# Ultra Low-Power MAC Layer Wake-Up-Frame Scheme for Low-Cost and Low-Traffic Wireless Sensor Networks

Der Fakultät für Elektrotechnik und Informationstechnik
der Universität Dortmund vorgelegte

Dissertation

zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften

von

Xiaolei Shi
geboren am 26. Juli 1974 in Shanghai

| | |
|---|---|
| 1. Gutachter: | Prof. Dr.-Ing. Uwe Schwiegelshohn |
| 2. Gutachter: | Prof. Dr.-Ing. Rüdiger Kays |
| Dekan: | Prof. Dr.-Ing. Edmund Handschin |

| | |
|---|---|
| Tag der Einreiching: | 1 Juni 2005 |
| Tag der Prüfung: | 14 Dezember 2005 |
| Stand von: | 4 März 2006 |

*To my wife Lan Chen and my daughter Wenyue Shi.*

献给我的妻子陈岚和女儿施文悦

# Abstract

Wireless sensor networks, as a key enabling technology of Ubiquitous Computing, have been a booming research topic in the recent years. Upon designing a low-cost wireless sensor device, power consumption is one of the most important issues, because cheap batteries are normally the power suppliers. Since the RF transceiver is one of the biggest power consumers in such a sensor device, enabling the RF transceiver to sleep as much as possible is the preferred method to save power, which is normally realized by MAC layer duty cycle scheduling.

This dissertation proposes a MAC layer wake-up-frame scheme to wake up an RF transceiver on-demand to minimize the standby waiting time in receive mode to save power. Analytical and simulation results show that, for a low-traffic wireless sensor network, this scheme gives significant system battery lifetime gain compared to the traditional methods.

Furthermore, the combination of the wake-up-frame scheme and a complementary low-power MAC protocol is discussed. Analytical computation and simulation prove that the combined scheme achieves a further optimized solution in the sense of power-saving, while other important system parameters, such as response time and channel efficiency, are limited to a reasonable range.

***Keywords:*** low-power MAC, wireless sensor network, duty cycle scheduling.

# Acknowledgements

I would like to thank Dr. Guido Stromberg for his many suggestions and constant support during this research. I am also thankful to Prof. Uwe Schwiegelshohn for his kind supervision.

I would also thank Dr. Werner Weber, the head of the Emerging Technologies Laboratory in Corporate Research at Infineon Technologies, for his kind leadership, and my other colleagues Yvonne Gsottberger, Thomas Sturm, Jan Dienstuhl, Daniel Bichler, Christl Lauterbach, Stephan Jung, Rupert Glaser, Markus Schnell, Eike Ruttkowski, Bernhard Knoll, Christoph Braun, Domnic Savio, and Martin Verbeek for their support and friendship.

I would like to also mention Ms. Chong Cai for the contribution of her master thesis, and Mr. Mario Hernan Castaneda Garcia for his excellent job as a working student.

Of course, I am grateful to my wife for her love and care, and to my daughter (born on 2004.11.04), who brings me a lot of happiness and takes me also a large amount of time. :)

I would also thank my mother and my parents-in-law. They helped me a lot to take care of the baby.

Finally, I wish to thank the following: Edwin Naroska, Holger Linde, Peter Schramm, Peter Resch, and Jörg Platte.

Xiaolei Shi
June 2005

# Table of Contents

# List of Tables

# List of Figures

# Glossary

| | |
|---|---|
| **ACK** | Acknowledgement, 3.3 |
| **ARQ** | Automatic Repeat reQuest, 3.1 |
| **ASK** | Amplitude Shift Keying, 3.2 |
| **BER** | Bit Error Rate, 6.2 |
| **BI** | Beacon Interval in IEEE802.15.4 MAC protocol, 3.3 |
| **BO** | Beacon Order in IEEE802.15.4 MAC protocol, 3.3 |
| **CAP** | Contention Access Period in IEEE802.15.4 MAC protocol, 3.3 |
| **CDF** | Cumulative Distribution Function, 6.2 |
| **CFP** | Contention Free Period in IEEE802.15.4 MAC protocol, 3.3 |
| **CRC** | Cyclic Redundancy Check, 4.2 |
| **CSMA** | Carrier Sense Multiple Access, 3.3 |
| **DRD** | Data Rate Detection function provided by TDA525x RF transceiver, 4.1 |
| **ECC** | Error Correction Coding, 3.1 |
| **FPGA** | Field Programmable Gate Array, 3.2 |
| **FSK** | Frequency Shift Keying, 3.2 |
| **GTS** | Guaranteed Time Slot in IEEE802.15.4 MAC protocol, 3.3 |
| **GUI** | Graphic User Interface, 2.1 |

**HARD** Hard partitioned case in indoor wireless channel, 6.2

**ISI** Inter-Symbol-Interference, **6.2**
**ISM** Industrial Scientific and Medical, **3.2**

**LOS** Line-Of-Sight, **6.2**

**MAC** Media Access Control, **1.0**
**MANET** Mobile Ad hoc NETwork, **3.1**

**OBS** OBStructed, **6.2**
**OPEN** Open plan case in indoor wireless channel, 6.2

**PDF** Probability Density Function, **6.2**
**PHY** Physical layer, **3.3**

**RF** Radio Frequency, **1.0**
**RSSI** Received Signal Strength Indicator, **3.3**
**RTC** Real Time Clock, **3.2**

**SIRCIM** Simulation of Indoor Radio Channel IMpulse response, **6.2**
**SNR** Signal-to-Noise Ratio, **6.2**
**SOFT** Soft partitioned case in indoor wireless channel, 6.2
**SWUF** Short Wake-Up-Frame, **4.2**

**TDMA** Time Division Multiple Access, **3.4**

**UPnP** Universal Plug and Play, **2.3**

**WiseMAC** **Wi**reless **Se**nsor **MAC**, 3.3
**WiseNET** **Wi**reless **Se**nsor **NET**work, 2.2
**WPAN** Wireless Personal Area Network, **3.3**
**WSN** Wireless Sensor Network, **2.2**

**WUF** Wake-Up-Frame, **4.0**

**WUP** Wake-Up-Preamble, **3.3**

**WUS** Wake-Up-Signal, including WUP, WUF, and data repetition, **3.3**

**XML** eXtensible Markup Language, **2.3**

# Chapter 1

# Introduction

Ubiquitous Computing has emerged as a hot research topic in the recent years. It integrates many computers and electronic devices into our living and working environments, automatically providing us with services according to our demands without attracting our attention. Ubiquitous Computing touches most aspects of the modern information and communication technologies from the applications, man-machine-interface, data models, semantic protocols, and security down to low-layer communication protocols and hardware design.

In Ubiquitous Computing, a large amount of devices integrated into the environment are small low-cost sensors, which collect various environmental information for the needs of the upper-layer applications. Since the sensors shall be unobtrusive, wireless communication is the preferred solution to deliver these information from the low-cost sensors to central control computers or the like. For this reason, wireless sensor networks become a key enabling technology for Ubiquitous Computing.

One of the most important issues in a wireless sensor network is power consumption because the sensors are mostly powered by low-cost batteries and frequently changing the batteries for a large number of devices is obviously unacceptable. Therefore, many researchers are striving to reduce the power consumption from different aspects of the system design.

Among many possible power saving methods, minimizing the power consumed by the Radio Frequency (RF) transceiver is a very efficient way, because the RF transceiver is normally the most or one of the most power hungry components in such a small sensor

device. Traditionally, an RF transceiver has to keep staying in receive mode to wait for possible incoming packets. This is extremely inefficient for a low traffic wireless sensor network, since most of the energy is consumed by standby waiting instead of data communication. To this end, some approaches were proposed to enable the RF transceiver to sleep more by scheduling the duty cycle of the RF transceiver in the Media Access Control (MAC) layer.

This dissertation proposes an ultra low-power MAC layer protocol for low-cost and low-traffic wireless sensor networks, called the wake-up-frame scheme. It wakes up the RF transceiver on-demand by a wake-up-signal to minimize the standby waiting time in receive mode. Analytical and simulation results show that this scheme achieves a significant battery lifetime gain compared to the traditional schemes.

Furthermore, the wake-up-frame scheme and another low-power MAC protocol called WiseMAC are based on similar technology but are complementary to each other. Thus, the combination of these two schemes could generate an even better overall solution. To prove this idea, analytical computations and simulations have been conducted to compare the performance of the new combined scheme to the individual schemes. The results show that the combined scheme gives the lowest power consumption. Concurrently, other important system parameters, such as response time and channel efficiency, are also limited to a reasonable range.

This dissertation is organized as follows:

- **Chapter 1** provides the technical background of this dissertation by giving an introduction to Ubiquitous Computing, wireless sensor networks, and a so-called Sindrion system. The Sindrion system is a sensor network that bridges the small low-cost sensor devices to UPnP (Universal Plug and Play) networked high-end devices. Note that the proposed wake-up-frame scheme has been designed for the Sindrion system, but could also be applied to other wireless sensor networks.

- **Chapter 2** discusses the power consumption issue of wireless sensor networks and defines the problem to be solved by this dissertation. Some related work is introduced, and their benefits and disadvantages are discussed.

- **Chapter 3** first describes the wake-up-frame scheme from hardware architecture to protocol details. Then, the combination of the wake-up-frame scheme and the

WiseMAC is discussed.

- **Chapter 4** conducts the analytical computation to estimate the gain and cost of the wake-up-frame scheme and the combined scheme.
- **Chapter 5** introduces the simulation models including the indoor wireless channel model, the communication protocols, and the application traffic model.
- **Chapter 6** illustrates and analyzes the simulation results.
- **Chapter 7** finally concludes the dissertation.

# Chapter 2

# Wireless Sensor Networks

## 2.1 Ubiquitous Computing

In 1991, Mark Weiser first addressed the concept of Ubiquitous Computing [1–3], which is named as the third wave in computing after the mainframes era and the personal computing era.

In Ubiquitous Computing, a large number of computers or other electronic devices with computing ability are pervasively embedded into the environment so that they become a part of the background, attracting no human attention any more. Unlike personal computers, Ubiquitous Computing is invisible to users, but it is everywhere and automatically provides services meeting user's demands. That is the reason why it is also called Pervasive Computing or Invisible Computing.

Ubiquitous Computing touches almost every aspect of the modern silicon-based information and communication technology. Some examples are given as follows:

- Ubiquitous Computing needs huge efforts to create a new relationship between humans and computers. It is not just a problem of man-machine-interface such as a friendly Graphic User Interface (GUI), but a property of the whole context of usage of the machine, which addresses many sophisticated technologies such as machine learning, data mining, and so on.
- Interacting with computers everywhere induces also severe privacy problem for an individual. Consequently, security technology will play a more and more important

role in the third computing era.

- The distributed computers and devices need to communicate with each other to deliver information, so wireless communication becomes a key enabling technology for Ubiquitous Computing.

- A large quantity of computers and devices means that the cost per device is an important issue, and invisibility means that minimization is also very important. Therefore, new semiconductor technologies are desired to lower the cost and shrink the size of the device.

Among these examples, the wireless communication, more specifically low-cost and low-power wireless sensor networks, is the scope addressed by this dissertation.

## 2.2 Wireless Sensor Networks

The devices in a Ubiquitous Computing environment comprise a large number of sensors. They collect various environmental information such as temperature, humidity, brightness, and motion of objects to support applications like smart house or intelligent building. This information should be delivered to a central computer or device for further processing.

Since connecting such a large number of low-cost sensors by wires is in most cases infeasible, wireless communication becomes the only choice for the transportation of the sensed information. For this reason, Wireless Sensor Network (WSN) is attracting more and more research attention all over the world.

There are many restrictions on designing a WSN. First, sensor devices should be sufficiently small and cheap for being pervasively distributed into the background environment, which limits the complexity and computing capability of the device. As a result, only low-cost RF transceivers, inexpensive and small batteries, low-performance microcontrollers, little memory, and low-complexity algorithms can be used to meet the relatively high application requirements. This is the major challenge faced by the designers of WSNs.

So far, there are many projects addressing the WSN focusing on different aspects. Some examples are as follows.

- **PicoRadio** is a famous WSN project conducted by wireless research center of the University of California in Berkeley [4]. This project includes almost all areas of WSNs: RF circuit design, networking, positioning, low-voltage digital design, antenna design, low-power analog design, energy scavenging, etc.
- **WiseNET** (**Wi**reless **Se**nsor **NET**work) is a project from Swiss Center for Electronics and Microtechnology (CSEM) [5]. It focuses on ultra low-power design by combining a complex system-on-chip sensor device with a low-power MAC protocol called WiseMAC.
- **Sindrion** is a research project at Infineon Technologies [6]. It aims at integrating ultra low-power system-on-chip sensor and actuator devices into one of the most established middleware platforms for distributed semantic services, namely Universal Plug and Play (UPnP).

