checks whether there is a need to transmit or receive data packet. If so, it turns on the main radio module for corresponding activities. The main radio module and the MCU exchange necessary data during transmission and/or reception.

It can be seen that during the whole wake-up process, the battery module is only needed to support the main radio module. The wake-up module operates in an entirely passive way. Without any power supply from the

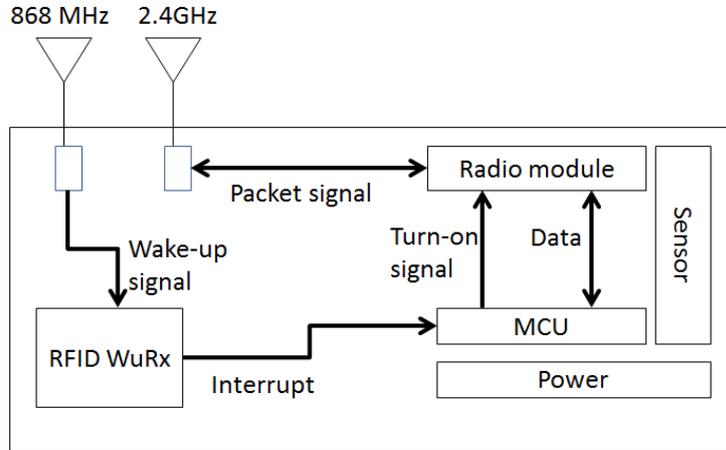


Figure 3.1: Block diagram of a WuRx.

sensor node itself, it solely relies on the RF energy that WuRx collected from the wake-up signal. This is similar to the working mechanism of passive RFID. RFID tags are powered by RF signal from the RFID reader. In fact, the RFID technology has been considered as one of the best solution to achieve PWUR due to its off-the-shelf availability.

However, as we have discussed in Section 2.1, while providing the advantage of purely passive working manner, PWUR suffers from a shorter working range compared to active wake-up radio. There are mainly 2 limitation factors:

1) Regulation constraints on the maximum transmission power allowed. According to UHF and EPC global standards [8], at the frequency range of 865 ~ 867.6 MHz, a maximum of 2 W ERP (3.8 W EIRP) of transmission power is allowed in European region (Table 3.1).

433 MHz.	Ultra High Frequency (UHF), ISM, backscatter coupling, active RFID	10 ~ 100 mW
865 ~ 868 MHz.	UHF RFID, Listen Before Talk, backscatter coupling	100 mW ERP Europe
865.6 ~ 867.6 MHz.	UHF RFID, Listen Before Talk, backscatter coupling	2 W ERP (3.8 W EIRP) Europe
865.6 ~ 868 MHz.	UHF Short Range Device (SRD), backscatter coupling	500 mW ERP Europe
902 ~ 928 MHz.	UHF SRD, backscatter coupling	4 W EIRP - spread spectrum, USA/Canada
2.400 ~ 2.483 GHz.	Super High Frequency (SHF) backscatter coupling	4 W - spread spectrum, USA/Canada
2.446 ~ 2.454 GHz.	SHF RFID and Automatic Vehicle Identification	0.5 W EIRP outdoor, 4 W EIRP indoor

Table 3.1: Regulation on maxim transmission power [8].

2) Hardware limitation which leads to the lower sensitivity of WuRx. Normally, the off-the-shelf main radio receiver e.g. a JN5148 node is between

-92 ~ -95 dBm [5]. However, for PWUR circuit in [14], the sensitivity can only reach the level of -29.3 dBm. If  $P_r < -29.3dBm$ , the WuRx is not able to harvest RF energy from wake up signal since  $P_r < P_{loss}$ .

Taken the 2 factors above into consideration, the expected operating distance for our range-based PWUR can be calculated with the following logarithm form of the Friis equation of pathloss, with a term for polarization loss included:

$$P_r = P_t - 20\log\left(\frac{4\pi d}{\lambda}\right) + G_t + G_r - L_p \quad (3.1)$$

In Equation 3.1, EIRP of the transmitter is  $P_t + G_t = 20\log 3.8W = 35.798 dBm$ . At the center frequency of 868 MHz,  $\lambda = 0.345m$ . The receive antenna gain is  $G_r = 0 dBi$ , and the polarization loss is  $L_p = 3 dB$ .  $L_p$  occurs because only half of the power transmitted from the circularly polarized transmit antenna is received by the linearly polarized receive dipole antenna. Using the operating threshold of -29.3 dBm from [14], Equation 3.1 predicts a maximum operational range of 35m. For ID-based wake up component (WISP), wake-up range is derived similarly [29]. But it goes without saying that ID-based wake-up has a even shorter wake-up range.

However, let us not forget this operational range is derived with the free space pathloss model. When a more realistic pathloss model is applied, the operational range of PWUR may also vary. We will discuss this more intensively in Chapter 4.

## 3.2 Analytical model for duty cycling and PWUR

One of the main reasons that wake-up radio is proposed is that the traditional duty cycling mechanism in WSN faces an energy-latency trade-off. In Section 1.3, we have explained how the energy-latency trade-off happens. In order to prove that PWUR is able to break this trade-off and provide a better performance on both ends, we need to first propose our analytical models for duty cycling and PWUR approach respectively.

### 3.2.1 Duty Cycling

In this part, we first introduce the analytical model of duty cycling, which is shown in Figure 3.2. The analytical model describes a communication process between the ZigBee coordinator and ZigBee end-device.

#### *Latency*

In Figure 3.2, x-axis describes the latency performance and y-axis represents the power consumption of each phase. The end-device undergoes the *s2a* (sleep to active) phase to shift from the sleep mode to active. Once it turns

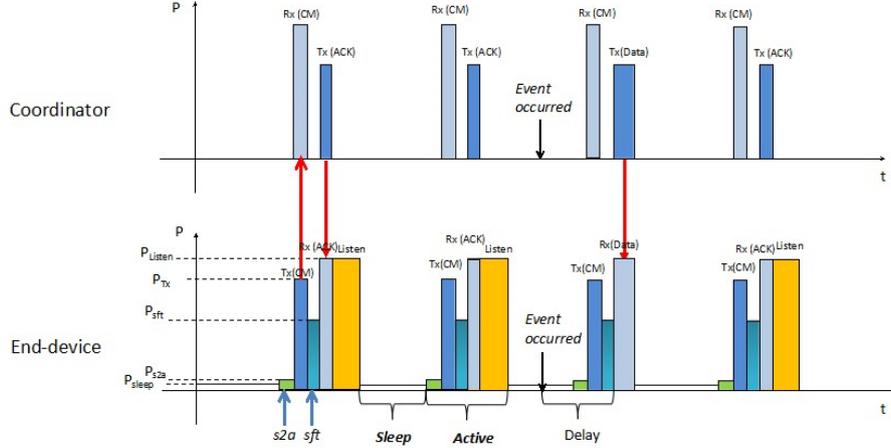


Figure 3.2: Analytical model of duty cycling.

active, it sends out a *CM* (Control Message) to its coordinator. The *CM* may contain 2 sorts of information: the updated real-time sensed information (temperature, humidity, etc) and the address of the sender end-device. *CM* is sent periodically so that the coordinator can be aware of both the condition and the presence of all the end-nodes in the network.

Once the *CM* is successfully received, coordinator sends an *ACK* (Acknowledgment) to end-device to inform the successful transmission. During this interval, end-device has gone through the *sft* (shift) phase from Tx mode to Rx mode in order to receive the *ACK*. In [9], the default setting of maximum waiting time for *ACK* is 50 ms. End-device remains in the listening mode for 50 ms for the *ACK* message. 50 ms is redundant enough to handle the multiple packets transmissions in one cycle introduced by multiple nodes scenario in Section 3.3 or higher packet rate. After the listening phase, the node goes back to the sleep mode.

One cycle length of the end-device is composed of active period  $T_{active}$  and sleeping period  $T_{sleep}$ . In the process we discussed above, the active period is the sum of  $s2a$ , *CM*, *sft* and listen phase. If the duty cycle is represented as  $\eta$ , we have Equation 3.2 3.3 3.4 below.

$$T = T_{active} + T_{sleep} \quad (3.2)$$

$$T_{active} = T_{s2a} + T_{CM} + T_{sft} + T_{listen} \quad (3.3)$$

$$T = T_{active}/\eta \quad (3.4)$$

We set the *CM* to be 32 bytes (256 bits). This is an example value. The length of *CM* should be long enough to carry both the sensor's real-

time condition and address information. But also the length should not be comparable to the length of data packet in order to lower the channel load. And from [6] we can also get the exact value for  $T_{s2a}$ ,  $T_{CM}$ ,  $T_{sft}$ . Therefore,  $T_{active}$  is considered to be a fixed value. The cycle length  $T$  only depends on the duty cycle value. For example, if the duty cycle is 10 %, from Equation 3.4 we can derive that  $T$  is 10 times of  $T_{active}$ . When  $\eta$  is 1%,  $T$  is 100 times of  $T_{active}$ .

The “event” in Figure 3.2 represents that coordinator has generated a packet for the end-device. However, it is obvious that the packet will not be received if the end-device is in the sleeping mode. As a result, the packet will only be sent if the  $CM$  from the destination end-device is received at the coordinator side. In Figure 3.2, an event occurred during the sleeping period of the second cycle. The coordinator has to wait until it receives the  $CM$  from the third cycle to start the transmission. The packet delay is thus the time between the real transmission and the event occurrence. In this case,  $ACK$  will not be transmitted anymore because the data packet has a higher priority. Also, if the end-device can receive the data packet successfully, then it also proves that its  $CM$  is received so that a separate  $ACK$  message becomes redundant.

As we investigate deeper into the model, we find out that an event could happened in 3 different time domains ( $T1$ ,  $T2$ ,  $T3$ ) which will results in different delay values as shown in Figure 3.3. Here we have

$$T1 = T_{s2a} + T_{CM} + T_{sft}$$

$$T2 = T_{listen}$$

$$T3 = T_{sleep}$$

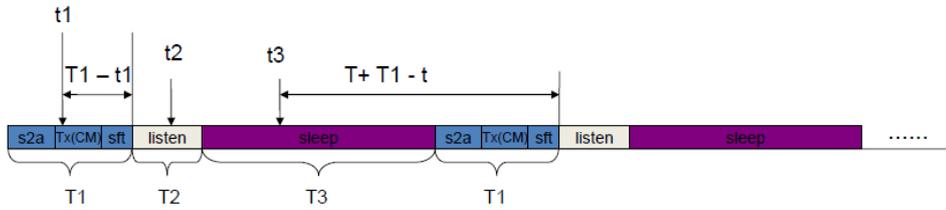


Figure 3.3: Packet delay calculation.

Suppose the event occurrence time is  $t$ . When  $t \in T1$ , end-device has to wait for the reception of  $CM$  to be completed. So  $delay = T1 - t$ . When  $t \in T2$ , the end-device is already listening, so transmission can start right away,  $delay = 0$ . When  $t \in T3$ , end-device is in sleep mode. The coordinator has to wait until the next  $CM$  from the same end-device, so here we have  $delay = T + T1 - t$ .

### Energy Consumption

Unlike latency performance which is only related to the packet arrival time. Energy consumption performance is also related to the packet arrival rate. To be more specific, the relation between packet interval  $T_I$  and cycle length  $T$  affects the energy consumption per packet. Let  $n = T_I / T$ .  $n > 1$  means that it takes more than one cycle to successfully receive a packet. For example,  $n = 100$ . Between the two consecutive packet arrivals, the end-device has gone through 100 cycles. On the other hand,  $n < 1$  means that more than one packet arrive within a single cycle. This relation is illustrated in Figure 3.4.