Since the work of this dissertation is a part of the Sindrion project, the concept of the Sindrion will be introduced in the next section.

## 2.3 Sindrion Basic Concept

So far, flexible application-layer support for autonomous device interaction is supported by high-tech computer appliances. One of the most established middleware platforms for this purpose is Universal Plug and Play (UPnP) [7]. It defines a set of universal, open, eXtensible Markup Language (XML) based protocols allowing the semantic description and control of various devices. Thereby, it supports the flexible co-operation of network nodes.

Unfortunately, the technical realization of the UPnP protocols is complex. Devices that support the entire UPnP protocol stack are thus power demanding and expensive. As a result, many small and cheap peripheral devices, e.g. small sensors and actuators, are excluded from UPnP networks.

To this end, a distributed system architecture called Sindrion has been proposed to feature an effective solution for integrating low-cost WSNs into UPnP networks [6]. The basic idea is to source out the complex UPnP operations from the wireless sensor devices, called Sindrion Transceivers, to dedicated network components, called terminals. The terminals

are assumed to have relatively loose cost and power restrictions. Fig. 2.1 shows the basic communication procedures between a Sindrion Transceiver and a terminal. The three steps as shown in the figure are explained as follows.

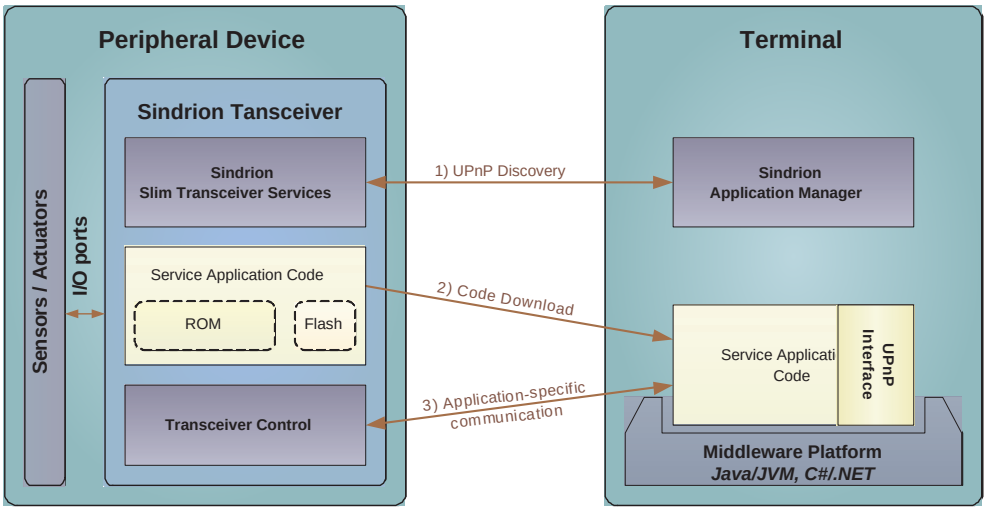


Figure 2.1: Sindrion System

1. **Discovery:** The two end devices find each other in the standard UPnP discovery phase [7].

2. **Code Download:** The services provided by the Sindrion Transceiver is described by service application code, which is stored in the transceiver's ROM. If the terminal does not yet contain this application code, it will be downloaded by the terminal.

3. **Application-Specific Communication:** The following communication between the downloaded service application on the terminal and its counterpart on the Sindrion Transceiver may be completely application-specific and does not have to be defined by any standard.

By this way, the Sindrion Transceiver is integrated into the UPnP network through the standard but relatively simple UPnP discovery protocol. Other complex UPnP protocols are sourced out to the terminal, which has high computational capability to act as a UPnP proxy for the Sindrion Transceiver. As a result, the Sindrion Transceiver can use low-performance components to minimize the cost.

In the Sindrion system, the terminal normally has loose energy restriction, e.g. a desktop is line-powered and a laptop or a PDA can be easily recharged. However, the Sindrion Transceiver is in many cases powered by cheap battery. Since changing or recharging the battery for a large number of Sindrion Transceivers is infeasible, power-saving is a very important design concern of the Sindrion Transceiver. Chp. 3 will discuss different possibilities to save power for a Sindrion Transceiver.

# Chapter 3

# Power Consumption Issue in WSNs

As aforementioned, power-saving is one of the most important design issues for the low-cost battery-powered wireless sensor devices. This chapter first gives an overview on possible power-saving methods for wireless sensor networks. Second, the reason why scheduling the duty cycle of the MAC layer is the most efficient way for the Sindrion system to save power is presented. Then, some related low-power MAC protocols are introduced, and their advantages and disadvantages are discussed. Finally, the possibilities to further improve the available schemes are figured out, which defines the goal of this dissertation.

## 3.1 Introduction of Power Saving Methods

In [4, 8, 9], many power-saving possibilities that could be adopted throughout the whole design of WSN devices are discussed. They can be divided into three major categories: low-power circuit design, fewer transmission, and more sleep.

### 3.1.1 Low-Power Circuit Design

Low power semiconductor and circuit design is a very important way to save power for a device, e.g. lowering the voltage of the digital circuits, optimizing the power amplifier in the RF analog circuit, minimizing the leakage of the devices, etc.

### 3.1.2 Fewer Transmission

The RF transceiver is one of the biggest power consumers in a wireless sensor device, so reducing the level of transmission power and the number of transmitted bits is supposed to be a very efficient way to save power.

**Lower Transmission Power:** The relation between the received power $P_r$ and transmission power $P_t$ is $P_r \sim \frac{P_t}{d^n}$, where $d$ is the distance and $n$ is an exponent ranging from $2$ to $4$. Therefore, adjusting the transmission power to a minimum level that just keeps the connectivity of the network can dramatically decrease the power consumption of the RF front end. Two typical examples of this way to save power are stated in [10, 11].

**Reduce Transmitted Bits:** The bits sent to the air include two parts: payload and overhead. Both of them can be reduced.

The payload can be dramatically shortened by data compression or by fusing the information from neighboring devices before a long distance transmission [12, 13]. However, it makes the system more complicated and requires more computation, which also consumes power. So a tradeoff is needed.

The overhead includes the header of data packets, the retransmitted data packets, and the signaling packets.

The header of the data packets could be shortened by using shorter addresses (e.g. encoded MAC address [14]) and header compression (e.g. TCP/IP header compression [15]).

To reduce the number of the retransmitted data packets, one typical way is using Error Correction Coding (ECC), which can recover disturbed packets so that packet loss rate is reduced. However, ECC increases the length of the packet on the other side, so compromise is needed for certain channel condition.

The other way to minimize the retransmission overhead is using proper Automatic Repeat reQuest (ARQ) protocols, e.g. data link layer ACK and transport layer TCP. However, ARQ also adds signalling packets causing additional overhead. So the tradeoff between the signaling and retransmission becomes the key issue. Some researches have been done on this topic, e.g. modified versions of TCP have been proposed to optimize the TCP performance in Mobile Ad hoc NETwork (MANET) and WSNs [16–19].

### 3.1.3 More Sleep

Switching the whole or part of the system to the sleep mode as much as possible could save power with a factor of tens or even hundreds. Normally, either high-layer application scheduling (e.g. [20–23]) or low-layer MAC scheduling (e.g. [24–26]) could determine when and which part of the system should be switched off. The disadvantage of this idea is that a sleeping system is apt to be inertia for a happening event, consequently the communication delay is increased. Therefore, the tradeoff between sleep and delay, i.e. power and delay, for a certain application becomes the key issue. To this end, the questions faced by a system designer are how to divide the system into multiple hardware power domains and how to arrange the sleeping schedules for these domains according to the requirements of the application.

## 3.2 Power Saving Methods for Sindrion

Among the above three power-saving methods, low-power circuit design is out of the scope of the Sindrion project. Nevertheless, further discussion is needed to determine what the key is to save power for the Sindrion Transceiver: *fewer transmission*, *more sleep*, or both. This decision should be based on both the software and hardware requirements of the Sindrion Transceiver.

The Sindrion system, as a kind of sensor/actuator network, is a low-rate low-traffic wireless network, because sensing and controlling occur quite rarely with intervals ranging from minutes up to months in most applications. For example, the temperature of a house might be measured each 10 minutes, but the water meter may be read out only once in several months.

In a low-traffic network, devices are mostly in idle state. As a result, most power is consumed by the idle state instead of data transmission. Therefore, using *fewer transmission* to minimize the power consumption of each data transmission is supposed to be not very effective, which will be proved later in Sec. 5.2.4.

On the other hand, enabling more components to sleep more during the idle state could have a significant effect on power-saving in the low-traffic case, which needs the support from a dedicated multiple power domain hardware structure and power management algorithms. To roughly estimate the power consumption of this method, we should first take a

look at the hardware components chosen for the prototype of the Sindrion Transceiver.

- **RF transceiver:** Data is transmitted and received by a commercial RF transceiver TDA5255E1 [27], which works in the 434MHz Industrial, Scientific, and Medical (ISM) band using Amplitude Shift Keying (ASK) or Frequency Shift Keying (FSK) modulation scheme. The data rate is less than 100 kbit/s.

- **FPGA:** For synchronization, modulation coding, and ECC, a Field Programmable Gate Array (FPGA) [28] is needed to process the data to and from the RF transceiver.

- **Microcontroller:** High-layer protocol processing is conducted in a single-chip high-performance 16-bit microcontroller with multiple on-chip A/D converters, timers, and memory modules [29]. It fits the requirements of the UPnP and TCP/UDP/IP protocol stack. Furthermore, it also supports idle and power down modes with flexible power management.

- **RTC:** A Real Time Clock (RTC) is needed by the UPnP discovery protocol for refreshing purpose. A flash microcontroller [30] is used, which provides more implementation flexibility than a normal RTC.

- **Memories:** Memory includes external RAM [31] and flash [32].

- **Others:** There are some additional components used, e.g. the wake-up logic of the RF transceiver, the FPGA, and the microcontroller.

The power consumption of these components is listed in Tab. 3.1.

Table 3.1: Power Consumption of the Components

| **Components** | **Work Power** | **Standby Power** |
|---|---:|---:|
| RF transceiver | Tx: 39.9 $mW$ / Rx: 27$mW$ | Sleep: 27 $\mu W$ |
| FPGA | Active: 20 $mW$ | Leakage: 1.5 $\mu W$ |
| Microcontroller | Active: 33 $mW$ | Sleep: 90 $\mu W$ |
| RTC | Working: 230 $\mu W$ | Idle: 75 $\mu W$ |
| Memories | Working: 87 $mW$ | Idle: 3.6 $\mu W$ |
| Others | - | Leakage: 15 $\mu W$ |
| Note: system voltage – 3 V, microcontroller clock frequency – 9 MHz. | | |

Among the above components, it is not difficult to switch the microcontroller and the FPGA to sleep or off mode when there is no traffic on the network. However, the RF transceiver should normally keep staying in receive mode because a packet could come at any time. So how to enable the RF transceiver to sleep during the idle state without severely affecting the data communication becomes the key question to be answered. This is normally solved by so-called MAC layer duty cycle scheduling, where the duty cycle denotes the percentage of the working time of a component. It is estimated in [4] that $1\%$ duty cycle for the RF transceiver can reduce the overall system power consumption by a factor of at least 50.

As shown in Tab. 3.1, the power consumption of the Sindrion Transceiver in a standby mode without duty cycle scheduling is

$$0.1851\,mW + 27\,mW = 27.1851\,mW. \tag{3.1}$$

The power consumption with a $1\%$ duty cycle of the RF transceiver is

$$0.1851\,mW + (27\,mW \times 0.01 + 0.027\,mW \times 0.99) = 0.48183\,mW. \tag{3.2}$$

So, power is saved by a factor of about $56$.

The above estimation shows that MAC layer duty cycle scheduling for the RF transceiver could save a huge amount of system power for the Sindrion Transceiver. Therefore, the MAC layer duty cycle scheduling is the key power-saving method for the Sindrion Transceiver.

Until now, several low-power MAC layer duty cycle scheduling schemes have been proposed. Some typical examples will be briefly presented in the next section.

## 3.3 Available Low-Power MAC Layer Duty Cycle Scheduling Schemes

So far, quite some researches have been done on low-power MAC layer duty cycle scheduling, e.g. [24–26, 33–36]. The IEEE802.15.4 MAC, the preamble sampling, and the WiseMAC are three mostly related work and will be briefly introduced in this section.

### 3.3.1 IEEE 802.15.4 LR-WPAN MAC Layer

The IEEE 802.15.4 Low-Rate Wireless Personal Area Network (WPAN) standard was first issued in October 2003 [33]. There is also an industrial alliance called Zigbee using this open global standard. The IEEE 802.15.4 standard specifies the Physical (PHY) layer and the MAC layer protocol to enable low-cost low-power short-range wireless devices to communicate. So, it could also be a choice for WSNs.

The IEEE 802.15.4 standard provides a MAC layer duty cycle scheduling scheme to save power in a master-slave star topology. A so-called superframe structure is used to define the basic arbitration scheme of the network, as shown in Fig. 3.1. The master of the star, which has relatively unlimited power and computing capability compared to the slaves, periodically broadcasts a beacon indicating the structure of the superframe, i.e. the Beacon Interval (BI), the superframe duration, and the number of slots for the Contention Access Period (CAP) and the Contention Free Period (CFP). The minimum beacon length is 26 bytes. The length of the beacon interval cannot be arbitrarily chosen, but is determined by

$$BI = \frac{16 \cdot 60 \cdot 2^{BO}}{SymbolRate}, \tag{3.3}$$

where the Beacon Order (BO) is an integer between $0$ and $14$ and SymbolRate is the symbol rate of the RF transceiver.

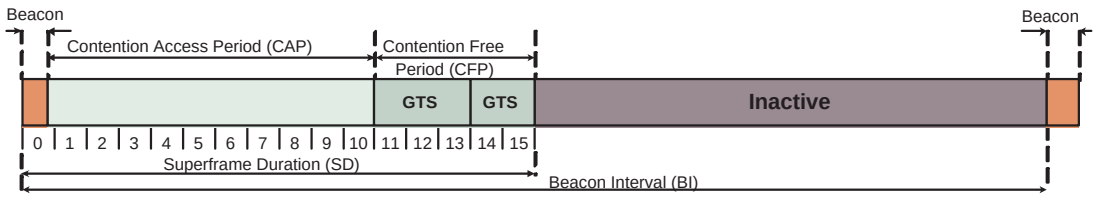


Figure 3.1: The IEEE 802.15.4 MAC Superframe Structure

The beacon also includes the information about to which slave(s) the master has packet(s) to transmit in the following period. A slave should periodically wake itself up to listen to the beacon. If the beacon indicates that there is a pending packet for the slave or if the slave has a packet for the master, the slave keeps awake for the communication in the CAP or CFP. If there is no packet, the slave goes back to sleep until the next beacon.

In the CAP, slaves use slotted Carrier Sense Multiple Access (CSMA) with exponential backoff to contend with each other for the channel. After getting the access right to the

channel, an uplink (slave to master) frame is directly sent to the master, then the master answers an acknowledgement (ACK). This is called a DATA-ACK session. To send a downlink (master to slave) frame, the slave should first send a data request command to the master so that the master knows the availability of the slave. The master then sends the downlink data frame to the slave and waits for an ACK from the slave. This is called a REQUEST-DATA-ACK session.

In the CFP, a Guaranteed Time Slot (GTS) can be assigned to a certain slave by the master in the beacon to support time critical streaming applications, e.g. continuous control signals from a mouse or a game panel. No CSMA is needed in GTSs.

To realize broadcast, the master puts the broadcast packet from the upper layer into the beacon as payload, so that all the slaves can receive it during listening to the beacon. However, because of the limited space of the payload in the beacon, up to only 52-byte upper-layer broadcast packets are supported.

The IEEE 802.15.4 MAC layer duty cycle scheduling scheme enables a slave to be awake for only the beacon length in each period when there is no traffic. The ratio between the beacon length and the beacon interval is the duty cycle. Since the beacon length is nearly a constant, the only way to reduce the duty cycle is to increase the beacon interval, which increases the response time on the other side. So a proper beacon interval should be chosen according to the power and delay requirements of the application.

Note that the superframe can only be used in a star topology with one master, because beacons from multiple masters could confuse a slave that can hear all of them. For other topologies, the IEEE 802.15.4 standard provides no power-saving solution.

### 3.3.2 Preamble Sampling

Preamble Sampling is another MAC layer duty cycle scheduling scheme proposed in [34]. The basic idea of this scheme is shown in Fig. 3.2.

The RF transceiver of a device periodically wakes itself up for a very short time, during which it detects whether the channel is occupied according to a Received Signal Strength Indicator (RSSI). If the channel is occupied, the RF transceiver wakes up other parts of the device to listen to the signal on the channel. Otherwise, it returns to sleep. This is the so-called sampling. To send a frame to such an RF transceiver that is sampling the channel, a transmitter has to first transmit a Wake-Up-Signal (WUS) with a length that is at least

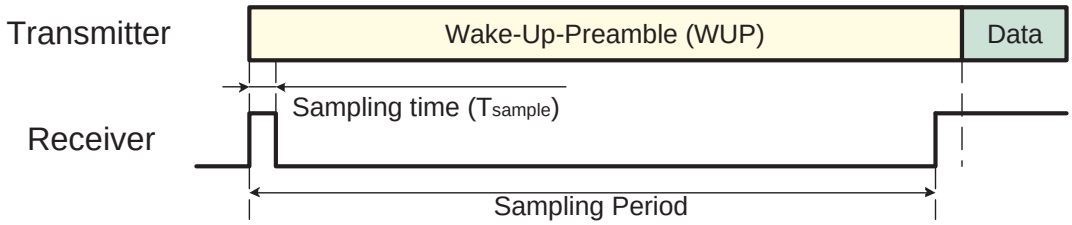


Figure 3.2: The Preamble Sampling Basic Idea

equal to the sampling period, so that the sampling RF transceiver is ensured to hear the WUS and wakes up the device to receive the data frame following the WUS. The WUS in [34] is a simple sync preamble, called Wake-Up-Preamble (WUP).

The preamble sampling scheme is more efficient in the sense of power-saving compared to the IEEE 802.15.4 MAC because of the following two reasons. First, the sampling time ($T_{sample}$) is only the receiving time of several bits. It is much shorter than the time required for hearing a beacon, so the duty cycle of the preamble sampling scheme is less than that of the IEEE 802.15.4 MAC if the sampling period is chosen equal to the beacon interval. Second, sampling the RSSI can be executed by the RF transceiver alone without the aid from other modules, while the IEEE 802.15.4 MAC needs a MAC module to parse the beacon. So the power consumption during the time for sampling the preamble is also less than that for listening to the beacon in the IEEE 802.15.4 MAC. Another advantage of the preamble sampling scheme is that it could be also deployed into a multiple-master topology or even a peer-to-peer topology.

One drawback of the preamble sampling scheme is its low efficiency of channel capacity because of the transmission of the long WUS. The length of the WUS is equal to the sampling period, which is ideally adapted to the response time of the application – in the order of hundreds of milliseconds or even longer. However, the data frame length in WSNs is normally quite short and takes only tens of milliseconds to transmit. As a result, the WUS become the major traffic on the channel instead of normal frames, which dramatically reduces the channel capacity for the normal frames. That is the reason why the preamble sampling scheme is only well-suited for networks with sporadic traffic where the channel capacity is not a critical factor.

Another disadvantage of the preamble sampling scheme is the high transmission energy consumed for sending the long WUS. So it is normally used in a master-slave topology,

in which the energy constraint of the master is loose so that first, the transmission energy of the long WUS is not a heavy burden any more; second, the master can keep in receive mode and the slave needs no WUS before sending a data packet to the master.

The other negative effect caused by the long WUS is overhearing. Since the WUS has the length of the sampling period, all devices in the transmission range are woken up. Those devices that hear the WUS but are not the destination of the packet are called overhearers. They also have to keep awake until they receive the destination address of the data frame. This overhearing effect drastically decreases the power efficiency achieved by the short-period low-power RSSI sampling.

### 3.3.3 WiseMAC

To reduce the disadvantages caused by the long WUS, the so-called WiseMAC (**Wi**reless **Se**nsor **MAC**) based on the preamble sampling scheme has been proposed to shorten the length of the WUS [35–38].

The underlying idea is that if a device knows the sampling schedules of its neighbors, it can send a very short WUS to wake up the destination device without inducing many overhearers. However, to know the schedule, synchronization among neighbors is necessary.

The WiseMAC achieves a loose synchronization by storing a neighbor's sampling schedule that is piggybacked on the last received ACK from the neighbor. So a very short WUS can be used for the next frame according to the stored schedule. However, the instability of the cheap quartz in small devices causes quite large clock drift, so that the length of the WUS must be prolonged again to compensate for the clock drift, especially if the schedule has been received a long time ago.

Assuming the frequency tolerance $\theta$ of the quartz is around $30$ ppm, and the time elapsed since the last received schedule is $L$, the transmitter should start to send a $(4\theta L + T_{sample})$-long WUS $2\theta L$ ahead of the stored schedule to ensure to wake up the destination device, as shown in Fig. 3.3. Therefore, the higher is the traffic, the smaller is the $L$, so the shorter is the WUS. As a result, overhearing and channel occupation are dramatically reduced, so that the WiseMAC also works well in relatively high-traffic networks.

The performance of the WiseMAC protocol in a master-slave star topology has been evaluated in [35]. It consumes around half of the power as the IEEE 802.15.4 MAC in most cases. In [36], the WiseMAC protocol has also been compared to two other famous sensor

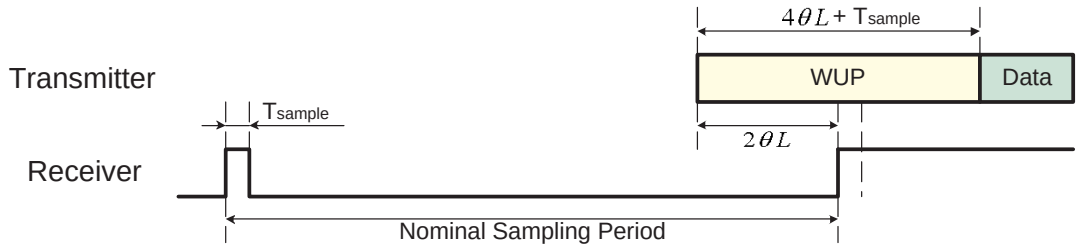


Figure 3.3: The WiseMAC Basic Idea

MAC protocols, S-MAC [25] and T-MAC [26], in a peer-to-peer topology with multihop routing. The WiseMAC shows again a better result in the sense of power consumption.

Nevertheless, there are still many long WUS in the WiseMAC because of the clock drift compensation. In this case, the inventors of the WiseMAC have suggested to repeat the data frame in the WUS instead of using the meaningless WUP, as shown in Fig. 3.4. The data repetition scheme, which will be called the Rep scheme later, can dramatically decrease the negative effect caused by overhearing, because each overhearer can receive a complete data frame after being woken up so that it knows it is an overhearer and goes back to sleep immediately. The average waiting time for an overhear is $1.5$ times of the data frame transmission time.

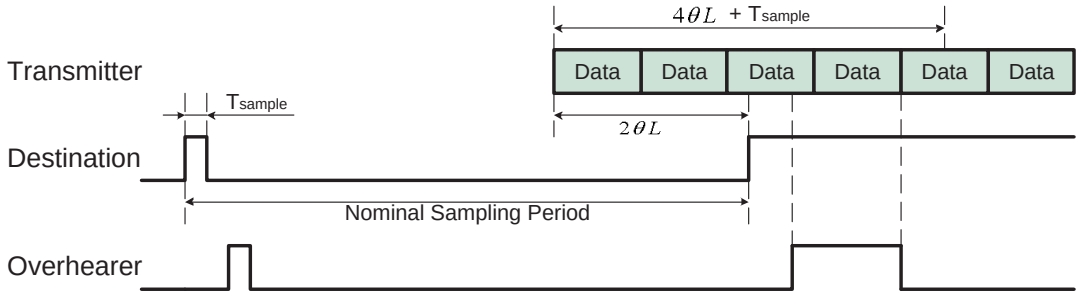


Figure 3.4: The Data Repetition Basic Idea

## 3.4 Possibilities for Further Improvement

As described above, the IEEE 802.15.4 MAC is an open global standard. It synchronizes all the devices by a combination of Time Division Multiple Access (TDMA) and random

access. However, it only offers a power-saving mechanism for master-slave star topology, which highly limits its range of usage.