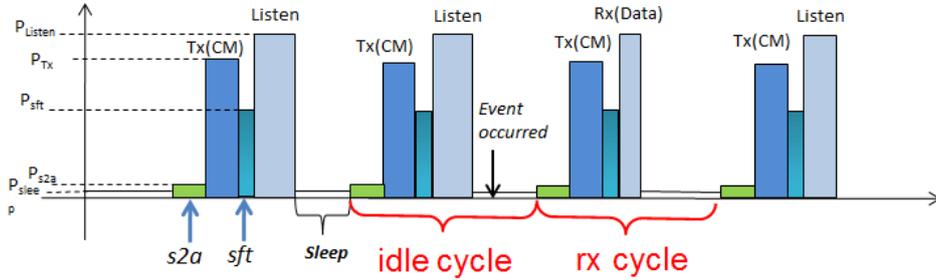


Figure 3.4: Idle cycle and rx\_cycle.

In Figure 3.4, event occurred during the sleeping period of the second cycle. The first two cycles do not receive any data packets from the coordinator. Idle listening happens in these two cycles, which is why we call them idle cycle. However, when the listening period starts at the third cycle, end-device receives a data packet from coordinator. We call this cycle an rx cycle. The time to receive a data packet  $T_{data}$  is different than  $T_{listen}$ , so energy consumption of an idle cycle and rx cycle is different as shown in Equation 3.5 and 3.6.

$$E_{idle\_cycle} = E_{s2a} + E_{CM} + E_{sft} + P_{rx} \times T_{listen} + P_{sleep} \times T_{sleep} \quad (3.5)$$

$$E_{rx\_cycle} = E_{s2a} + E_{CM} + E_{sft} + P_{rx} \times T_{data} + P_{sleep} \times T_{sleep} \quad (3.6)$$

When  $n > 1$ , it takes  $n$  cycles to receive a packet. Only one of them is rx\_cycle. So average energy consumption per packet can be calculated by  $E = (n-1) \times E_{idle\_cycle} + E_{rx\_cycle}$ . When  $n < 1$ , we have multiple events occurred at one cycle. The average number of packets received at each cycle is  $1/n$ . So we can make sure that no idle\_cycle exists because the end-device receives at least one packet during every listening period. Then the energy consumption per packet is again related to which time domain (T1, T2, and T3 in Figure 3.3) the first packet falls into the cycle. Figure 3.5) shows all

the 3 possible scenarios for that matter. We made an assumption that there will not be two consecutive packets arriving within T1 and T2, meaning that except for the first packet, all the following packets are all in T3. We will prove later that this is a valid assumption.

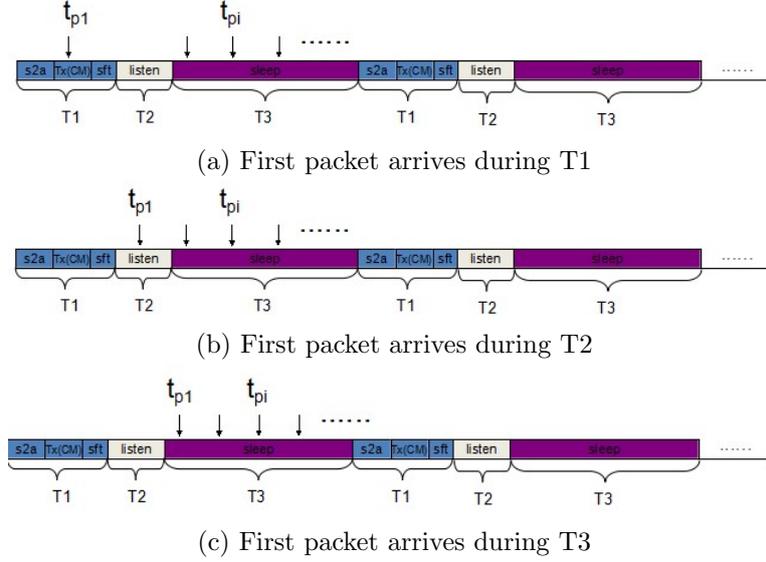


Figure 3.5: Energy consumption calculation.

In Figure 3.5, if  $t_{p1} \in T1$ , it will be received in T2 of the current cycle. If  $t_{p1} \in T2$ , node has to do keep listening in T2 before it catches the packet. If  $t_{p1} \in T3$ , it can only be received in T2 of the next packet. Different amount of energy are thus consumed according to the arrival time of the first packet in the current cycle.

### ***Packet arrival process***

For previous work analyzing duty cycling performance [12][30], packets arrives in a deterministic way. The packet interval  $T_I$  is a fixed value which is decided by the data packet rate  $\lambda$  (pkt/min). In our analysis, we believe this is not realistic enough. Therefore, instead of uniform distribution, in our analytical model packet arrives following the Poisson distribution. Let  $k$  equals to the number of packets that arrive in one minute, we have the probability mass function of events number below.

$$\Pr (X = k) = \frac{\lambda^k e^{-\lambda}}{k!} \quad (3.7)$$

The mean packet rate is  $\lambda$ . the mean value for packet interval time is thus  $1/\lambda$ . Probability density function  $f(x,\lambda)$  of the packet interval time therefore follows the exponential distribution as in Equation 3.8.

$$f(x; \lambda) = \lambda e^{-\lambda x} \quad x \in [0, \infty) \quad (3.8)$$

We choose 4 typical duty cycle values in our analysis, namely 0.1 %, 0.5%, 1%, 10%. But for each duty cycle value, the length of T1 and T2 is fixed.  $T_{active} = T_{s2a} + T_{CM} + T_{sft} = 0.0028$  s,  $T_2 = 0.05$  s. Suppose that a packet arrives at the beginning of T2. Here we choose the highest rate in our analysis,  $\lambda = 60$  pkt/min = 1 pkt/s. The possibility that interval between 2 successive packet arrivals  $T_I < 0.05$  is shown in Equation 3.9.

$$Pr(x < 0.05) = 1 - e^{-\lambda x} = 1 - e^{-0.05} = 0.049 \quad (3.9)$$

Equation 3.9 shows the probability for 2 successive packets in the same T2 is quite low even for the highest packet rate. We can imagine for a lower packet rate like  $\lambda = 3$  pkt/min = 0.05 pkt/s, the probability is even lower (Pr = 2.5e-3).

Thus it is safe to make our assumption when the packet arrives following the Poisson distribution.

### **Algorithm**

Now we have introduced the latency, energy and packet arrival process in our analytical model. We now list all the parameter values that we use in our model (Table 3.2 3.3)

Symbol	Explanation	Typical Value
$T_{s2a}$	startup delay from sleep mode to active mode	1.63 ms
$T_{CM}$	time consumed to transmit a CM packet (32 bytes)	1024 us
$T_{sft}$	settling time due to mode shift between transmission mode and receiving mode	130 us
$T_{listen}$	time for listening to incoming packets	50 ms
$T_{data}$	time to tx/rx a packet (1296 bits)	5.184 ms
$T_{active}$	active period of one cycle unit	52.784 ms
$\eta$	duty cycle of the sensor node	0.1%, 0.5%, 1%, 10%
$\lambda$	number of packets transmitted to per minutes (pkt/min)	low rate: 0.5, 1, 1.5, 2, 2.5, 3 high rate: 10, 20, 30, 40, 50, 60

Table 3.2: Duty cycling parameters table I.[6]

Parameters	Power
energy consumption in sleep mode	2.7 uw
average energy consumption during sleep mode to standby mode	855 uw
energy to transmit one CM packet @0dBm output power	33.9 mw
average energy consumption during transition between transmission mode and receiving mode	24.6 mw
energy consumption in receiving mode @1Mbps	35.4 mw

Table 3.3: Duty cycling parameters table II.[6]

Our algorithm of calculating average latency and energy consumption per packet of duty cycling is shown below Algorithm 1. It is implemented in MATLAB.

### 3.2.2 PWUR

The analytical model for passive wake up radio is much simpler compared to duty cycling. The reason is that the end-device can stay in the sleep mode unless it receives a wake up signal from the coordinator (WuTx). We no longer need to care about duty cycles, packet rates, etc. The PWUR analytical model is shown in Figure 3.6.

Figure 3.6 only shows the end-device (WuRx) side of the system, since its energy consumption requirements are more challenging compared to WuTx. When end-device receives a wake-up signal, it needs a certain time to accumulate enough RF energy in order to generate an output voltage that exceeds the threshold. The WU part describes this phase. And since the wake-up process is entirely passive, the power consumed is equal to  $P_{sleep}$ . We have discussed in Section 2.1 that the wake-up delay is related to wake-up signal strength, capacitance of the storing capacitor, and the threshold voltage.

After the node is woken up, it follows the same process as in the duty cycling model. Equation 3.10 and 3.11 describe the latency and energy performance of PWUR.

$$T_{delay} = T_{WU} + T_{s2a} + T_{CM} + T_{sft} \quad (3.10)$$

<pre> <b>Algorithm 3.1 latency</b> for i = 1:12 % 12 different packet rates   for j = 1:4 % 4 different duty cycles     Generate packet arrival time t<sub>p</sub>;     % exponential distribution     if t<sub>p</sub> &gt; T<sub>j</sub> (cycle length)       discard this packet;     end     N = number of packets in one cycle;     sum_latency = 0; % total latency     for n = 1:N       switch t<sub>p</sub> % packet delay         case 0 &lt; t<sub>p</sub> &lt;= T1           d = T1 - t<sub>p</sub>;         case T1 &lt; t<sub>p</sub> &lt;= T2           d = 0;         case T2 &lt; t<sub>p</sub> &lt;= T3           d = T + T1 - t<sub>p</sub>;       end       sum_latency = sum_latency + d;     end     latency = sum_latency / N;     % average latency per packet   end end </pre>	<pre> <b>Algorithm 3.2 energy consumption</b> for i = 1:12 % 12 different packet rates   for j = 1:4 % 4 different duty cycles     Generate packet arrival time t<sub>p</sub>;     Interval = 1/λ<sub>j</sub>; % average packet interval     n = Interval / T<sub>j</sub>;     if n &gt; 1 % more than 1 cycle between       packets         Energy = N<sub>rx</sub>* E<sub>rx</sub> + N<sub>idle</sub>* E<sub>idle</sub>;         % energy of idle cycles + rx cycles       else % N packets in one cycle         sum_energy = 0;         for n = 1:N           switch t<sub>1</sub> % arrival time of 1<sup>st</sup>             packet in the cycle               case 0 &lt; t<sub>p</sub> &lt;= T1 E = E1;               case T1 &lt; t<sub>p</sub> &lt;= T2 E = E2;               case T2 &lt; t<sub>p</sub> &lt;= T3 E = E3;             end             sum_energy = sum_energy + d;           end         end         energy = sum_energy / N;         % average energy consumption per packet       end     end   end end </pre>
---	--

**Algorithm 1:** Latency and energy consumption per packet calculation

$$E_{total} = P_{sleep} \times (T_{total} - T_{s2a} - T_{rx} - T_{CM} - T_{sft}) + E_{s2a} + E_{rx} + E_{CM} + E_{sft} \quad (3.11)$$

Equation 3.10 shows that latency is independent from packet rate. The only exception is when the following packet has already arrived before the wake-up process of current packet is done. However, this scenario requires an extremely high packet rate since wake-up process often takes very short time. Since  $T_{s2a}$ ,  $T_{CM}$ ,  $T_{sft}$  are all fixed values, the wake-up delay is the only variable here. It can be calculated from Equation 2.1 and 2.2.

In Equation 3.11,  $T_{total}$  is the total length of time that the node works.  $E_{rx}$  is the energy consumed to receive a data packet. We do not have to worry about the lower duty cycling and longer sleeping period will bring us longer latency. So Also, for the energy consumption, since the node can stay asleep unless an event occurred, it consumes  $P_{sleep}$  for most of the time. There are no idle cycles like in the duty cycle approach any more. Therefore, the on-demand wake-up effectively prevent the idle listening problem.