In the preamble sampling scheme, the RSSI sampling allows the receiver to work more power efficient for shorter time during the on-duty period compared to the beacon listening in the IEEE 802.15.4 MAC. It gives potential to achieve a much more power-efficient scheme. The other big advantage of the preamble sampling is that it could be used in any network topology.

However, the preamble sampling scheme is totally asynchronous, so that a very long WUS is needed, which not only is channel inefficient, but also generates too many overhearers wasting a large amount of energy. To this end, the WiseMAC protocol has been proposed to shorten the length of the WUS by loosely synchronizing the neighboring devices, which improves the preamble sampling scheme.

Nonetheless, long WUSs are still needed in many cases. Firstly, the very beginning packets between the neighbors should be sent with a long WUS, which occurs frequently for a mobile device. Secondly, the clock drift compensation requires a longer WUS that could be up to the length of the sampling period for a low-traffic case. Thirdly, a broadcast packet must always use a long WUS to wake up all the neighbors. Besides application layer broadcast and multicast traffic, broadcast packets are also sent frequently in networks with mobile devices during network discovery, handshaking, multihop routing, etc.

To this end, the Rep scheme has been proposed to reduce the negative effect caused by the long WUS. Although data repetition can improve the WiseMAC, it is not the best choice in case of long WUS because of the following reasons:

- Repeating data is not efficient enough for long data frames, because each overhear has to wait $1.5$ times of the data frame transmission time on the average.
- The destination of a unicast frame still has to wait until the end of the last repetition of the data frame to answer the ACK, wasting a lot of energy.

The Sindrion system has quite a number of long packets because of the UPnP and TCP/UDP/IP protocol stack. The `advertisement` and `search` messages in UPnP are at least 100 bytes in length [7], and the application control packet is from 2 to 500 bytes in length (mostly around 20 bytes). Adding the TCP/UDP/IP header, the MAC header, and the PHY overhead such as ECC and modulation coding, the overall length of a data

frame is longer than 100 bytes and even up to several hundreds bytes, which is very long for WSNs.

Therefore, for the Sindrion system, there is still space for further improving the preamble sampling scheme by optimizing the case of long data frames.

# Chapter 4

# Ultra Low Power MAC Layer Wake-Up-Frame Scheme

As discussed in Chp. 3, MAC layer duty cycle scheduling combined with a dedicated hardware power domain structure is the key method to realize an ultra low-power Sindrion Transceiver. In this chapter, the so-called Wake-Up-Frame (WUF) scheme is proposed to further improve the preamble sampling scheme with the help of an elaborate multiple power domain hardware architecture. Furthermore, the combination of the WUF scheme and the WiseMAC is also discussed.

Note that to make the description clearer, the basic preamble sampling scheme using the WUP in [34] is called the WUP scheme in the rest of the dissertation, the term *preamble sampling* denotes only the general concept that a receiver periodically samples the channel, and the *preamble sampling scheme* includes all the schemes that use the *preamble sampling* technology.

## 4.1 Multiple Power Domain Hardware Architecture

Fig. 4.1 shows the basic hardware structure of the Sindrion Transceiver prototype. It comprises four major parts: an RF transceiver, an FPGA, a microcontroller, and an RTC. Note that external memory and sensor/actuator are not included in this figure for simplicity.

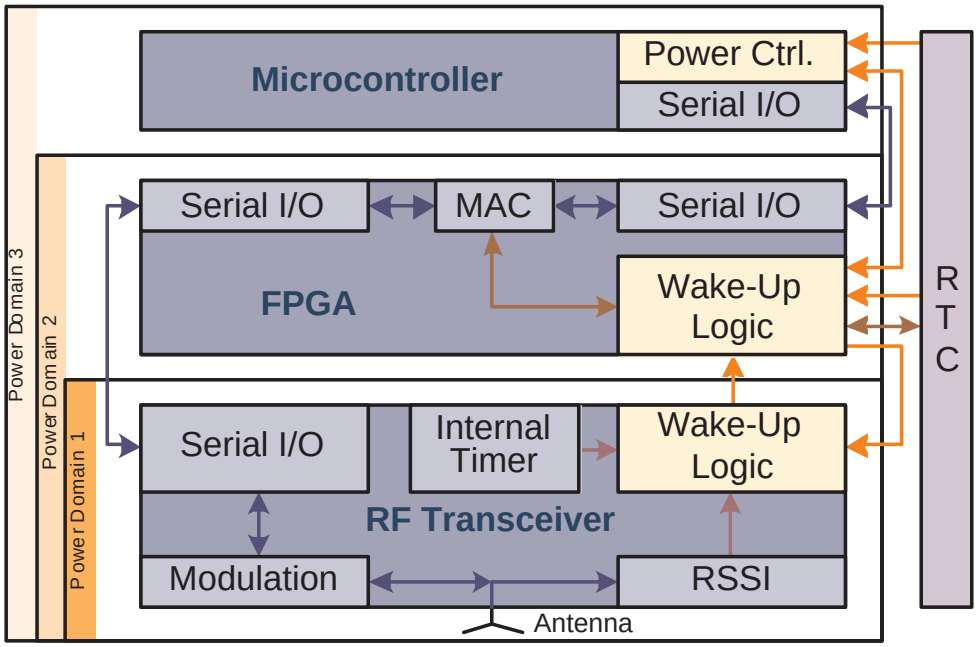


Figure 4.1: The Sindrion Hardware Architecture

The device is divided into three power domains as follows.

1. **RF Transceiver:** When there is no traffic on the network, only the RF transceiver periodically wakes itself up to sample the channel.

   The TDA 5255E1 [27] is chosen as the RF transceiver, because it provides a so-called self-polling mode, which facilitates the preamble sampling scheme by default. In the self-polling mode, the RF transceiver is periodically woken up by an internal timer. Upon sampling, the RF transceiver detects the level of the RSSI and the data rate of the demodulated signal. If the RSSI exceeds a programmable threshold and the data rate matches a predetermined value, a pulse is generated on an output pin, which could be used to wake up other components of the device. This Data Rate Detection (DRD) function is completed within a very short time period, typcially the length of 3 Manchester-coded bits, i.e. 6 ASK/FSK modulated symbols.

   Compared to the traditional preamble sampling based simply on the RSSI, this DRD function could significantly improve the system power efficiency by decreasing the unnecessary wake-up. If only the RSSI is used as the sampling criteria, strong noise or interference from other systems using the same frequency band could also wake

up the device. This is especially important for a wireless network operating in the ISM-band, where typically a number of systems share the same frequency band generating a lot of interference. Moreover, different data rates can be used for the WUS and other normal frames, respectively, so that data frames and ACKs cannot wake up overhearers because of the DRD, which further reduces the probability of unnecessary wake-up. To further prove the advantage of using DRD, quantitative analysis is conducted in Sec. 5.2.4.

2. **FPGA:** When the RF transceiver detects a signal with enough strength and valid data rate, the FPGA [28] is woken up to receive the data frame. If the frame is addressed to this device, the FPGA wakes up the microcontroller.

3. **Microcontroller:** After being woken up by the FPGA, the microcontroller [29] processes the received data frame accordingly.

Besides, an RTC [30] is always running for the need of UPnP discovery. This RTC will be also utilized by the WUF scheme, which will be introduced later.

By combining this delicate power domain structure, the MAC layer duty cycle scheduling, and other power management facilities of the microcontroller, different components can be woken up only on demand to save power.

## 4.2 The Wake-Up-Frame Scheme Description

### 4.2.1 The Wake-Up-Preamble Scheme

Before introducing the WUF scheme, it is necessary to first show how the WUP scheme works on the aforementioned hardware architecture (See Fig. 4.2).

In the figure, a low level of the line means the component is in sleep state, a high level in working state (receiving, transmitting, idling, or calculating), and a slope in setup state. The meanings of the shown parameters are: $T_{sample}$ and $T_{cycle}$ are the times for sampling and cycle period (i.e. the sampling period in Sec. 3.3.2), respectively; $T_{s\_rf}$, $T_{s\_fpga}$, and $T_{s\_\mu c}$ denoted by slopes are the setup times for the RF transceiver, the FPGA, and the microcontroller; $T_{data}$ and $T_{ack}$ are the times needed to transmit a data frame and an ACK frame; $T_{to\_addr}$ is the time to transmit a data frame until the destination MAC address; $T_t$

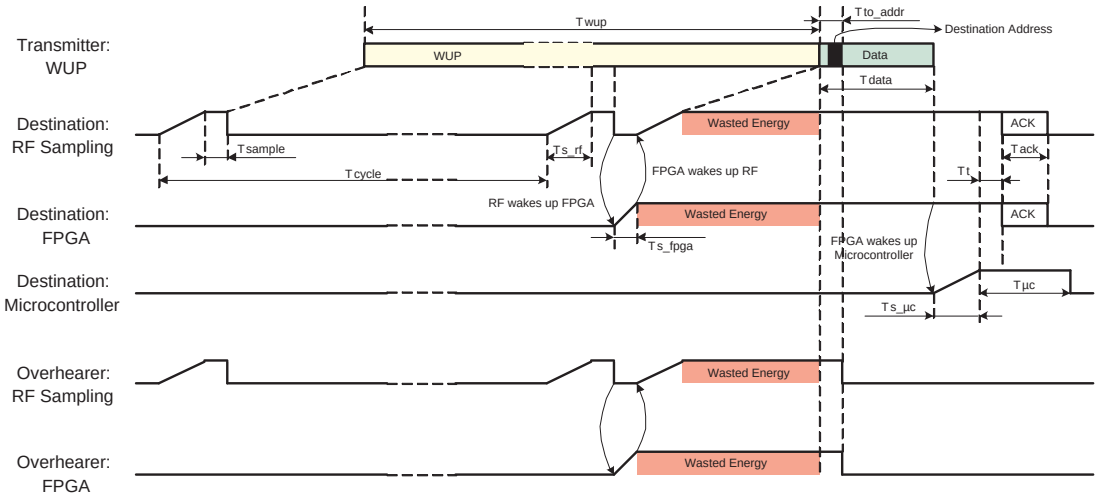


Figure 4.2: The Wake-Up-Preamble Scheme

is the transition time between transmit and receive modes of the RF transceiver; $T_{\mu c}$ is the time needed for the microcontroller to process the data frame.

The operation of the transmitter, the destination, and the overhearer in the WUP scheme are as follows:

- **Transmitter:** For calculating the length of the WUP ($T_{wup}$), the worst case should be taken into consideration. I.e. the beginning of the WUP just misses the sampling period of the destination, and the setup time of the RF transceiver and of the FPGA should also be included.

- **Destination:** When the destination RF transceiver detects the WUP, it wakes up the FPGA, and then the FPGA sets the RF transceiver into normal receive mode. Afterwards, both the RF transceiver and the FPGA have to wait for the data frame. After the data frame has been received by the FPGA, the microcontroller is woken up to answer the ACK and process the data. After sending the ACK, the RF transceiver and the FPGA go to sleep.

- **Overhearer:** The RF transceiver and the FPGA are woken up in the same way as in the destination case. However, when the FPGA receives the destination MAC address of the data frame, it knows that it is an overhearer, and then it switches off

the RF transceiver and itself. Note that it is not necessary to wait until the end of the data frame, but just until the destination MAC address.

The negative effect of the long WUP is obvious. Both the destination and the overhearer have nothing to do but wait during the shadowed areas as shown in Fig. 4.2. The length of this unnecessary waiting time is approximately $T_{cycle}/2$ on the average, which is in the order of hundreds millisecond or even longer, wasting a large amount of energy.

### 4.2.2 The Wake-Up-Frame Scheme

To eliminate the unnecessary waiting time in the WUP scheme, we propose to send a wake-up-frame (WUF) instead of the meaningless WUP. The WUF comprises multiple Short Wake-Up-Frames (SWUFs) as shown in Fig. 4.3. Each SWUF is a complete but very short MAC frame containing the destination MAC address, a special position field, and a CRC checksum. The position field is a successively decreasing integer indicating the position of the SWUF in the whole WUF, as shown by the number $n$ down to 1 in the figure.