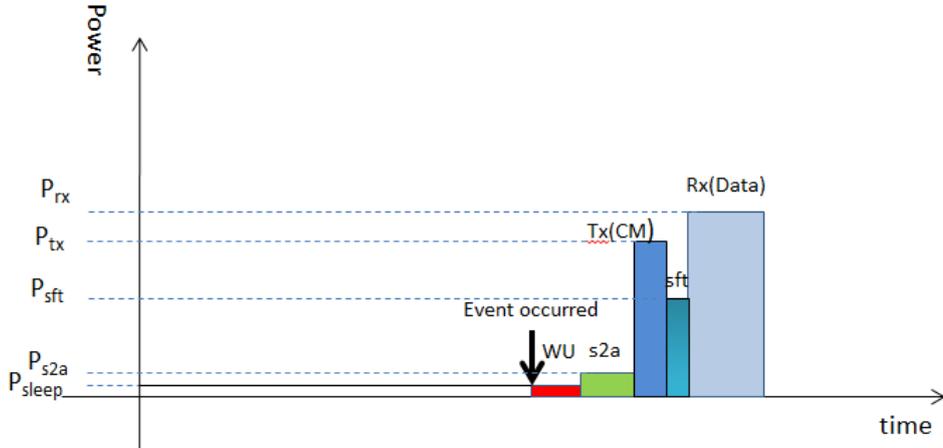


Figure 3.6: Analytical model of PWUR.

### 3.3 Multiple access control

So far, the models are not including the multi-node overhead. In WSN, medium access control is a big issue since there are often tens or hundreds of end-devices in one network competing for the medium at the same time. That is why in this section we add the multi-nodes scenario into our analytical model. MAC protocols are mostly based on TDMA and CSMA/CA. In this section we first introduce how they are used in our analysis. Then we propose an approach to combine these two schemes.

#### 3.3.1 TDMA

Given there are  $M$  nodes connected to the same central coordinator. As they all need to communicate with the coordinator, channel access control is needed in the system.

In duty cycling, TDMA scheme assigns one time slot of the whole cycle unit to each end-node to handle the multiple access control. The length of one time slot should be at least longer than necessary time for one successful data packet transmission. Otherwise, before one node finishing its transmission, the next slot for another node has already started. In our case, the slot length is the length of the active period within a cycle.

However, TDMA has its own limitation. Suppose we have duty cycling value  $n = 10\%$  for each end-device. The maximum number of end-nodes  $N$  the system can support without collision is  $1/10\% = 10$ . (Equation 3.12)

$$N = 1/n \quad (3.12)$$

It is obvious that when  $n = 10\%$ , 10% time of an end-node's cycle is active. So during the 90% of sleeping time, other nodes may turn into their

active periods. The time resource is effectively spreaded to all the end-nodes. The details are shown in Figure 3.7.

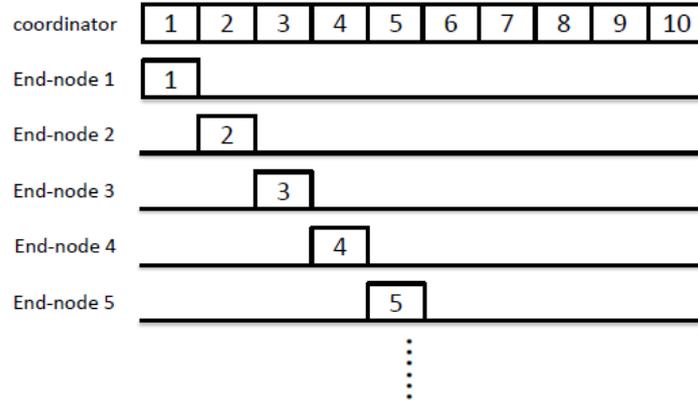


Figure 3.7: TDMA in duty cycling.

In Figure 3.7, following the TDMA scheme, coordinator sends data to end-node that is active at the time. When there are 10 end-nodes in a 10% duty cycling system, the coordinator reaches its full capacity. If there are more than 10 end-nodes, the coordinator cannot find another slots for the extra nodes.

From Table 3.2, we can derive that 10% duty cycling means end-node updates its information around every 0.5s. This is a very high rate in the real WSN. Most systems that follows a lower duty cycling, 1% for example (cycle length is around 5s, thus end-node updates around every 5s), is able to handle more end-nodes (100 nodes for 1% duty cycling) according to Equation 3.12.

So we come to the conclusion that if duty cycle is high and we have large number of end-nodes to the same coordinator, TDMA cannot handle the system needs anymore. We need to think of another way to increase the capacity.

### 3.3.2 CSMA/CA

There are few versions of CSMA/CA for different communication standards. In ZigBee, we use the unslotted CSMA/CA defined in [9]. But instead of scheduling each nodes transmission asynchronously like in TDMA, nodes can start the CSMA/CA at the same time and compete for the medium.

When the CSMA/CA starts, the node first does an initial random back-off with back-off length varying from 0 to  $2^{BE} - 1$  back-off units. Then it performs the clear channel assessment (CCA) to check if the channel is free at the moment. If so, then the node successfully obtains the medium and starts the transmission. If not, the node starts another back-off. Until

the number of back-offs exceeded the pre-defined limits, the node start the retransmission. When maximum number of retransmission is reached and the node still fails to obtain the medium and start the transmission, the node claims the failure of competing for the medium and discard the packet.

Compared to TDMA, performing CSMA/CA does not need the coordinator reserve a time slot for a specific end-device. Especially when the slot length can be much longer than the actual data transmission time, CSMA/CA provides a more dynamic, efficient way to make use of the channel. However, when the number of nodes increases, especially when too many nodes start CSMA/CA process simultaneously to compete for the medium, CSMA/CA's performance drops as collisions are not perfectly avoided as in TDMA.

In PWUR-enabled ZigBee network, the CSMA/CA process is invoked by the wake-up process. So wake up delay is added before the CSMA/CA initial back-off.

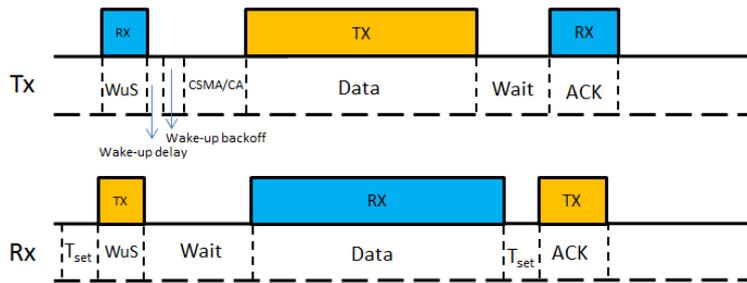


Figure 3.8: Communication process in PWUR-enabled WSN.

As shown in Figure 3.8, due to the difference of each node's wake-up delay length, their CSMA/CA starting time differs differently. In the later chapter we will see that this variance can sometimes improve the overall network performance significantly since the absolute simultaneous transmissions in large network lead to massive amount of collisions. By shifting the CSMA/CA starting time of multiple nodes differently, such collisions can be avoided to a great extent.

### 3.3.3 Combination

Overhead of TDMA is introduced by the design of time-slot length. Overhead of CSMA/CA is introduced by CSMA/CA back-off and collisions. We have pointed out that, in the duty cycling, TDMA is sometimes not enough to handle large number of end-nodes in the system, while CSMA's performance drops if too many nodes start channel contention simultaneously. We now combine TDMA with CSMA/CA to solve this issue. The basic idea is to first use TDMA to divide a large network into smaller clusters. Within

each cluster, we apply CSMA/CA to solve the multiple end-nodes problem.

For example, we have discussed that as Equation 3.12 shows, 10% duty cycling system can only handle 10 nodes because there are only 10 slots each cycle in total to be assigned. However, by applying our approach, each slot is no longer assigned to a node, but a cluster with multiple nodes. We now have  $1/10\% = 10$  clusters. When  $N = 30$ , number of nodes in one cluster  $N'$  is  $N/10 = 3$ . When  $N = 50$ ,  $N' = 5$ , etc. Nodes within the same cluster share the same timeslot. Coordinator only deals with the cluster which is active at the time. Within one cluster, nodes send out CM following CSMA/CA. Figure 3.9 shows the process.

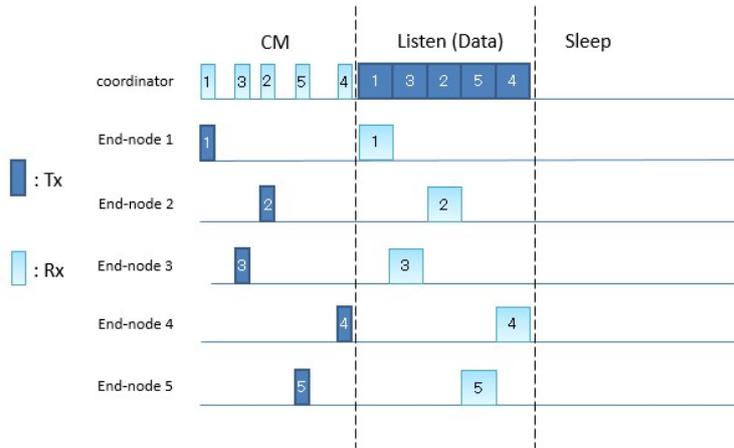


Figure 3.9: CSMA/CA and TDMA Combination.

End-node 1 ~ 5 share the same timeslot. In the first CM phase, they contend for sending out its CM to the coordinator. Coordinator receives the CM, knowing both the condition and presence of the sender node. The coordinator also checks whether it has a packet for the sender node. Then after receiving the Control Message from every node of the cluster, in the Listen phase, coordinator starts to send data packets (if any) to the CM sender node in the order of how it receives the CMs. After the data packets have been successfully sent to the end-node, this cluster of end-nodes goes to its sleeping phase. And coordinator also moves on to deal with the next cluster.

The approach in Figure 3.9 will bring several changes in our existing analytical model.

1)  $T_1$  in Figure 3.3 becomes longer due to CSMA/CA process of CM packets. Instead of just sending out one CM like in Figure 3.3, CM period continues until the last CM within a cluster has been successfully received by the coordinator. As the number of nodes in each cluster increases, the length of CM period also increases.

2) Buffering time in  $T_2$  needs to be considered if coordinator has packets

for multiple nodes within the same cluster. After CM period, coordinator checks if it has packets to for nodes in this cluster. If coordinator has packets for more than one node in the cluster, it sends them at the order of how CM is received. When the data packet is still in the buffer, its destination node needs to remain in the listening mode.

3) Longer CM period leads to higher energy consumption and latency value. A node has to spend more energy on monitoring the channel condition (to see if it is free) before sending CM packet.

4) Longer listening time may be needed according to the number of nodes within one cluster. If listening time is too short, before coordinator finishes sending out all the packets in the buffer, end-node has gone into sleep period.



## Chapter 4

# Simulations

In this chapter, we first introduce how our simulations are set up. Then the results and analysis on passive wake up radio performance are shown.

We want to implement the architecture that is shown in Figure 3.1 in a simulator. From our understanding, there is no previous work that has done this before. What we need in our design is a WuTx and a WuRx model within each contains the main radio module and the wake-up radio module. Wake-up radio module is able to control the transmission and/or reception of the main radio module based on the wake-up signal.

We implement our model in OPNET [7] since it already has a stable ZigBee model. This is a big advantage compared to some other popular simulators like ns-3. Also, its 3-tier (network model, node model, process model) hierarchical designing framework makes it easy and straightforward to implement our new architecture.

### 4.1 Simulation setup

#### 4.1.1 ZigBee model in OPNET

OPNET has its default ZigBee model. To build a ZigBee network in OPNET, We can first select the network topology to be either star, mesh or cluster-tree topology. Three kinds of ZigBee devices are defined to support different network topologies, namely ZigBee coordinator, ZigBee router and ZigBee end-device. Figure 4.1 shows an example cluster-tree ZigBee network which contains one coordinator, 5 routers and 28 end-devices. Coordinators control the formation and security of networks. Routers extend the range of networks. End devices perform specific sensing or control functions. In a smart home network, the coordinator may be a home theater control system with advanced support for lighting and security. Devices such as light fixtures, thermostats and air conditioners could be configured as routing devices. Simple devices such as light switches and security sensors could be end devices.

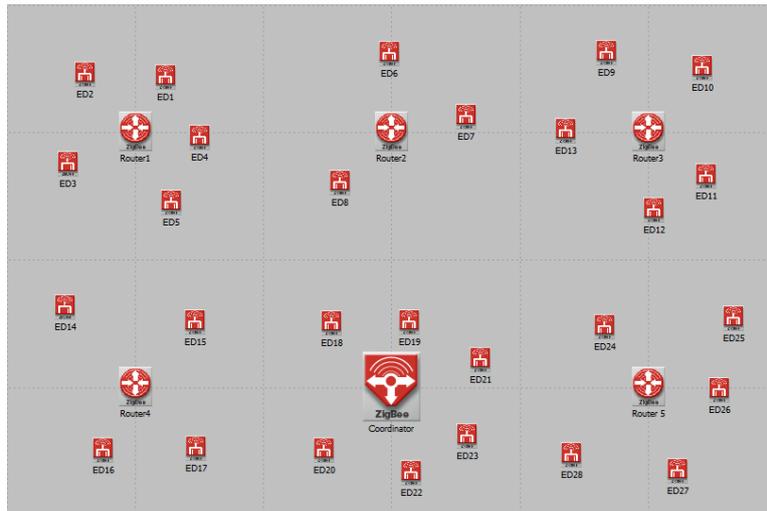


Figure 4.1: Cluster-tree ZigBee network.

After defining the network model, we need to define the node model as our next step. Figure 4.2 illustrates the ZigBee node model in OPNET.

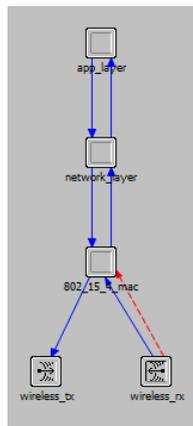


Figure 4.2: ZigBee node model.

As shown in Figure 4.2, each node is made up of different modules which are responsible for the tasks of a specific layer as in OSI model. Each module works based on its process model which determines the modules tasks and characteristics. We see the wireless transmitter and receiver which make the physical layer of the ZigBee node model. Three other modules represent the MAC, network, and application layers of the node. The physical layer contains the transmission and reception components which are responsible for data exchange. The MAC module provides the functional and procedural means to transfer data and to detect and correct errors occurred in the phys-

ical layer. The network layer is responsible for transferring variable length data, and for routing packets delivery including routing through intermediate routers. Finally, the application layer is responsible for identifying communication partners, determining resource availability, and synchronizing communication. In our work, we mostly focus on the MAC layer module. In fact, in the next section, you will see that we modified the default IEEE 802.15.4 MAC process model shown in Figure 4.3.

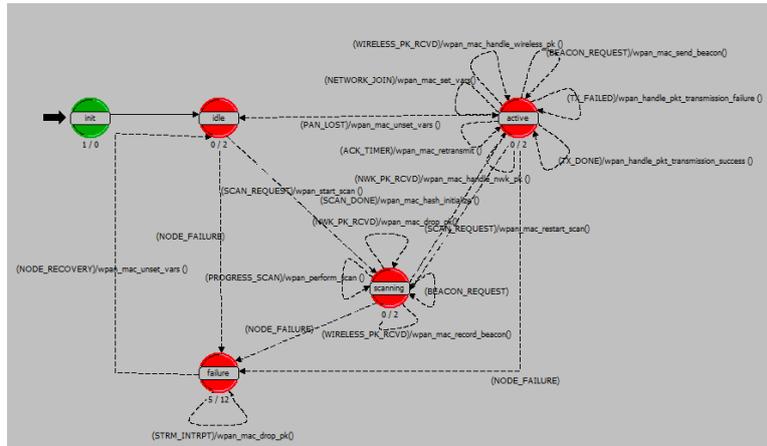


Figure 4.3: ZigBee MAC process model.

Process model contains states, transitions and conditions. A green (forced) state is a state that the process goes to another state when the functions of this state are done (e.g. int state in Figure 4.3). In contrast, the process remains in the red (unforced) state, after finishing their tasks, until an instruction comes or an interrupt occurs indicating that the process should go to another state (e.g. idle state in Figure 4.3). The transitions that connect the states define the conditions for going from one state to another. These conditions and the tasks that should be done when they happen are defined in the process model's function block.

The MAC of the ZigBee node works based on CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance). ZigBee MAC module is made of a main process model which invokes the CSMA/CA as a child process whenever it intends to transmit data. CSMA/CA protocol in OPNET is defined by five states (Figure 4.4): Initialization, idle state, channel checking, back-off state, and transmission.

So now we have introduced the ZigBee model in OPNET. Figure 4.5 is the architectural view of the model structure.

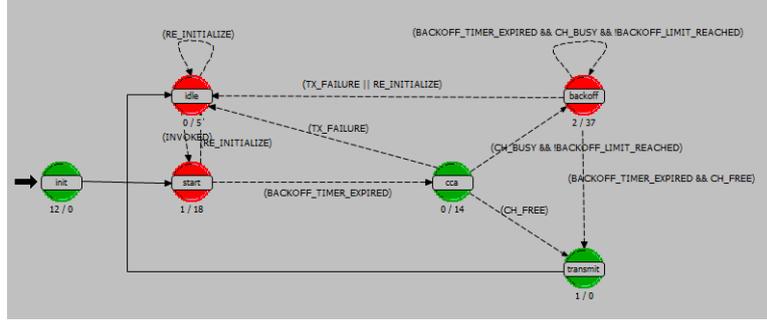


Figure 4.4: CSMA/CA process model.

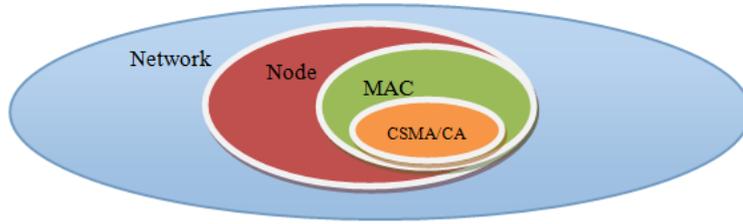


Figure 4.5: ZigBee OPNET model in a nutshell.

#### 4.1.2 Wake-up radio implementation

Our wake-up radio model was built on top of the models we discussed in Section 4.1.1. The difference is, our new PWUR-enabled node model contains a wake-up module along with the main radio module (Figure 4.6). The wake-up module is able to control the communication of the main radio module.

The wake-up module and main radio module operate on separate channels. A wake-up process starts from WuS (wake-up signal) generator at WuTx (coordinator) side generating a WuS. If the signal strength is higher than WuRx’s sensitivity, WuRx(end-node) is able to harvest energy from WuS. Wake-up delay is the time consumed until WuRx harvest enough energy. It is calculated by,

$$WU\_delay = E / (P_r - P_{loss}) \quad (4.1)$$

where E is the energy needed to generate the interrupt voltage (Equation 2.12.2).

The WuS processor calculates the WU\_delay and returns this value to the MAC module of main radio. The main radio does not send its buffered packets until it receives the WU\_delay input from wake-up radio module, suggesting a successful wake-up.

Once a node is woken up, it transmits all the packets in the current buffer. We modified the 802.15.4 CSMA/CA module so that the first transmitted

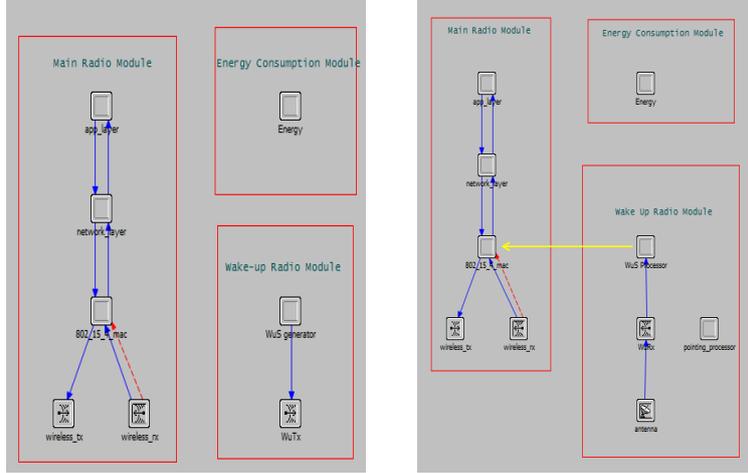


Figure 4.6: Wake-up radio node model (left:WuTx; right:WuRx).

packet's first backoff length equals to the sum of wake-up delay and the original random CSMA/CA backoff. For a retransmission attempt, CSMA/CA still follows its normal back-off mechanism since the wake-up delay has been taken into consideration before its first transmission attempt. Also, for all the other buffered packets other than the first transmitted one, wake-up process does not affect their CSMA/CA backoff. They can start immediately after the previous packet in the buffer is transmitted.

On the other hand, if  $P_r$  of WuS is lower than  $P_{loss}$ , meaning that WuRx is not able to harvest RF energy, thus it is a failed wake-up. Main radio's MAC module buffer the packets from its application, network module until there is a successful wake-up.

So we made two major changes to the original OPNET ZigBee model:

1) Originally, when ZigBee's MAC module receive a packet from its upper layer (network layer), it only checks the availability of MAC and then immediately invokes the CSMA/CA process to transmit the data packet on PHY. Now invoking CSMA/CA is also controlled by WuS processor.

2) The initial CSMA/CA backoff of the first packet transmission takes place after WU\_delay.

The introduction of PWUR affect the sensor node's communication process, which is implemented as the Algorithm 2.

```

Initialization;
for  $i = 1 \rightarrow N$  do
  if i has buffered data then
    if i is woken up by WuTx then
      | after WU_delay, i start its CSMA/CA process;
      | transmit all the packets in the buffer;
    else
      | buffer the data
    end
  else
    | generate new data packet;
    | wait for new wake-up;
  end
end

```

**Algorithm 2:** WuRx’s communication process

### 4.1.3 Simulation assumptions

To simplify our simulations, we made the following assumptions.

- Energy costs for sensing activities are ignored as they will not impact the performance evaluation
- Propagation delay is ignored.
- The link is ideal, meaning that packet is correctly received if the node is within the WuTx’s wake-up range unless collision happens.
- The length of one data packet is 128 bytes. Together with all the overhead, total packet length is 162 bytes.
- The length of a wake-up signal should be long enough to charge the RFID tag on WuRx. If not, we assume multiple WuSs are sent at one go. Also, receiving power stays constant within the same signal.
- If not mentioned specifically, packet rate is set to be 1 s/pkt.
- Wake-signal’s transmission power takes the maximum value allowed by Table 3.1, which is 3.8 W EIRP at the frequency of 868 MHz.

As for other more specific simulation assumptions and requirements, we will introduce them in the correspond section if necessary.

## 4.2 Results and analysis

### 4.2.1 Duty cycling VS PWUR

This section answers Problem 1 in Section 1.4: end-nodes start to receive packets after being woken up, what would be its energy and latency performance compared to the traditional duty cycling approach?

In Chapter 3, we have discussed about our analytical model of duty cycling and PWUR in WSN. Duty cycling and PWUR represent two different approaches to improve the overall performance of WSN, pre-scheduled and on-demand approach. We compare them in terms of both latency and energy consumption. Finally, to make our analysis more applicable to the multiple users scenario, we analyze the performance of the TDMA & CSMA/CA combination approach that we proposed in Section 3.3.3. In this set of simulations focused on latency performance, we first generate a series of packets following the Poisson distribution. Based on the average packet rates, simulations are divided into the high-rate (10 ~ 60 pkt/min) and low-rate (0.5 ~ 3 pkt/min) group, in order to cover a wide range of applications.

#### 4.2.1.1 Latency

In duty cycling, the average packet delay is related to the packet arrival time. In our simulation, packet arrives following Poisson distribution. When the packet arrival time is generated, latency is calculated as in Figure 3.3.

Given a specific duty cycling value, which decides the cycle length according to Equation 3.4, and a specific packet rate., there can be 0, 1, or more than one packet arrives during one cycle length. To calculate packet delay, we can neglect the scenario of 0 packet in one cycle. And when there are more than one packet in one cycle, the average packet delay is the mean value for all packets which arrives to the specific cycle.