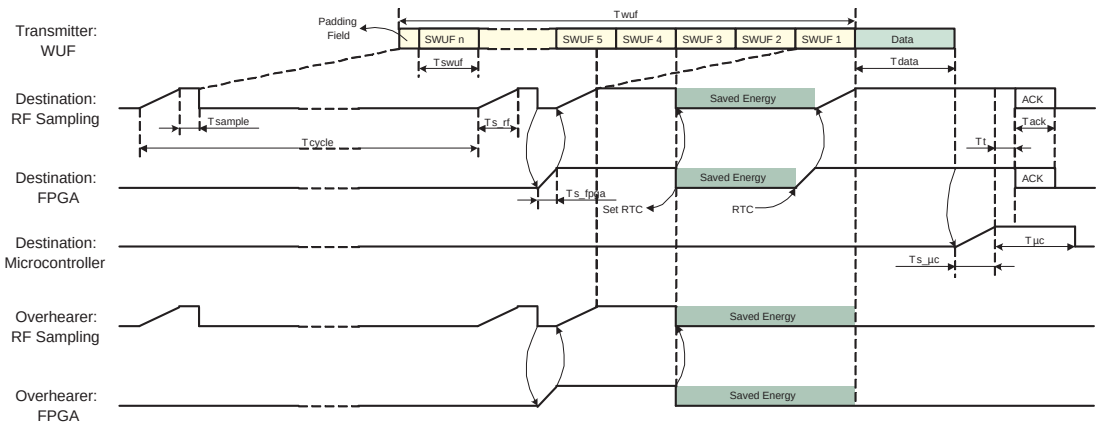


Figure 4.3: The Wake-Up-Frame Scheme

The operation of the transmitter, the destination, and the overhearer in the WUF scheme are as follows:

- **Transmitter:** The length of the SWUF ($T_{swuf}$) is a constant because all frame fields are fixed. For calculating the length of the WUF ($T_{wuf}$), the worst case should be

taken into consideration. I.e. the beginning of the WUF just misses the sampling period of the destination, and the setup time of the RF transceiver ($T_{s\_rf}$) and the FPGA ($T_{s\_fpga}$) should also be included. Furthermore, an additional SWUF should be added to ensure that the receiver can hear at least one complete SWUF after being woken up. Note that $(T_{cycle} + T_{s\_fpga} + T_{s\_rf})$ normally cannot be exactly divided by the $T_{swuf}$, so an extra padding field is used at the beginning of the WUF, as shown in the Fig. 4.3.

- **Destination:** When the destination RF transceiver detects the WUF at any place, it wakes up the FPGA, and then the FPGA sets the RF transceiver into normal receive mode. Afterwards, the FPGA can receive a complete SWUF, e.g. SWUF number 4 in the figure. So it knows that it is the destination according to the destination MAC address in the SWUF. Then it calculates the remaining time before the data frame comes based on the position field. The calculation is simple because the length and the data rate of the SWUF are constants and known by each device. If the calculated result shows that the remaining time is still long enough, the FPGA switches off the RF transceiver, sets a timer in the RTC, and goes to sleep. Just before the data frame comes, the RTC wakes up the FPGA, then the FPGA wakes up the RF transceiver, so that they can receive the data frame. When the data frame is correctly received, the microcontroller is woken up to answer the ACK and process the data.

- **Overhearer:** The RF transceiver and the FPGA are woken up in the same way as in the destination case. However, when the FPGA receives an SWUF, it knows that it is an overhearer so that it can switch off the RF transceiver and itself immediately.

Note that, unlike in the WUP scheme, the FPGA of both the destination and the overhearer must receive a complete SWUF with correct Cyclic Redundancy Check (CRC) to ensure that the decoded the destination MAC address is correct. If the CRC check is wrong, the FPGA should wait for another SWUF.

By sending multiple SWUFs, the RF transceiver and the FPGA of both the destination and the overhearer can go to sleep during the unnecessary waiting time in the WUP scheme, as shown by the shadowed areas in Fig. 4.3. The on time of the RF transceiver and the FPGA are limited to around $1.5T_{swuf}$ on the average. Since the SWUF is very short containing only the basic MAC header, the destination MAC address, and a one-byte position field, it takes only several milliseconds for transmitting. Therefore, the WUF scheme

minimizes the unnecessary waiting time from hundreds of milliseconds down to several milliseconds for each data frame. This drastically reduces the power consumption of both the destination and the overhearer.

## 4.3 The Wake-Up-Frame Scheme Analysis

This section qualitatively analyzes the gain and cost of the WUF scheme compared to the IEEE802.15.4 MAC and the WUP scheme based on the specially designed hardware architecture.

- **IEEE 802.15.4 MAC:** As aforementioned in Sec. 3.3.1, the IEEE 802.15.4 MAC is a global open standard supporting low-rate time critical streaming applications. However, it provides power-saving mechanism only for a star topology with one master. Moreover, the duty cycle scheduling using the periodical beacon generates relatively high overhead, which not only consumes the power but also occupies the channel. This overhead should not be so significant for WPANs with high traffic. However, it could be a heavy burden for low-traffic WSNs. Another point is that the programming effort and the code size for the IEEE 802.15.4 MAC are relatively high, because supporting a combination of TDMA and CSMA random access is not a so simple task, especially for a low-cost slave device.

- **The WUP Scheme:** The WUP scheme based on the preamble sampling technology uses short-period and low-power DRD sampling instead of listening to the beacon in the IEEE 802.15.4 MAC. It not only dramatically decreases the power consumption, but also enables the protocol to work in any topology. The drawback of the WUP scheme is that the very long WUP for each data frame generates a long unnecessary waiting time for the destination as well as the overhears. Furthermore, it occupies a lot of channel so that it is not well suited to work in a high-traffic network. However, when the traffic is low, the long WUP could occupy less channel than the periodical beacon in the IEEE 802.15.4 MAC. Concerning the programming complexity, the WUP scheme using CSMA with exponential backoff is definitely simpler than the IEEE 802.15.4 MAC.

- **The WUF Scheme:** The WUF scheme drastically decreases the average unnecessary waiting time for both the destination and the overhearer from $0.5T_{cycle}$ down

to $1.5T_{swuf}$. However, the WUF has basically the same length as the WUP, so it still works only for low-traffic networks. The complexity overhead introduced by the WUF scheme is not significant. Firstly, the FPGA decodes the MAC address of the SWUF. However, address decoding is nevertheless needed for data frame, so the same block can be reused. Secondly, the FPGA calculates the remaining time before the data frame according to the received position field. This is just a single multiplication of two integers: the position field and a constant denoting the transmitting time of an SWUF. Thirdly, the FPGA needs a hardware connection and a software routine for setting the RTC, which is also not a big issue. All in all, the additional effort introduced by the WUF scheme is minor compared to the significant amount of power it could save.

## 4.4 Wake-Up-Frame and WiseMAC Combination

The WUF scheme saves power by minimizing the unnecessary waiting time. But it cannot solve the low channel capacity problem caused by the long WUS. However, as mentioned in Sec. 3.3.3, the WiseMAC shortens the length of the WUS to reduce the channel occupation as well as the power consumption. Nevertheless, long WUS still exists widely because of the clock drift, the mobility of devices, and the broadcast and multicast traffic. The WiseMAC has also proposed to repeat the data frame instead of the meaningless WUP to further reduce the power consumption in case of a long WUS. However, the Rep scheme has an average unnecessary waiting time of around $1.5T_{data}$ for overhearers. It is much longer than the $1.5T_{swuf}$ in the WUF scheme, especially for the quite long Sindrion data frame as mentioned in Sec. 3.4.

The WUF scheme and the WiseMAC optimize the preamble sampling technology via different methods and they are complementary to each other. Therefore, a combination of them is supposed to generate a further optimized solution based on the preamble sampling technology. The way of combination is just transmitting multiple SWUFs instead of the meaningless WUP in the WiseMAC protocol when the WUS is long.

To estimate the performance of the proposed WUF scheme and the idea of combining the WUF scheme and the WiseMAC, analytical computations will be conducted in Chp. 5 to compare them with the related work in a master-slave star topology.

# Chapter 5

# Analytical Computation

In this chapter, analytical computation is conducted to evaluate the performance of the proposed WUF scheme and the combination of the WUF scheme and the WiseMAC. For simplicity, the analytical computation is based on a master-slave star topology with only downlink traffic that obeys the Poisson process traffic model (See Sec. 6.2.5 for the detailed explanation of the traffic model). The power consumption of the slave, response time, and channel occupation of the following schemes are computed for both the unicast and broadcast traffic.

- **The WUP scheme:** The original preamble sampling scheme using meaningless WUP.
- **The Rep scheme:** It repeats the data frame instead of the meaningless WUP.
- **The WUF scheme:** The WUF is used instead of the meaningless WUP. This is the scheme proposed in Chp. 4.
- **The WiseMAC+WUP scheme:** The basis of the WiseMAC using shortened meaningless WUP.
- **The WiseMAC+Rep scheme:** Repetition of the data frame is used instead of the meaningless WUP, if the WUS in the WiseMAC is long.
- **The WiseMAC+WUF scheme:** The WUF instead of the meaningless WUP is used, if the WUS in the WiseMAC is long. This is the combination scheme proposed in Chp. 4.

- **Ideal Preamble Sampling:** It is the ideal case for the preamble sampling technology. I.e. the transmitter always knows the exact schedule of the destination so that a very short WUP can be used, i.e. the length of the sampling time. It is used as a reference for the ideal case.
- **The IEEE 802.15.4 MAC:** The IEEE 802.15.4 MAC using superframe and beacon.

## 5.1 Computation Principle

The detailed computation procedures are depicted in Appendix A. This chapter will only use the WiseMAC+WUP as an example to explain the principle of the computation. The meanings and values of the used parameters are listed in Appendix B.

The interarrival interval of a Poisson process traffic model is exponentially distributed with a probability density function

$$p(t) = \lambda e^{-\lambda t}, \tag{5.1}$$

and a cumulative distribution function

$$f(t) = 1 - e^{-\lambda t}. \tag{5.2}$$

As mentioned in Sec. 3.3.3, under a certain threshold of the interval, the WiseMAC enables a shortened WUS, which is denoted by

$$T_{th} = \frac{T_{cycle}}{4\theta}, \tag{5.3}$$

where $\theta$ is the clock drift of the device.

The probability to generate an interval that is less than $T_{th}$ (case 1) is

$$p_1 = p(t < T_{th}) = f(T_{th}) = 1 - e^{-\lambda T_{th}}, \tag{5.4}$$

and that is larger than $T_{th}$ (case 2) is

$$p_2 = p(t \geq T_{th}) = 1 - f(T_{th}) = e^{-\lambda T_{th}}. \tag{5.5}$$

**Case 1:** $interval < T_{th}$

As shown in Fig. 5.1, when the interval is less than $T_{th}$, the length of the shortened WUP is

$$T_{pre1} = 4\theta T_{itvl} + T_{sample} + T_{s\_fpga} + T_{s\_rf}, \tag{5.6}$$

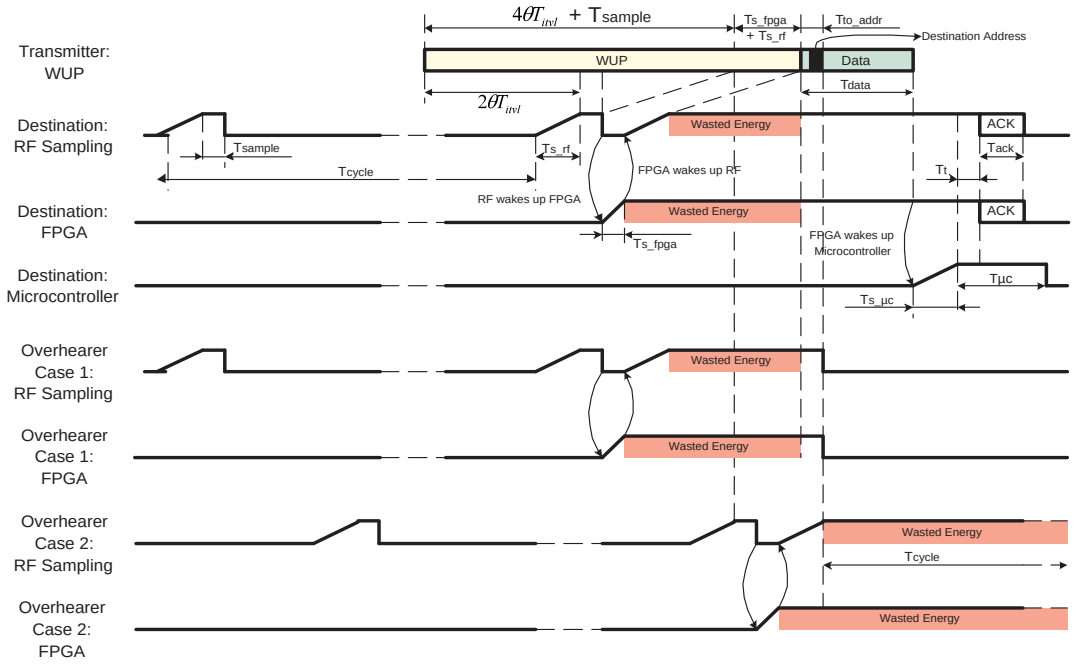


Figure 5.1: The WiseMAC+WUP Detail

where $T_{itvl}$ is the Poisson interarrival interval. The average power consumption of receiving one packet for the destination device is

$$\begin{aligned}\overline{P_{dst1}} &= \hat{P}_{rf\_rx}(2\theta T_{itvl} + T_{s\_fpga} + T_{s\_rf} - T_{s\_fpga} + T_{data} + T_{s\_\mu c}) \\ &\quad +\hat{P}_{rf\_tx}(T_t + T_{ack}) \\ &\quad +\hat{P}_{fpga}(2\theta T_{itvl} + T_{s\_fpga} + T_{s\_rf} + T_{data} + T_{s\_\mu c} + T_t + T_{ack}) \\ &\quad +\hat{P}_{\mu c}(T_{s\_\mu c} + T_{\mu c}) \\ &\doteq AT_{itvl} + B. \end{aligned} \tag{5.7}$$

Note that the power consumption of a component during setup time is assumed to be the normal power consumption of the component, e.g. RF receiving power is used for RF setup.