To prove our theoretical analysis is valid, we hereby list the delay distribution in Figure 4.7 for the case when there is only one packet in the cycle. Cycle length is 52.784s, 10.557s, 5.278s and 0.525s for 0.1%, 0.5%, 1% and 10% duty cycling respectively (Equation 3.4). We take the 10 % duty cycling scheme as an example, where from Equation 3.2 to 3.4 and Table 3.2, we have

$$T1 = 0.0028s, T2 = 0.05s, T3 = 0.4751s$$

When packets arrive in T1, T2 or T3, packet delay turns out to be within different ranges. Figure 4.7 proves that the probability distribution of packet delay is related to the proportion of T1, T2 and T3 compared to the entire cycle length. For example, for the 10 % duty cycling scheme, probability of packet delay is 0 is  $P_{(d=0)} = T2/T = 0.0947$ . Our simulation result 9.33% complies with the result.

As for the latency of PWUR,  $T_{WU}$  in Equation 3.10 is calculated based on the parameters of circuit in [14], given the wake-up distance is 5 m. Using the parameters listed in Table3.1, we obtained the average packet delay of both duty cycling and PWUR in Figure 4.8.

Figure 4.8 shows the latency performance comparison of duty cycling scheme and PWUR. It is obvious that the lower the duty cycling is, the longer the average packet delay will be. This is due to the longer sleeping

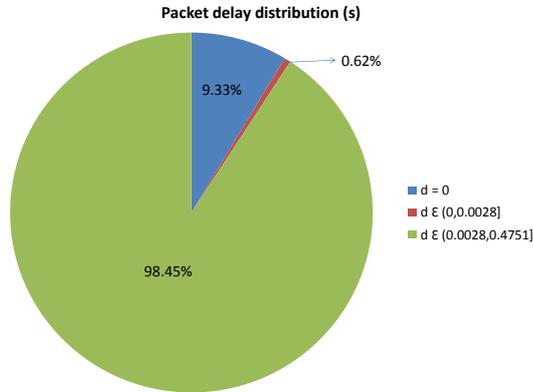


Figure 4.7: Packet delay probability distribution).

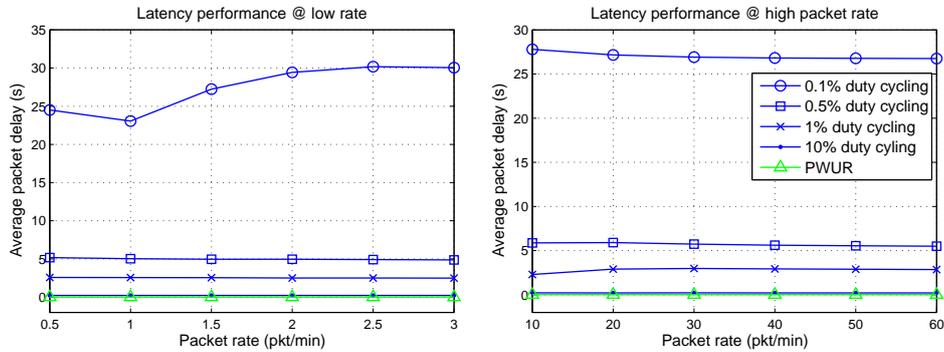


Figure 4.8: Latency performance comparison.

period in each cycle. However, the latency performance of PWUR surpasses the one with lowest duty cycling (0.1%). It proves that the on-demand PWUR approach is indeed superior to handle the delay-sensitive real-time applications.

#### 4.2.1.2 Energy Consumption

In terms of energy consumption, PWUR is supposed to have a better performance since it avoids idle listening in duty cycling which consumes considerable amount of energy. However, this is not always the case. When the packet rate becomes higher, too frequent wake-ups may lead to even higher energy consumption. Our simulation result in Figure 4.9 also proves the point.

In the low packet rate group, as we expected, higher duty cycling will lead to higher energy consumption. PWUR scheme consumes less energy per packet since it only consumes extremely low energy in the sleep mode when there is no packet arriving. However, as shown in Figure 4.9, when

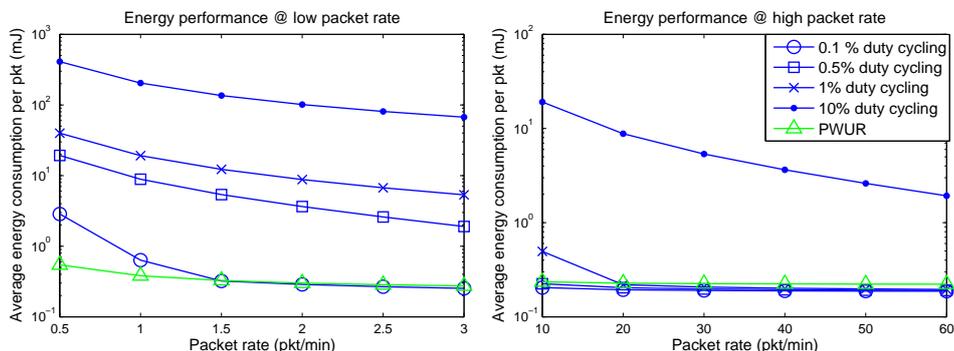


Figure 4.9: Energy consumption performance comparison.

packet rate exceeds 1.5 packets per minute, the advantage of PWUR over low duty cycling scheme starts to vanish. Equation 3.11 explains that the energy consumption per pkt of PWUR does not vary too much with the packet rate. However, when packet rate gets higher, the possibility that no packet arrives during one cycle decreases, thus number of idle listenings decreases which eventually leads to lower average energy consumption per pkt. As the packet rate continues to increase, it could be even more energy-efficient to use low duty cycling scheme rather than PWUR. Because every single packet received by PWUR requires a separate wake-up process which consumes energy. However, multiple packets arrives during one cycle time only needs a single “active” period in duty cycling.

The energy consumption per packet does not vary too much among 0.1%, 0.5% and 1% duty cycling schemes. This is because their duty length  $T$  (52.784s, 10.557s, 5.278s) is comparable or significantly longer than average packet interval  $1/\lambda$  (1 to 6s at high rate). Idle-listening is unlikely to happen. That is why energy consumption performance for these three duty cycling schemes is quite desirable which even surpasses PWUR slightly

#### 4.2.1.3 Multiple nodes scenario

The latency and energy consumption performance analysis above are all based on a per-link manner to investigate the difference between duty cycling and PWUR protocols themselves. However, this is not sufficient if we want to apply our results for analyzing a large WSN where the central coordinator has to communicate with multiple end-nodes. In Section 3.3.3, we proposed a TDMA & CSMA/CA combination approach on top of our analytical model to handle the multiple nodes scenario. In this section, we analyze its performance.

The packet rates range from 10 pkt/min to 60 pkt/min. Duty cycling is 10%. Therefore, end-nodes are divided into 10 clusters. We simulate the scenarios of 10, 30, 50, 100, 150, 200, 250 end-nodes in the network

respectively. So each cluster has 1, 3, 5, 10, 15, 20, 25 nodes. Table 4.1 illustrates the time consumed by CSMA/CA process within a cluster. We run our test program in OPNET and generate the average time (100 iterations) when the last node in the cluster has successfully transmitted its CM to the central coordinator. Results show that as the number of nodes within each cluster increases, CM period in Figure 3.9 becomes longer since the CSMA/CA process for the control message takes longer time. Notify here that the channel load for transmitting CM remains to be around 10% to 15%. If its channel load is higher, meaning that we have more nodes within one cluster, CSMA/CA can not function properly due to massive amount of collisions at the higher channel load.

Number of nodes within each cluster	Time consumed in CSMA/CA process (ms)
1	0
3	22.2
5	38.9
10	88.8
15	138.7
20	162
25	197.6

Table 4.1: CM period length

Figure 4.10 shows the latency performance of multiple nodes scenarios. TDMA first divides all the nodes into 10 clusters. We assume this process does not contribute to the whole delay. Then the packet delay within a cluster increases following the number of nodes. The average packet delay when network size is 250 nodes is about 6 times than there are only 10 nodes in the system.

Figure 4.11 shows the energy consumption performance of multiple nodes scenario. In fact, unlike the latency performance, the number of nodes does not affect too much on the average energy consumption per packet. The reason for that is, in WSN, extra energy consumption is mainly introduced by idle listening. When number of nodes increase, it does not bring extra idle-listenings to each node. The node only needs to spend extra energy on monitoring the channel in the longer CM periods for the purpose of CSMA/CA. This part introduces the slight increment on energy consumption, which is why in a network with 250 nodes, energy consumption per packet is slightly higher than the one with 10 nodes. For energy consump-

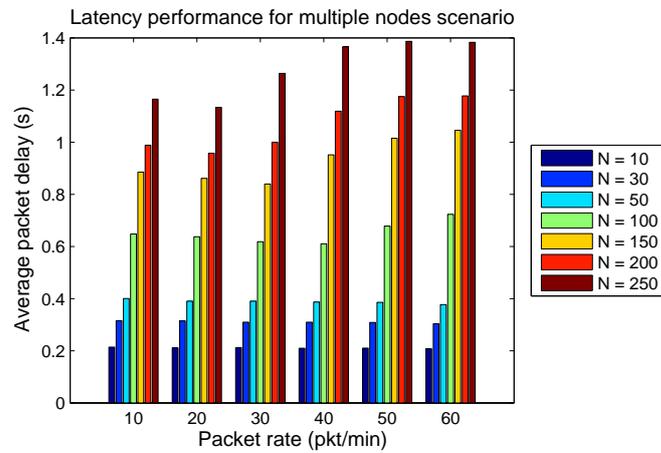


Figure 4.10: Latency performance of multiple nodes scenario.

tion, packet rate is the more decisive factor since it is directly related to the number of idle-listening periods a node has to endure in order to receive one packet.

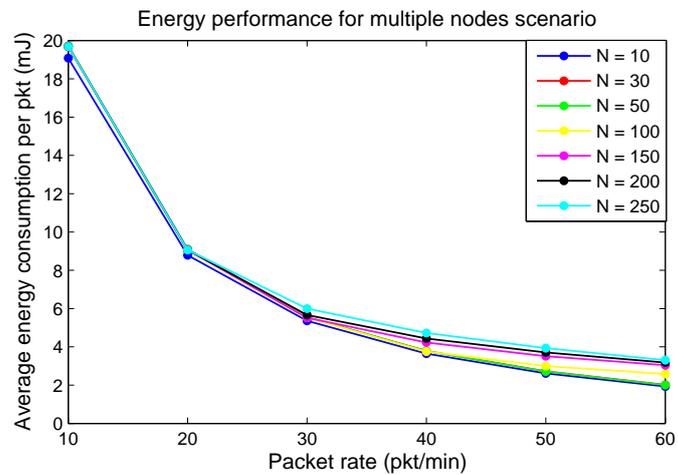


Figure 4.11: Energy consumption performance of multiple nodes scenario.

In summary, we introduce a combination approach of TDMA and CSMA on top of our analytical model, which brings different influence on the latency and energy performance. To be more specific, as the number of nodes increases, latency performance starts to drop, meanwhile the energy performance only suffers very limited impact.

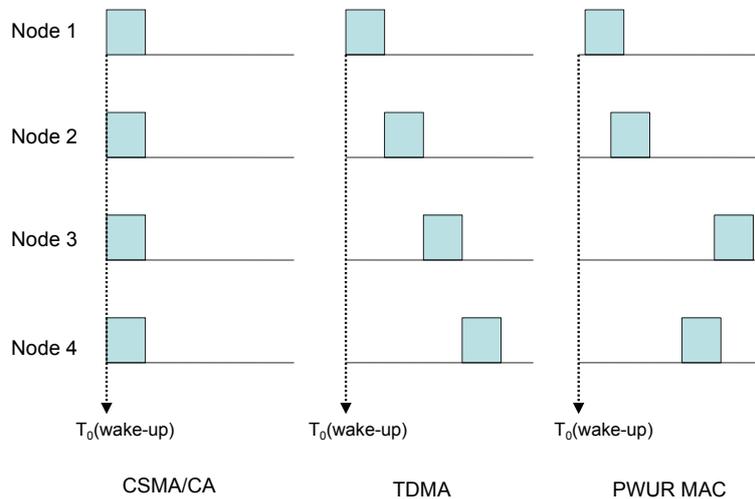


Figure 4.12: Three MAC approaches

#### 4.2.2 PWUR on CSMA/CA

This section answers Problem 2 in Section 1.4: end-nodes start to send packets after being woken up. What would the overall MAC layer performance be when such wake-up mechanism is introduced?

We will be discussing 3 approaches on the MAC layer: TDMA (to send data asynchronously), CSMA/CA (to send data synchronously) and PWUR MAC (to send data semi-synchronously). Figure 4.12 illustrates the difference among these three, which is mainly the starting time of packet transmission. The irregular delay forms of back-off in the PWUR MAC is introduced by the wake-up delay which is related to the received power of the WuS (Wake-up Signal).

##### CSMA/CA

When we introduced in Section 3.3.2 about CSMA/CA process in IEEE 802.15.4, we mentioned that if multiple nodes start CSMA/CA process simultaneously, system performance drops as the number of nodes increases. This is a more typical issue in a range-based wake-up ZigBee network. Coordinator sends wake-up signal (WuS) which could wake up the end-nodes in a certain range. After being woken up, end-nodes transmit packets to the coordinator following the CSMA/CA before the actual transmission of data

packet. Given the path loss of WuS follows the Friis model, nodes within the same range may have very slight difference in terms of their received WuS signal strength. Therefore, their wake-up delay is relatively the same, they will start the CSMA/CA simultaneously. This scenario will bring us two consequences, longer delay due to more back-offs and higher collision rate. Figure 4.13 and 4.14 illustrates delay and collision rate respectively given the number of nodes woken up simultaneously are from 10 to 100. We simulate the process for 5 minutes and the packet rate is 1 pkt/s.

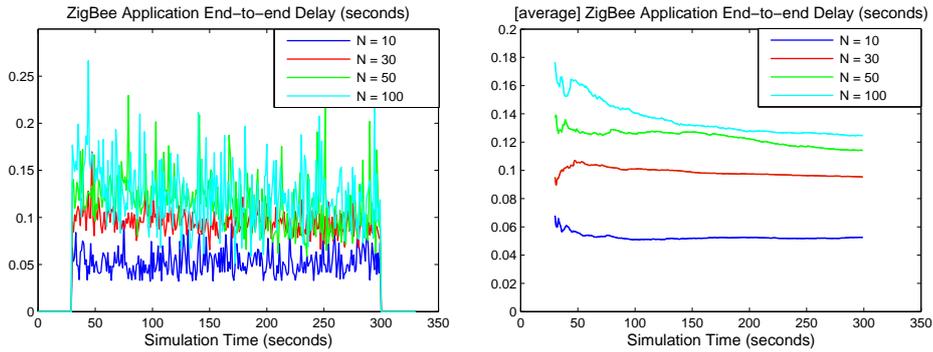


Figure 4.13: Data packet delay (left: real-time value; right: average value).

Figure 4.13 shows the growth of packet end-to-end delay as number of nodes increases. As more nodes competing the medium, more collisions take place. The number of back-offs and retransmissions increases. When retransmission attempts reach the limit, the node has to discard the packet. Figure 4.14 shows that packet delivery percentage drops as more packets are discarded due to collisions. When there are only 10 end-nodes in the network, almost 100% of packets are successfully delivered. However, as  $N$  is 100, the percentage drops to under 20%, meaning that during our 5-minute simulation period, only less than 20% of packets are received successfully by the coordinator.

If we multiply the number of users by the packet delivery percentage, we find out the outcome is always around 20. This suggests that, the full channel load that CSMA/CA can handle is 20 nodes sending packets at the same time (In Figure 4.14, when  $N = 10 < 20$ , delivery percentage is almost always 100%). If the node number exceeds 20, packets are dropped because the channel has been over-loaded.

## TDMA

Instead of starting the CSMA/CA process immediately after the nodes are woken up, we can also use TDMA to coordinate the transmissions of all the nodes in the system. While CSMA/CA is driven by the WuS of the RFID reader (WuTx), TDMA is a tag-driven approach where each WuRx

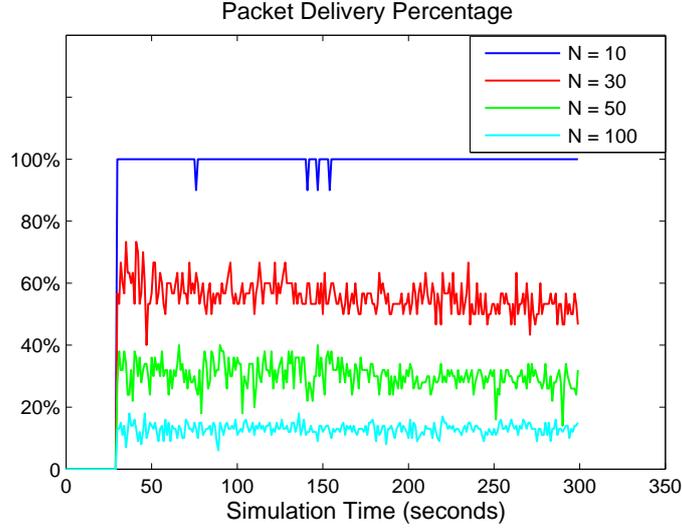


Figure 4.14: Successful packet delivery percentage

has a pre-set sequence number  $m$ , which ranges from 1 to  $N$  (total number of nodes). Suppose one time slot length in TDMA is  $t$ , upon woken up, each node backs off until the time  $mt$ . To avoid any collision, the value of  $t$  should be longer than the maximum possible value of the whole transmission time of one data packet so that slots do not overlap each other. In our simulation, we have

$$t = T_{processing} + T_{mac\_access} + T_{(tx\&rx\_Data)} + T_{(tx\&rx\_ACK)} \geq 0.0075s \quad (4.2)$$

The transmission of each node starts in an asynchronous mode. Its packet delivery percentage is 1 since collisions are all avoided. Figure 4.11 shows the packet delay of this asynchronous approach. The value of packet delay does not vary too much from time due to the absence of random back off delay in CSMA/CA.

### PWUR on CSMA

The previous analysis in the CSMA/CA part is based on the assumption that nodes wake up simultaneously. However, this is not always the case. If two WuRxs distances to WuTx is different such that it takes them different time to harvest enough energy from the WuS. This wake-up delay is applied before the CSMA/CA process starts. So we now have a semi-synchronous scheme for data transmission.

We place all the nodes randomly within the coordinators wake-up range. Each node has its own distance to the coordinator. Their individual wake-up delay can be calculated by Equation 4.1.2. For example,  $C_{se}$  in Figure 2.1 is

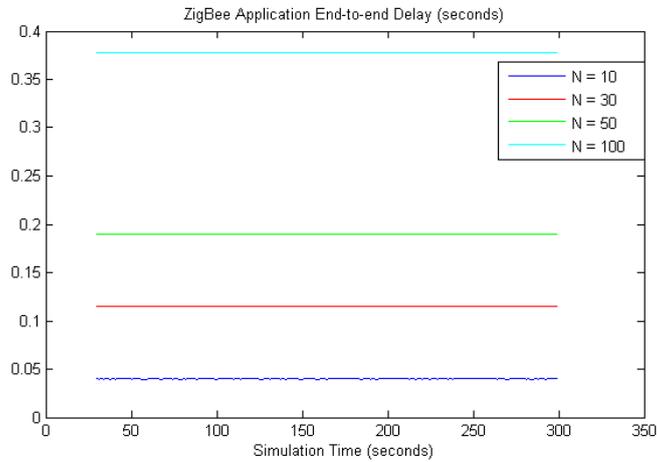


Figure 4.15: Data packet delay (Asynchronous mode)

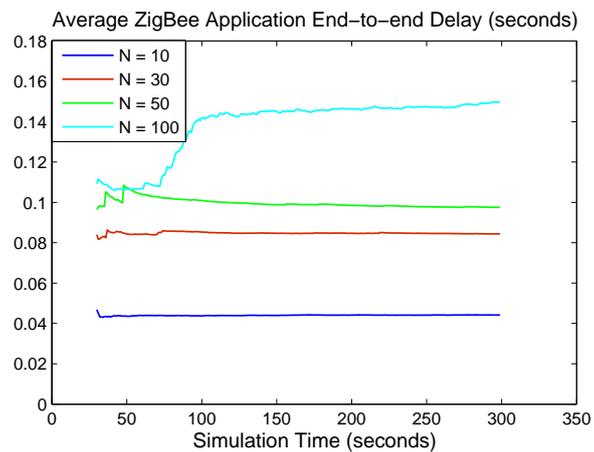


Figure 4.16: Average data packet delay

0.2uF, interrupt voltage is 0.7V, for a node which is 10 meters away from the coordinator, its wake-up delay is 4.6ms. For a node with 5 meters distance to WuTx, its wake-up delay is 1.1ms. Thus the CSMA/CA starting time for this two nodes has a 3.5ms shift.

Packet delay and successfully delivery rate are shown in Figure 4.16 and 4.17.

#### 4.2.2.1 Effective Packet Rate (EPR)

We have analyzed both packet delay and delivery percentage for all three MAC layer approaches: CSMA/CA (synchronous), TDMA (asynchronous) and PWUR on CSMA/CA (semi-synchronous). If we only look at the delay performance, we would find out that TDMA's performance is worse than the

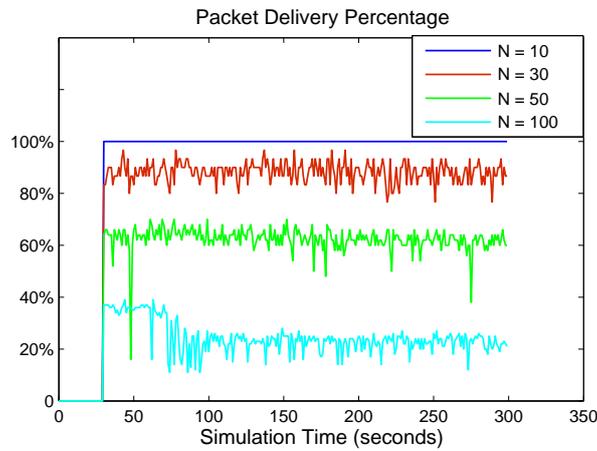


Figure 4.17: Successful packet delivery percentage

other CSMA/CA-based approach. Especially when there are a large number of nodes ( $N = 100$ ), its end-to-end packet delay is more than twice of the other two. Between CSMA/CA and PWUR MAC, there is no significant difference. (Table 4.2 )

Node Number \ MAC	CSMA/CA	TDMA	PWUR MAC
10	0.052	0.040	0.042
30	0.100	0.118	0.082
50	0.116	0.190	0.098
100	0.124	0.376	0.150

Table 4.2: Average packet delay (s)

However, latency alone does not represent the network overall reliability. In fact, the parameter of packet delay can only measure those successful delivered packets. It cannot reflect all the discarded packets. Therefore, to compare the overall performance, we need a new parameter that describes the network better.

Effective Packet Rate (EPR) of a network is the number of successfully delivered packets per second by all the nodes. It is calculated in Equation 4.3.