For an overhearer, there are two cases. In the first case, the overhearer is woken up by the first $(4\theta T_{itvl} + T_{sample})$ part of the $T_{pre1}$, so the overhearer can receive the following data frame and make decision according to the destination address. The probability of this case is

$$p_{ovh1\_1} = \frac{4\theta T_{itvl} + T_{sample}}{T_{pre1}}, \tag{5.8}$$

and the average power consumption in this case is

$$\begin{aligned}\overline{P_{ovh1\_1}} &= \hat{P}_{rf\_rx}(2\theta T_{itvl} + T_{s\_fpga} + T_{s\_rf} - T_{s\_fpga} + T_{to\_addr}) \\ &\quad +\hat{P}_{fpga}(2\theta T_{itvl} + T_{s\_fpga} + T_{s\_rf} + T_{to\_addr}) \\ &\doteq AT_{itvl} + C. \end{aligned} \tag{5.9}$$

In the second case, the overhearer is woken up by the last $(T_{s\_fpga} + T_{s\_rf})$ part of the $T_{pre1}$, so the overhearer has no time to setup to receive the following data frame. As a result, it has to wait one $T_{cycle}$ if nothing is received after being woken up, because overhearers do not know the length of the WUP. The probability of this case is

$$p_{ovh1\_2} = \frac{T_{s\_fpga} + T_{s\_rf}}{T_{pre1}}, \tag{5.10}$$

and the power consumption in this case is

$$P_{ovh1\_2} = \hat{P}_{rf\_rx}(T_{s\_rf} + T_{cycle}) + \hat{P}_{fpga}(T_{s\_fpga} + T_{s\_rf} + T_{cycle}). \tag{5.11}$$

So the average power consumption of receiving one packet for an overhearer is

$$\overline{P_{ovh1}} = \overline{P_{ovh1\_1}} \cdot p_{ovh1\_1} + P_{ovh1\_2} \cdot p_{ovh1\_2}. \tag{5.12}$$

Since the preamble is shortened, only part of the slaves are woken up to become overhearers. The probability that a slave is woken up is $\frac{T_{pre1}}{T_{cycle}}$. So the power consumption of receiving one packet for each slave in case 1 ($T_{itvl} < T_{th}$) is

$$\begin{aligned} \overline{P'_{pkt1}} &= \overline{P_{dst1}}\frac{1}{N} + \overline{P_{ovh1}}\frac{T_{pre1}}{T_{cycle}}\frac{N-1}{N} \\ &\doteq FT_{itvl}^2 + GT_{itvl} + H, \end{aligned} \tag{5.13}$$

where $N$ is the number of slaves, F, G and H are intermediate variables that are not functions of $T_{itvl}$. The average power consumption of receiving one packet for each slave is the expectation of $\overline{P'_{pkt1}}$

$$\begin{aligned} \overline{P_{pkt1}} &= E\left(\overline{P'_{pkt1}}\right) \\ &= \int_0^{T_{th}} (Ft^2 + Gt + H)\frac{p(t)}{p_1}dt \\ &= \int_0^{T_{th}} (Ft^2 + Gt + H)\frac{\lambda}{p_1}e^{-\lambda t}dt \\ &= 2F\frac{1}{\lambda^2} + \left(G - 2FT_{th}\frac{p_2}{p_1}\right)\frac{1}{\lambda} + \left(H - GT_{th}\frac{p_2}{p_1} - FT_{th}^2\frac{p_2}{p_1}\right). \end{aligned} \tag{5.14}$$

**Case 2:** $interval \geq T_{th}$

When the interval is greater than $T_{th}$, the length used for the WUP is the same as for the normal WUP scheme. Assume that the average power consumption of receiving one packet for each slave in case 2 is $\overline{P_{pkt2}}$.

**Overall**

All in all, the average power consumption of receiving one unicast packet for each slave in the WiseMAC+WUP is

$$\overline{P_{pkt_{WiseMAC+WUP\_U}}} = \overline{P_{pkt1}} \cdot p_1 + \overline{P_{pkt2}} \cdot p_2. \tag{5.15}$$

If the master sends unicast packets to each slave with a Poisson arrival rate of $\lambda$, the power consumption of each slave device is

$$P_{WiseMAC+WUP\_U} = \overline{P_{pkt_{WiseMAC+WUP\_U}}} \cdot \lambda N + P_{no\_traffic}, \tag{5.16}$$

where $P_{no\_traffic}$ is the power consumption of the device in no traffic case.

This is the basic principle for computing the power consumption of the WiseMAC+WUP scheme with downlink unicast traffic. Please see Appendix A for the detailed computation procedures of all the schemes.

## 5.2 Computation Results

The analytical computations of the different schemes have been evaluated using C program. Now the computational results will be presented and discussed.

### 5.2.1 Power Consumption and Battery Lifetime

#### No Traffic

The power consumption of the IEEE 802.15.4 MAC and the preamble sampling in the no traffic case is shown in Fig. 5.2. The preamble sampling denotes all the schemes based on the preamble sampling technology, because they have the same results when there is no traffic. Note that the cycle period of the IEEE 802.15.4 MAC cannot be chosen arbitrarily, but is calculated based on Equ. 3.3. With a BO from 3 to 8, the following cycle periods can be derived: 76.8 ms, 153.6 ms, 307.2 ms, 614.4 ms, 1228.8 ms, 2457.6 ms. As shown in Fig. 5.2, the preamble sampling consumes much less power than the IEEE 802.15.4 MAC, especially when the cycle period is short.

Which cycle period should be used is a very important issue, which is determined by the response time required by the application. The Sindrion system could be used for real time control purpose, e.g. switching on/off a light, so the response time should be shorter than one second. Therefore, a cycle period of 614.4ms is supposed to be a good choice, which will be used for the later computational results.

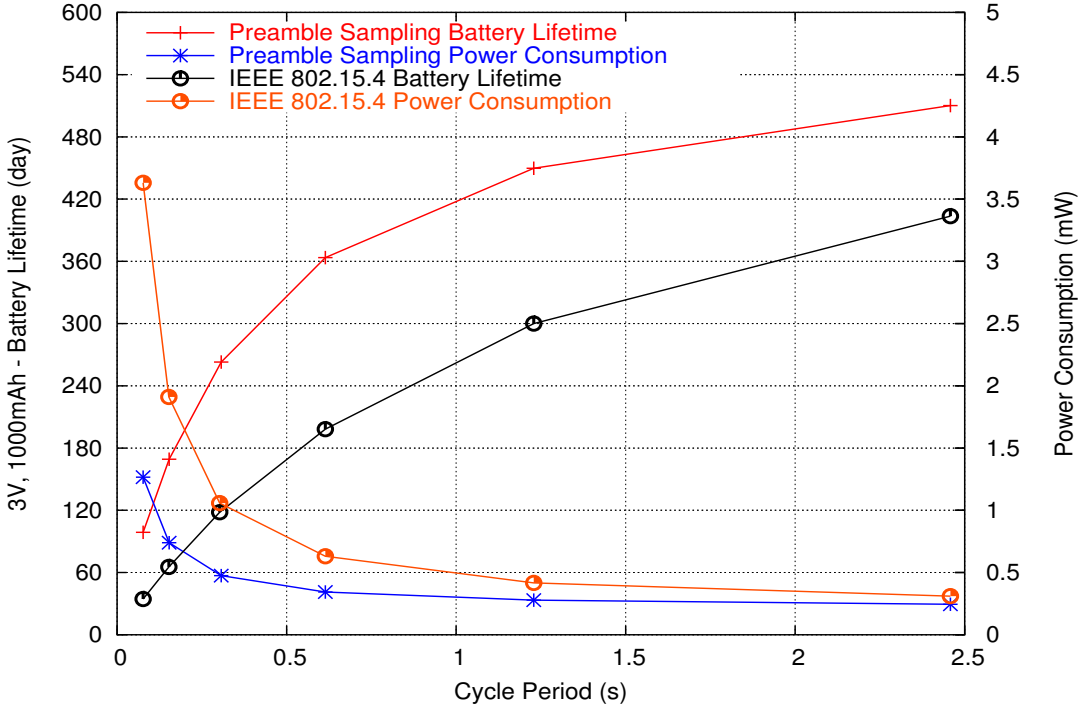


Figure 5.2: Battery Lifetime of Preamble Sampling and IEEE 802.15.4 MAC (No Traffic, Analytical)

If a 3V battery with 1000 mAh is assumed, the battery lifetime of the preamble sampling with a cycle period of 614.4 ms can reach more than one year, which is about twice of that of the IEEE 802.15.4 MAC. Note that the self-discharging effect of the battery is not considered here.

This no traffic case shows the advantage of using short-period and low-power DRD sampling compared to the IEEE 802.15.4 MAC beacon listening, which is exactly the reason why preamble sampling technology is potentially more power-efficient than the IEEE 802.15.4 MAC.

### Unicast

The battery lifetime of all the schemes in a star topology with 10 slaves and only downlink traffic is shown in Fig. 5.3. The cycle period of the preamble sampling and the beacon interval of the IEEE 802.15.4 MAC are both 614.4 ms. Packets from the master to each

slave are modelled by the Poisson process with an arrival rate of $\lambda$ indicated on the x-axis. Thus, the overall packet arrival rate in the network is $10 \cdot \lambda$. Note that only up to 1 packet/s overall traffic is considered, because the channel is almost saturated by the long WUS of the WUP scheme and the WUF scheme in such a traffic load. For the WiseMAC and the IEEE 802.15.4 MAC, even higher overall traffic is possible, but an overall traffic of 1 packet/s is already very high covering most applications for WSNs. The length of the data frame is 108 bytes (see traffic model in Sec. 6.2.5).

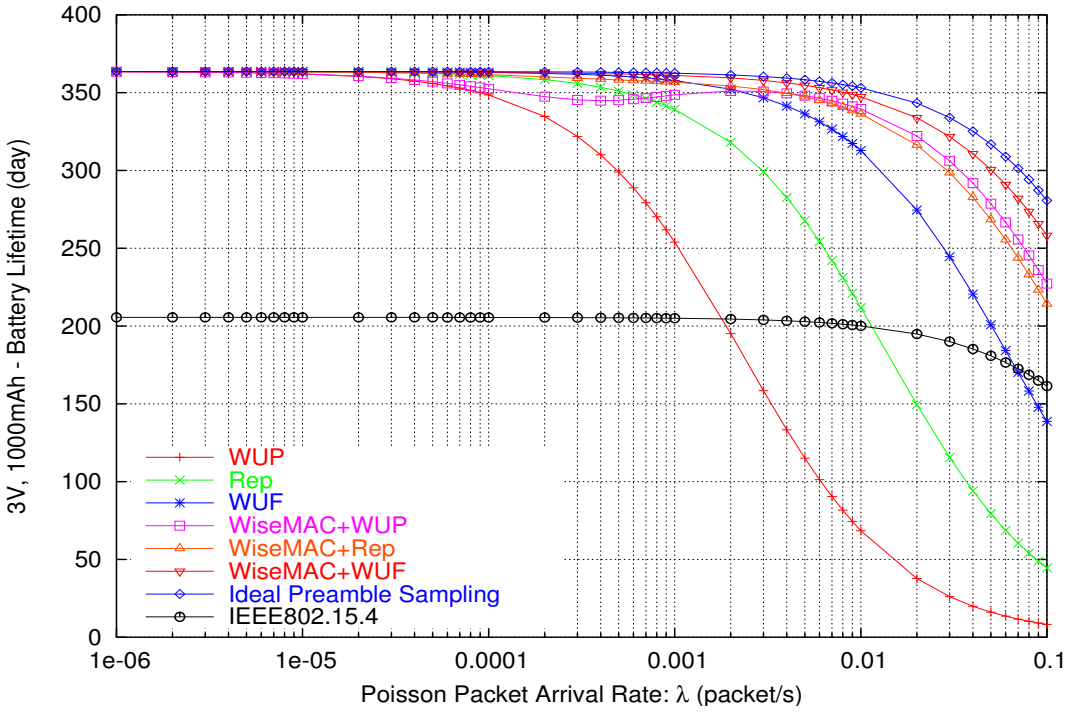


Figure 5.3: Battery Lifetime of all the Schemes
(10 slaves, Unicast, Downlink, Analytical)

From the Fig. 5.3, we can see that the IEEE 802.15.4 MAC is the least affected scheme by the increase of the traffic. However, its battery lifetime is short because of the inefficiency of beacon listening. The WUP scheme works very well when traffic is very low. However, it is worse than the IEEE 802.15.4 MAC in a high traffic load, because the unnecessary waiting time in both the destination and overhearers drastically decreases its battery lifetime. The Rep scheme shortens the average unnecessary waiting time from $0.5T_{cycle}$ to $1.5T_{data}$ for overhearers, so its battery lifetime is much longer than that of the WUP scheme

in high traffic load. However, since the $T_{data}$ is quite long in the Sindrion system, it still works not very well when traffic is very high. However, the WUF scheme has a further shortened average unnecessary waiting time of $1.5T_{swuf}$, which makes it have a battery lifetime that is almost always better than that of the IEEE 802.15.4 MAC.

The WiseMAC+WUP shortens the WUS especially when the traffic is high, so its battery lifetime is even better than the WUF scheme in high traffic load. However, its curve has a trough in medium traffic load, where it works not as good as the Rep scheme and the WUF scheme. To this end, the WiseMAC+Rep and the WiseMAC+WUF further improve the battery lifetime. However, the difference between these three schemes are not so clear in Fig. 5.3, because the small number of slaves causing too little overhearing. A clearer result is shown in Fig. 5.4, where the number of slaves is 100.

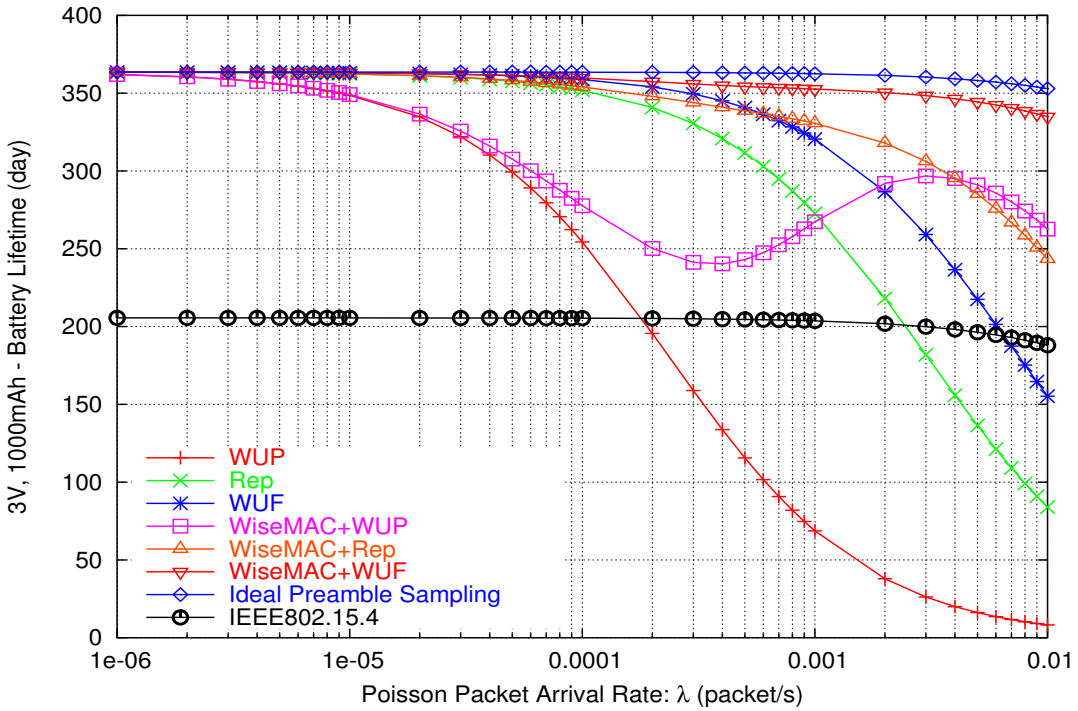


Figure 5.4: Battery Lifetime of all the Schemes
(100 slaves, Unicast, Downlink, Analytical)

In this case, the trough of the WiseMAC+WUP curve is much more distinct. The reason why the curve of the WiseMAC+WUP has such a shape can be explained as follows. When the traffic is very low, the WiseMAC+WUP makes no difference from the WUP

scheme, so they have almost the same tendency. With the increase of the traffic, more and more shortened WUSs are used, so the WiseMAC+WUP performs better than the WUP. When the traffic is further increased, the length of the WUS is quite short so that no much unnecessary waiting time is induced. As a result, the WiseMAC+WUP has even longer battery lifetime in high traffic case than in lower traffic case, so the curve goes up creating the trough. Nonetheless, the curve finally goes down again when the traffic is very high.

Compared to the WiseMAC+WUP, the Rep scheme and the WUF scheme are better in sense of power consumption during the trough of the WiseMAC+WUP curve, and are worse when the traffic is very high. Therefore, the combination schemes, i.e. the WiseMAC+Rep and the WiseMAC+WUF, generate better overall results as shown in the Fig. 5.4. The curves of the two combination schemes have no trough anymore but are monotonously descending.

Note that, in the very high traffic case, the performance of the WiseMAC+Rep is even worse than that of the WiseMAC+WUP. The reason is as follows. As mentioned in Sec. 4.1, different data rates are used for the WUS and normal frames, so that the data frame cannot wake up overhearers. However, the Rep scheme can use only one data rate for all repeated data frames, so the last repetition of the data can also wake up overhearers. When the traffic is high, the preamble is very short, so the WiseMAC+Rep generates much more overhearers than the WiseMAC+WUP wasting more energy. This phenomenon just shows the advantage of using different data rates for the WUS and normal frames. Note that an ACK can also wake up overhearers in the WiseMAC+Rep consuming more energy, which is not taken into consideration in the analytical calculation in Sec. A.6. However, the degradation caused by the ACK is supposed to be not very significant, because first the ACK is very short, and second a slave might not be able to hear an ACK from another slave because they could be out of the transmission range of each other.

The comparison between the WiseMAC+WUF and the WiseMAC+Rep is also shown in Fig. 5.4. When $\lambda$ ranges from $0.001$ packet/s to $0.01$ packet/s, the WiseMAC+WUF has a battery lifetime gain of from $14\%$ to $37\%$ over the WiseMAC+Rep. The reason behind is that, compared to the WiseMAC+Rep, the WiseMAC+WUF shortens the unnecessary waiting time of each overhearer by $1.5(T_{data} - T_{swuf})$ on the average for each transmitted data frame. So the longer and the more the data transmitted, the higher the gain that the WiseMAC+WUF achieves. Note that the WiseMAC+WUF shows a curve that is very close to the ideal case.

When the number of slaves is further increased to 1000, the battery lifetimes of the different schemes are shown in Fig. 5.5. Because of the large number of overhearers, the WiseMAC+WUP is always worse than the Rep scheme and the WUF scheme if the maximal overall traffic is limited to 1 packet/s. The WiseMAC+Rep also works not very well with such a large number of slaves. Its battery lifetime is shorter than the WUF scheme in most cases, and is similar to the IEEE 802.15.4 MAC when the traffic is very high. However, the WiseMAC+WUF still performs quite well, and a battery lifetime gain of up to $40\%$ over the WiseMAC+Rep can be achieved.

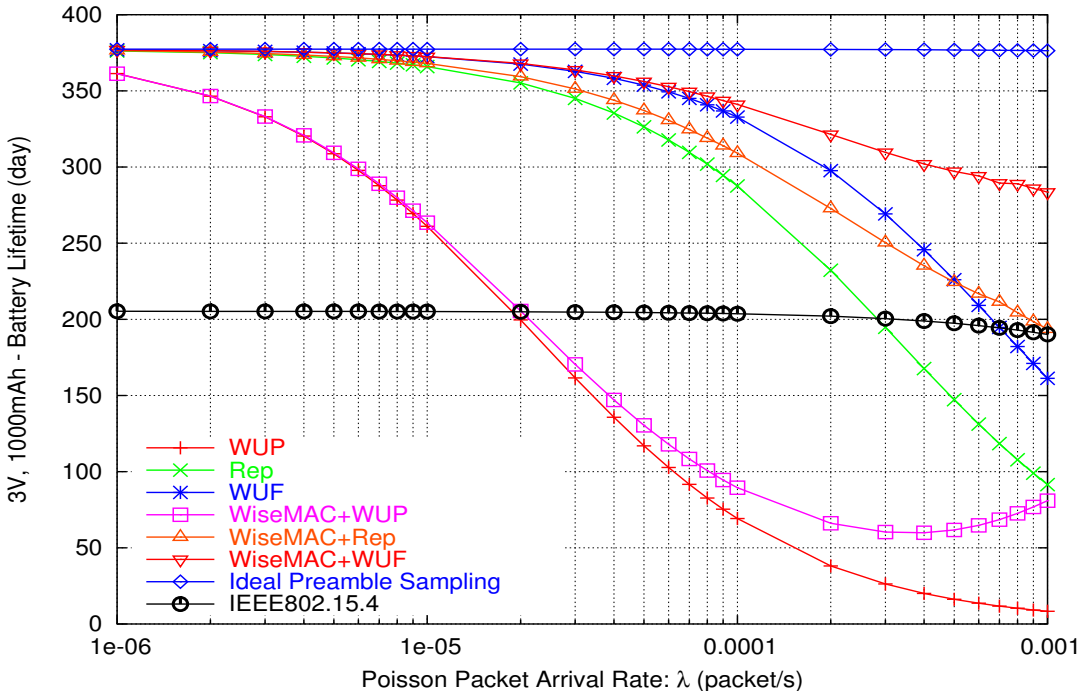


Figure 5.5: Battery Lifetime of all the Schemes
(1000 slaves, Unicast, Downlink, Analytical)

### Scalability

From Fig. 5.3 to Fig. 5.5, we can see that various schemes perform differently when the number of slaves increases. The scalability of these schemes are more clearly shown in Fig. 5.6, when the packet arrival rate to each slave is $\lambda = 0.001$ packet/s.

The curve of the IEEE 802.15.4 MAC is almost a constant showing that it scales very

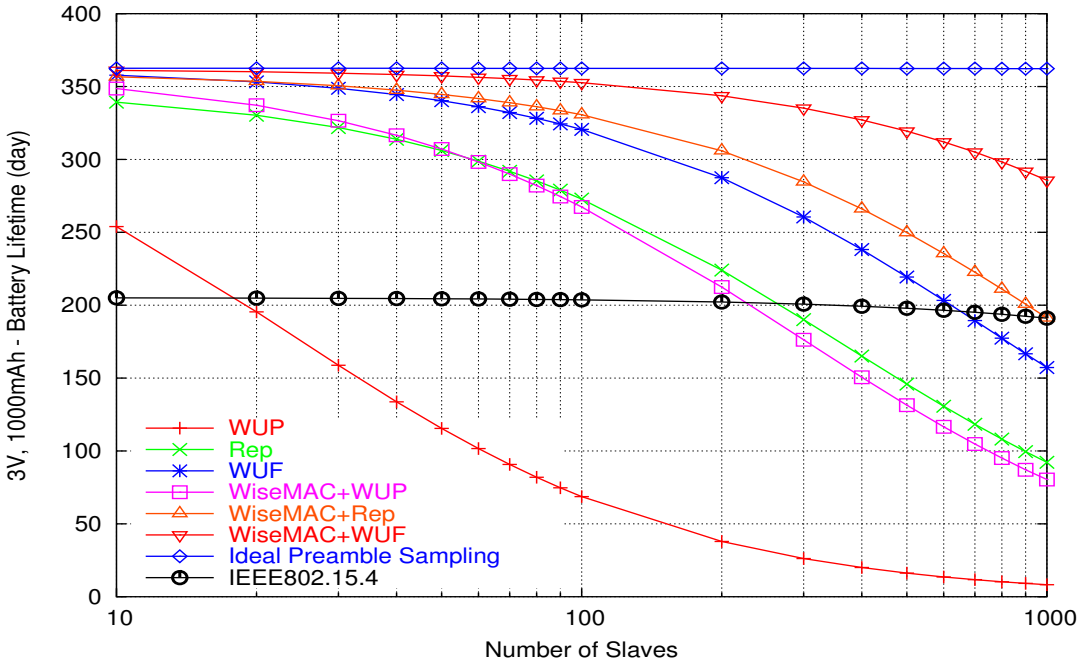


Figure 5.6: Scalability of all the Schemes
($\lambda = 0.001$, Unicast, Downlink, Analytical)

well. However, its battery lifetime is always at a relatively low level. The battery lifetime of the WUP scheme goes down very quickly when the number of slaves increases. Taking its curves in Fig. 5.3 to Fig. 5.5 into consideration, the WUP scheme is so sensitive to the traffic load and the number of slaves because of the long unnecessary waiting time that it is actually a practically infeasible scheme.