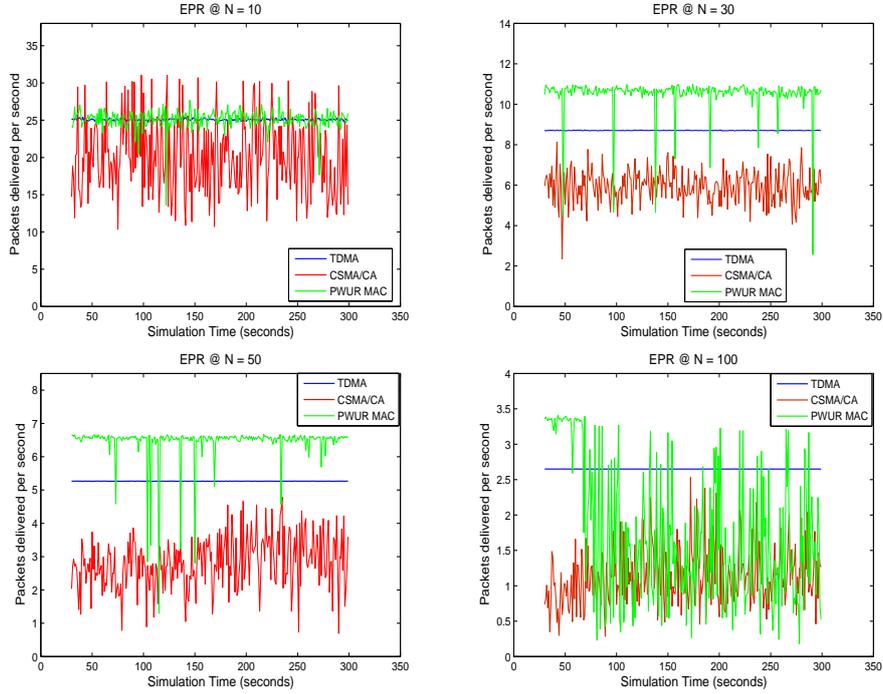


Figure 4.18: EPR comparison (real time value)

$$EPR = p \times \frac{1}{T} + (1 - p) \times \frac{1}{\infty} = \frac{p}{T} \quad (4.3)$$

$p$  is the packet delivery percentage, therefore  $1 - p$  is the probability that a packet is discarded.  $T$  is the packet end-to-end delay and  $\frac{1}{T}$  represents the packet rate of the successful transmissions (in pkt/s). When a packet is not successfully delivered, its end-to-end delay is  $\infty$ , therefore the packet rate is 0. The final result  $EPR = \frac{p}{T}$  is the mean value of packet rate. It shows that the Effective Packet Rate is dependent on both packet delay and delivery percentage.

For example, in a network, successfully transmitting one packet costs 10 ms on average. And only 70% transmissions are successful. By Equation 4.3, we have  $EPR$  equals to 70 pkt/s. 70 packets are successfully transmitted in 1 second. The concept of EPR is close to network throughput.

Figure 4.18 is the EPR comparison of three MAC layer approaches corresponds to different nodes densities.

Figure 4.18 shows the general trend of EPR value. EPR of CSMA/CA and PWUR MAC have a higher variance than TDMA. The average EPR are shown in Figure 4.19.

We can draw the following observations from Figure 4.19:

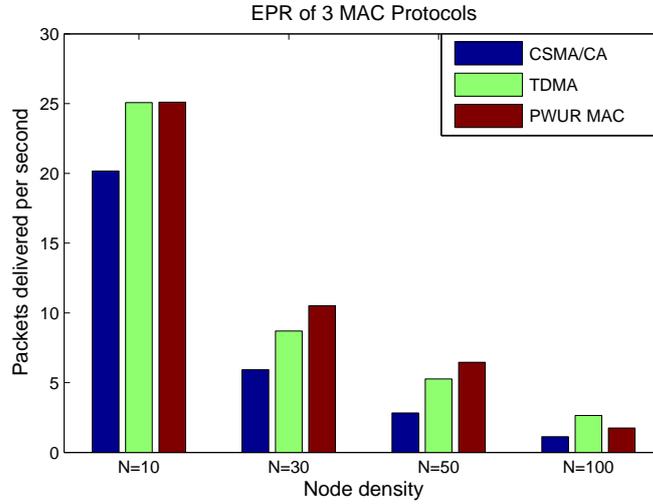


Figure 4.19: EPR comparison (average value)

1) Both TDMA and PWUR MAC outperform CSMA/CA of 802.15.4 in EPR.

2) When node number is relatively small, TDMA and PWUR MAC almost reach the maximum number of packets the network can possibly transmit. The EPR of TDMA and PWUR MAC at  $N = 10$  are both 25.1 pkt/s, while EPR of CSMA is 20.2 pkt/s. The margin to the optimal EPR in TDMA is introduced by the length of time slot. We set it to be the maximum possible time for one data transmission which brings overhead. The margin to the optimal EPR in PWUR MAC is introduced by the back-offs and retransmissions in CSMA/CA. Although it has been improved massively compared to the complete synchronous CSMA/CA, overhead still exists.

3) As node number increases, PWUR MAC starts to have higher EPR than asynchronous mode. Because, it does not wait for theoretical maximum interval between 2 transmissions. At  $N = 30$  and  $N = 50$ , the EPR of PWUR MAC is 77% and 128% higher than CSMA/CA, 21% and 22% higher than TDMA. However, when node density becomes too high, PWUR MAC cannot handle the collisions effectively anymore due to collisions. That is why at  $N = 100$ , TDMA has a slightly higher EPR than PWUR MAC.

4) The performance of PWUR MAC is related to wake-up delay of each WuRx.

#### 4.2.2.2 Influence of wireless channel performance

In the analysis above, propagation loss of WuS follows the Friis model in Equation 4.4.

$$P_r = P_t + G_t + G_r + 20 \log_{10}\left(\frac{\lambda}{4\pi d}\right) \quad (4.4)$$

where the distance between WuRx and WuTx determines the WuS signal strength. It is good enough if we simply want to test the MAC layer performance of our PUWR model. The wireless channel performance on the PHY layer is simplified. However, in the real-case scenario, the channel performance is a massively important factor in any wireless networks. Especially in our PWUR-based WSN, wake-up delay is directly related to  $P_r$  of WuS. It is worthwhile to use a separate section to investigate, when a more realistic propagation loss model is applied, what the influence of wireless channel performance will be.

We chose the log-distance path loss model in Equation 4.5.

$$PL = P_t - P_r = PL_0 + 10\gamma \log_{10} \frac{d}{d_0} + X_g \quad (4.5)$$

where  $PL$  is the total path loss,  $PL_0$  is the path loss at the reference distance  $d_0$ ,  $\gamma$  is the path loss exponent,  $X_g$  is a Gaussian random variable with zero mean, reflecting the attenuation caused by flat fading. In case of no fading, this variable is 0. In case of only shadow fading or slow fading, this random variable may have Gaussian distribution with  $\sigma$  standard deviation in dB, resulting in log-normal distribution of the received power in Watt. In an indoor environment like our case,  $\gamma$  is set to be 2.76 and  $X_g$  is 6 dB [28].

Given the distance between WuRx and WuTx, the signal strength of WuS is not fixed anymore. Nodes can be woken up when  $P_r$  of the WuS is higher than the WuRx circuits sensitivity level (-29.3 dBm). Figure 4.20 shows the theoretical and our simulation result on wake-up probability at different ranges.

The theoretical result was obtained by calculating the probability that received power of WuS was above -29.3 dBm. And Simulation result was obtained by OPNET simulations of calculating the number of successful wake-ups out the total number of iterations.

It can be seen that wake-up probability drops as distance increases. Between 0~4 meters, wake-up probability is almost 1. When the distance becomes 20 meters, the wake-up probability is around 0.2, meaning that when WuRx is 20 meters away from the coordinator, it almost take 5 WuSs to wake up the end-node once. The end-node continues to generate data packet at the rate of 1 pkt/s. It buffers all the packets until its WuRx successfully harvested enough energy from a WuS. Given the same nodes distribution as in the previous sections, we have the packet delay shown in Figure 4.21 using the log-distance path loss model.

The end-to-end delay can be as much as 2 to 18 seconds. This is significantly higher compared to the results with Friis model which stays around 0.1s (Figure 4.13). Failed wake-ups forced more packets into the buffer.

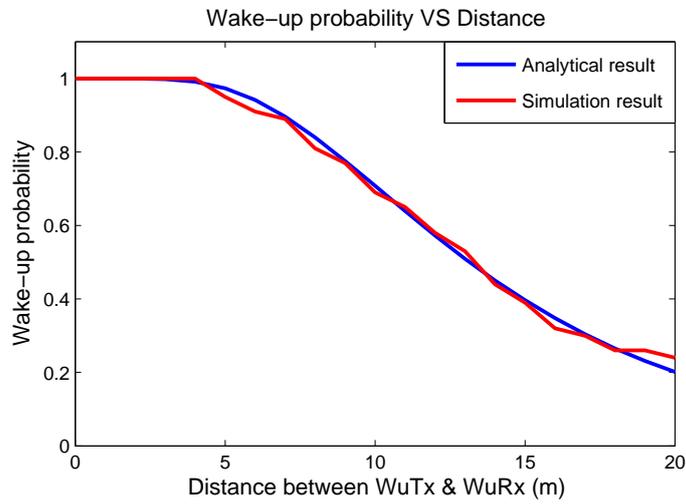


Figure 4.20: Relation between wake-up probability and distance

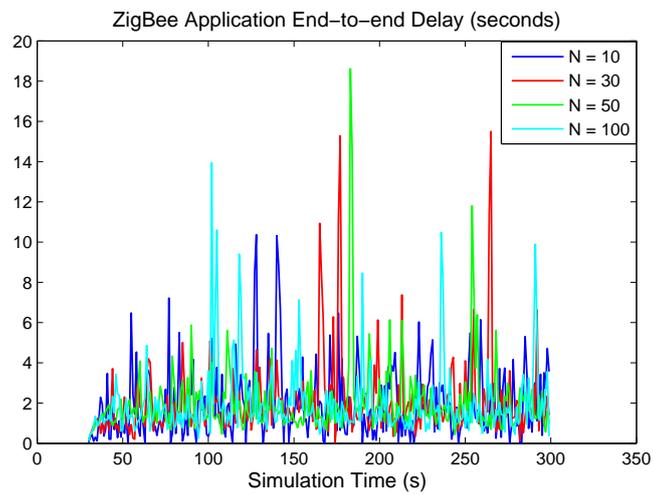


Figure 4.21: Packet delay with log-distance path loss model

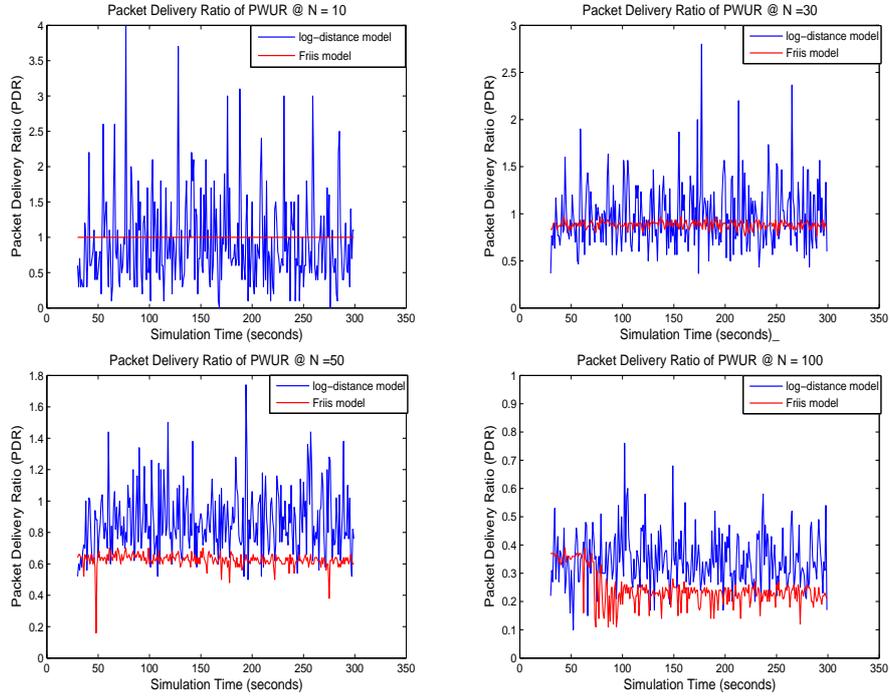


Figure 4.22: Influence of wireless channel performance on packet delivery ratio.

This already increases the packet delay. Moreover, when there is a successful wake-up, packets in the buffer take part in the medium contention with the other nodes, which leads to more collisions and back-offs.

The comparison of packet delivery ratio by using Friis model and log distance model to calculate propagation loss is shown in Figure 4.22. It is worth mentioning that, since multiple packets in the buffer can be transmitted after one single wake-up, the packet delivery ratio can surpass 1.

Deviation of packet delivery ratio using log-distance model is much more severe than the one with Friis model. It is predictable since the log-distance pathloss model make the channel more dynamic. Overall, when node density is high, a more dynamic wireless channel actually provide better packet delivery ratio.

In conclusion, when we apply a more realistic propagation loss model to our PWUR model, we find out that the packet delay performance becomes significantly worse, while the delivery ratio performance maintains similar or sometimes even better. In total, the EPR (network throughput) drops. In the real-world wireless channel, channel performance is even more dynamic and unpredictable, PWURs overall network performance may be even worse.

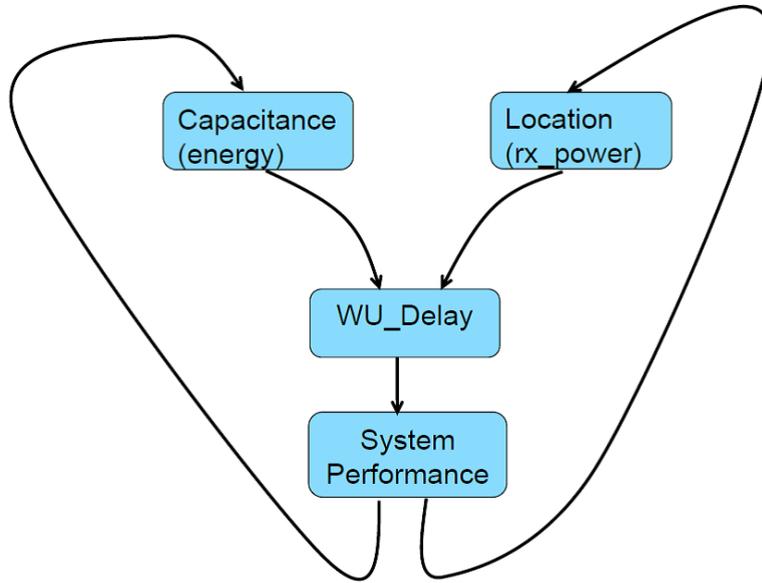


Figure 4.23: Network performance optimization loop

#### 4.2.2.3 Optimization loop

The wake-up delay of PWUR functions as an additional back-off before the CSMA/CA back-off starts. Since the back-off length differs from node to node, it improves the MAC layer performance by decreasing the total back-off delay and avoiding collisions in CSMA/CA. We find out that there is an optimization loop in this process.

From Equation 2.1 and 2.2, we understand that necessary amount of harvested energy  $E$  and the receive power of wake-up signal  $P_r$  will determine how long the wake-up process take.  $E$  is dependent on the capacitance of the capacitor that stores the energy.  $P_r$  is dependent on the location, the distance between WuRx and WuTx. Wake-up delay is calculated by  $E / P_r$ .

If we choose the right capacitance and location of WuRx, the wake-up delay of each node will lead to a better system performance. Vice versa, we can optimize both the PWUR circuit design and nodes distribution based on system performance. An optimization algorithm can be developed following this principle

## Chapter 5

# Conclusions and Future Work

### 5.1 Conclusions

The primary goal of this thesis work is to investigate how PWUR can affect the ZigBee network's overall performance. We use OPNET and MATLAB to build our simulations and focus on two specific problems to approach our goal.

In terms of the comparison of duty cycling and PWUR, we find out that PWUR can indeed break the energy-latency tradeoff in duty cycling. As for the latency performance, the on-demand PWUR outperforms the pre-scheduled duty cycling. However, as packet rate increases, energy consumption performance of low duty cycling scheme, which also avoids the idle-listening activity, is able to compete with PWUR. We also proposed our CSMA/CA + TDMA combination scheme to handle the multiple nodes scenario. When the number of nodes in a network increases, latency performance gets worse, while the node number does not affect energy consumption performance that much.

In terms of PWUR's influence on the MAC layer performance, we find out that the introduction of PWUR can improve the network throughput due to the initial wake-up delay before CSMA/CA back-off starts. However, the network size, nodes location, hardware design and pathloss model determine how much the performance improvement can be.

### 5.2 Future Work

A number of additions on this topic can be introduced in the future work.

First, in Section 3.3, we proposed the combination scheme of TDMA and CSMA/CA. This scheme can be optimized by investigating the relation between cluster size and network size. Clusters are divided by TDMA. Nodes

within the same cluster are divided by CSMA/CA. Both these two process include overhead. By selecting the right cluster size, the sum of TDMA and CSMA/CA's overhead will be minimized.

Second, in Section 4.2.2, we mentioned the optimization loop introduced by wake-up delay. Later study can focused on proposing an optimization algorithm to help both the circuit design and network distribution of PUWR-based WSN.

Last but not least, in this work, there are no real measurement data included. In the future, instead of purely based on theoretical analysis, similar study can make use of measured data by devices like WISP to make it more realistic.

# Bibliography

- [1] <http://www.zigbee.org/>.
- [2] <http://www.bluetooth.com/Pages/Bluetooth-Home.aspx>.
- [3] [http://en.hartcomm.org/hcp/tech/articles/wiHART\\_resources/witech\\_witechnote.html](http://en.hartcomm.org/hcp/tech/articles/wiHART_resources/witech_witechnote.html).
- [4] <https://www.isa.org/isa100/>.
- [5] <http://www.nxp.com/techzones/wireless-connectivity/jn51xx/jn516x.html>.
- [6] nrf24l01 single chip 2.4GHz transceiver product specification. <http://www.nordicsemi.com/index.cfm?obj=product&act=display&pro=89#>.
- [7] OPNET. <http://www.riverbed.com/products/performance-management-control/opnet.html>.
- [8] EPCglobal, UHF Class-1 Generation-2 Standard. [http://www.gs1.org/gsm/kc/epcglobal/uhfc1g2/uhfc1g2\\_1\\_2\\_0-standard-20080511.pdf](http://www.gs1.org/gsm/kc/epcglobal/uhfc1g2/uhfc1g2_1_2_0-standard-20080511.pdf), 2008.
- [9] IEEE 802.15.4 mac/phy standard for low-rate wireless personal area networks(lrwpans). <http://www.ieee802.org/15/pub/TG4.html>, 2013.
- [10] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, August 2002.
- [11] Th. Arampatzis, J. Lygeros, and S. Manesis. A survey of applications of wireless sensors and wireless sensor networks. pages 719–724. 2005 IEEE International Symposium, Mediterrean Conference on Control and Automation Intelligent Control, 2005.
- [12] He Ba, Ilker Demirkol, and Wendi Heinzelman. Passive wake-up radios: From devices to applications. *Ad Hoc Networks*, 11:2605 – 2621, August 2013.
- [13] L. Chen, S. Cool, H. Ba, W. Heinzelman, I. Demirkol, U. Muncuk, K. R. Chowdhury, and S. Basagni. Range extension of passive wake-up radio systems through energy harvesting. pages 1549 – 1554. Proc. IEEE ICC Budapest, Hungary, June 2013.
- [14] H. J. Cho, H. C. Kim, X. Yao, M. S. Kim, S. W. Kwon, T. J. Park, H. S. Kim, and Y. G. Yang. Highly sensitive cmos passive wake-up circuit. pages 1 – 4. Proc. Asian Pacific Microw. Conf. (APMC 2008), 2008.
- [15] W. Dargie and C.Poellabauer. *Fundamentals of wireless sensor networks: theory and practice*, pages 168 – 183 , 191 – 192. John Wiley and Sons, 2010.
- [16] J. Demirkol, C. Ersoy, and E. Onur. Wake-up receivers for wireless sensor networks: benefits and challenges. *IEEE Wireless Communications*, 16(4):88 – 96, 2009.

- [17] B.V.d. Doorn, W. Kavelaars, and K. Langendoen. A prototype low-cost wakeup radio for the 868 mhz band. *International Journal on Sensor Networks*, 5:22 – 32, 2009.
- [18] L. Gu and J. Stankovic. Radio-triggered wake-up capability for sensor networks. pages 27 – 36. Real-Time and Embedded Technology and Applications Symposium, 2004, IEEE, 2004.
- [19] R. Jurdak, A. G. Ruzzelli, and G. M. P O’Hare. Multi-hop rfid wake-up radio: design, evaluation and energy tradeoffs. pages 1 – 8. Proc. of 17th ICCCN Conference, IEEE, Aug 2008.
- [20] A. Khatibi, Y. Durmus, E. Onur, and I. Niemegeers. Event-driven mac protocol for dual-radio cooperation. pages 1 – 5. Vehicular Technology Conference (VTC Fall), IEEE, September 2012.
- [21] P. Le-Huy and S. Roy. Low-power 2.4 ghz wake-up radio for wireless sensor networks. pages 13–18. Int. Conf. on Wireless and Mobile Computing, Networking and Communication, 2008.
- [22] Sang Hoon Lee, Yong Soo Bae, and L Choi. On-demand radio wave sensor for wireless sensor networks: Towards a zero idle listening and zero sleep delay mac protocol. pages 560 – 566. Global Communications Conference (GLOBECOM), IEEE, December 2012.
- [23] H. Liu, M. Bolic, A. Nayak, and I. Stojmenovic. Taxonomy and challenges of the integration of rfid and wireless sensor networks. *IEEE Network*, 22(6):26 – 32, 2008.
- [24] M. Lont, D. Milosevic, P. G. M. Baltus, A. H. M. Van Roermund, and G. Dolmans. Analytical models for the wake-up receiver power budget for wireless sensor networks. pages 1–6. Proc. IEEE Global Telecommun. Conf. (GLOBECOM’09), 2009.
- [25] Charith Perera, Arkady Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos. Context aware computing for the internet of things: A survey. *IEEE Communications Surveys and Tutorials*, 16(1):414 – 454, 2013.
- [26] N. Pletcher, S. Gambini, and J. Rabaey. A 65 uw, 1.9 ghz rf to digital baseband wakeup receiver for wireless sensor nodes. pages 539–542. Proc. Custom Integr. Circuits Conf., 2007.
- [27] P. Radmand, A. Talevski, S. Petersen, and S. Carlsen. Comparison of industrial wsn standards. pages 632 – 637. 4th IEEE International Conference on Digital Ecosystems and Technologies (DEST), 2010.
- [28] T. S. Rappaport. *Wireless communications principles and practices*. Prentice-Hall, 2002.
- [29] A.P. Sample, D.J. Yeager, P.S. Powledge, A.V. Mamishev, and J.R. Smith. Design of an rfid-based battery-free programmable sensing platform. *IEEE Transactions on Instrumentation and Measurement*, 57(11):2608 – 2615, 2008.
- [30] L. Shan, T. Yunjian, and Z. Qin. Passive wake-up scheme for wireless sensor network. page 507. Proc. 2nd Int. Conf. Innovative Comp., Inform. Control, 2007.
- [31] K. Sohraby, D. Minoli, and T. Znati. *Wireless sensor networks: technology, protocols, and applications*, pages 203 – 209. John Wiley and Sons, 2007.
- [32] P. Sthapit and Jae-Young Pyun. Effects of radio triggered sensor mac protocol over wireless sensor network. pages 546 – 5516. Computer and Information Technology (CIT), IEEE, 2011.
- [33] L. Zhang and Z. Wang. Integration of rfid into wireless sensor networks: Architectures, opportunities and challenging problems. pages 433 – 469. Proc. Grid Coop. Comput. Workshops, 2006.

- [34] Y. Zhang, L. Huang, G. Dolmans, and H. de Groot. An analytical model for energy efficiency analysis of different wakeup radio schemes. pages 1148 –1152. Personal, Indoor and Mobile Radio Communications, IEEE, 2009.