When $\lambda = 0.001$ packet/s, the Rep scheme and the WiseMAC+WUP have almost the same battery lifetime as well as the scalability. The WiseMAC+Rep is slightly better than the WUF scheme, but it scales not very well and decreases down to the level of the IEEE 802.15.4 MAC finally. However, the WiseMAC+WUF performs still quite well even when the number of slaves goes up to 1000.

### Frame Length

As mentioned in Sec. 3.4 and Sec. 4.4, data repetition is not efficient for long data frames. To estimate how the schemes are affected by the length of the data, a frame length ranging from 50 bytes to 500 bytes is used in calculation when the number of slaves is 100

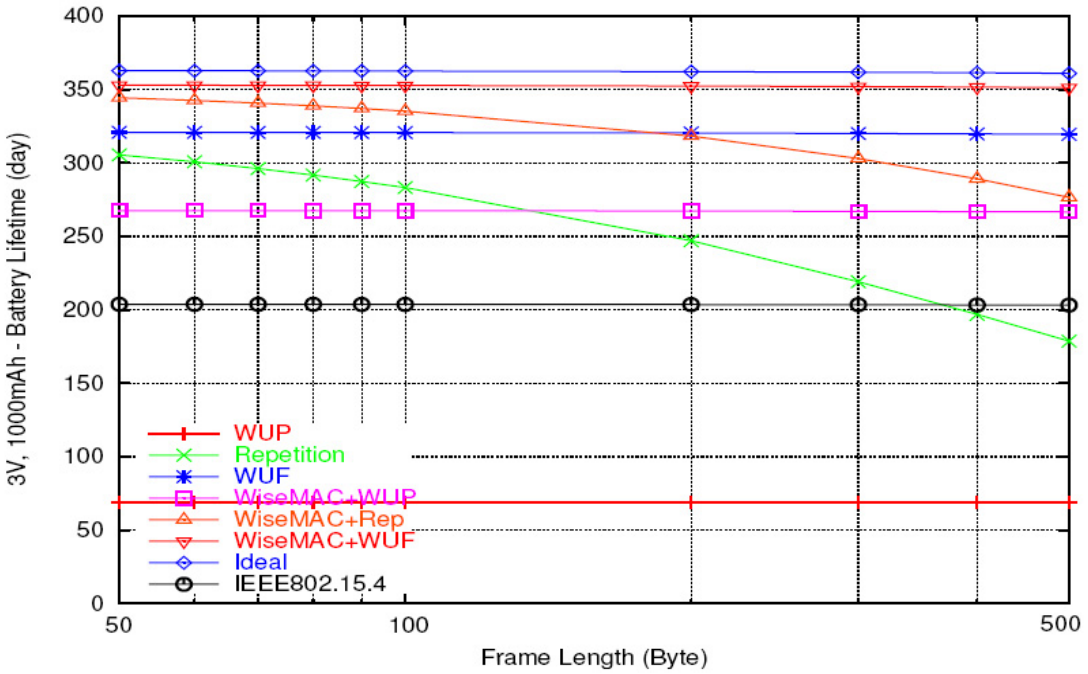


Figure 5.7: Frame Length of all the Schemes
($\lambda = 0.001$, Unicast, Downlink, Analytical)

and $\lambda = 0.001$ packet/s, as shown in Fig. 5.7. We can see that only the two data repetition schemes are sensitive to the frame length, because they have an average unnecessary waiting time of $1.5T_{data}$, while using WUF has a constant value of $1.5T_{swuf}$. Therefore, with the expansion of the frame length, the battery lifetime gain of the WiseMAC+WUF over the WiseMAC+Rep becomes increasingly bigger. When the frame length is over 200 bytes, the WiseMAC+Rep consumes even more power than the WUF scheme.

### Broadcast

The battery lifetime of different schemes with broadcast traffic is shown in Fig. 5.8, where $\lambda$ denotes the arrival rate of the broadcast packet from the master to all slaves. Since long WUS is necessary in the broadcast case for waking up all the slaves, the WiseMAC has no more effect, and there is also no ideal preamble sampling case. As a result, only the WUP scheme, the Rep scheme, the WUF scheme, and the IEEE 802.15.4 MAC are compared.

When traffic is very low, the preamble sampling has nearly twice of the battery lifetime as the IEEE 802.15.4 MAC. When the traffic increases, the WUP scheme again declines

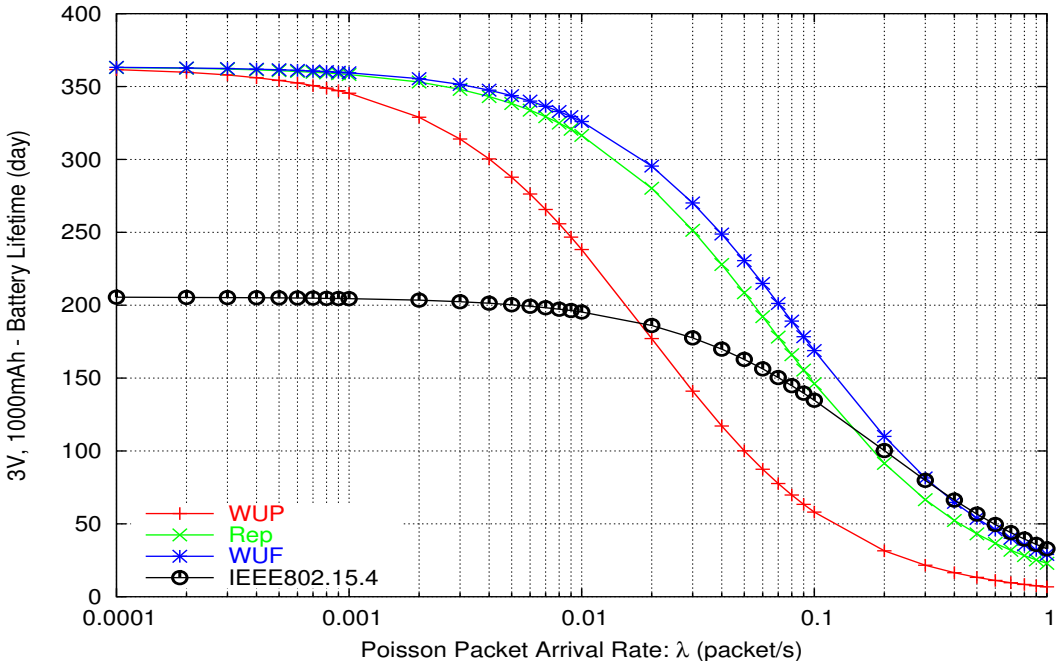


Figure 5.8: Battery Lifetime of all the Schemes
(Broadcast, Downlink, Analytical)

quickly, while the Rep scheme and the WUF scheme show much better results. However, the WUF scheme has only slight gain over the Rep scheme, because there is no overhearing in broadcast case. Note that the results shown in Fig. 5.8 are valid for any number of slaves.

### 5.2.2 Channel Occupation

As mentioned in Sec. 3.3.2, one big drawback of the preamble sampling scheme is the low channel efficiency caused by the long WUS. To this end, the WiseMAC shortens the length of the WUS to improve the power consumption as well as the channel efficiency. Next, the channel occupation of the basic preamble sampling scheme including the WUP, Rep, and WUF scheme, the WiseMAC+WUP/Rep/WUF, and the IEEE 802.15.4 MAC will be compared. Fig. 5.9 shows the channel occupation versus the packet arrival rate $\lambda$ in a star topology with 10 slaves.

The channel occupations of the three variants in the basic preamble sampling scheme are basically the same, which goes up drastically with the increase of the $\lambda$. However,

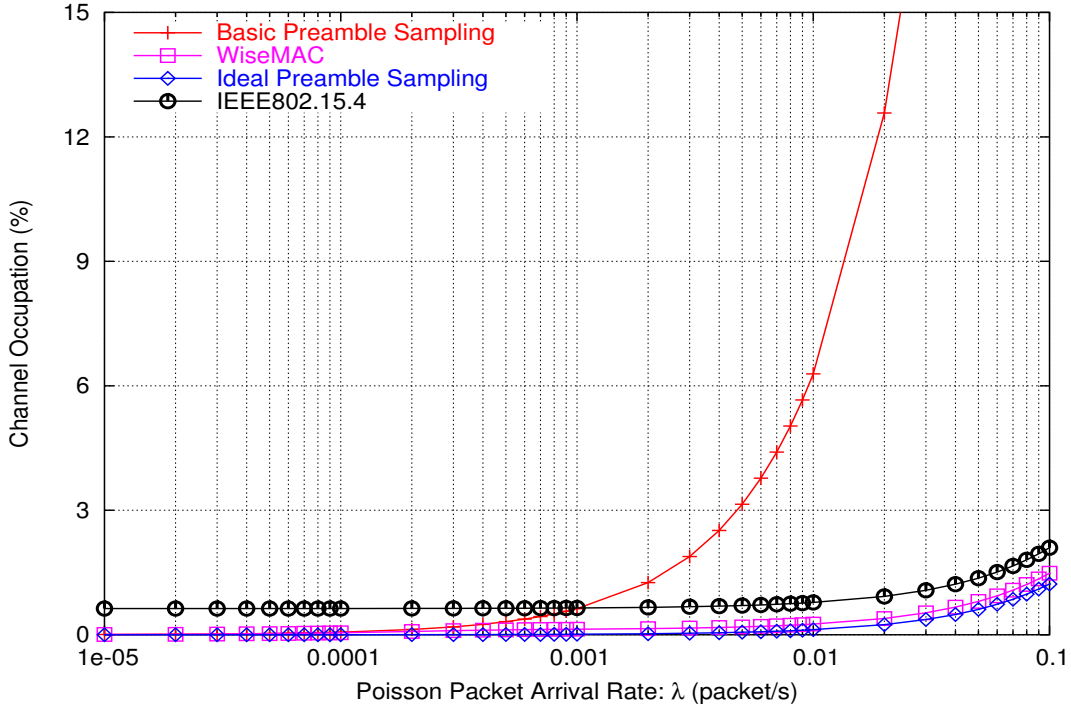


Figure 5.9: Channel Occupation of all the Schemes
(10 slaves, Unicast, Downlink, Analytical)

when $\lambda < 0.001$ packet/s, the basic preamble sampling scheme occupies even less channel than the IEEE 802.15.4 MAC. This is the working range of the basic preamble sampling scheme. When $\lambda > 0.01$ packet/s, the basic preamble sampling scheme occupies too much channel to work well. The three variants of the WiseMAC have also basically the same channel occupation, which is very close to the ideal case and is always much smaller than that of the IEEE 802.15.4 MAC.

Fig. 5.10 shows the channel occupation of the schemes with the number of slaves of 100. When $\lambda$ is very small, the WiseMAC occupies still less channel than the IEEE 802.15.4 MAC. When $\lambda$ increases, the channel occupation of the WiseMAC surpasses that of the IEEE 802.15.4 MAC. However, the difference in between is quite small.

When the number of the slaves is very big, e.g. 1000, the channel occupation of the WiseMAC is much bigger than that of the IEEE 802.15.4 MAC, as shown in Fig. 5.11. However, it is still limited into a reasonable range compared to the basic preamble sampling scheme.

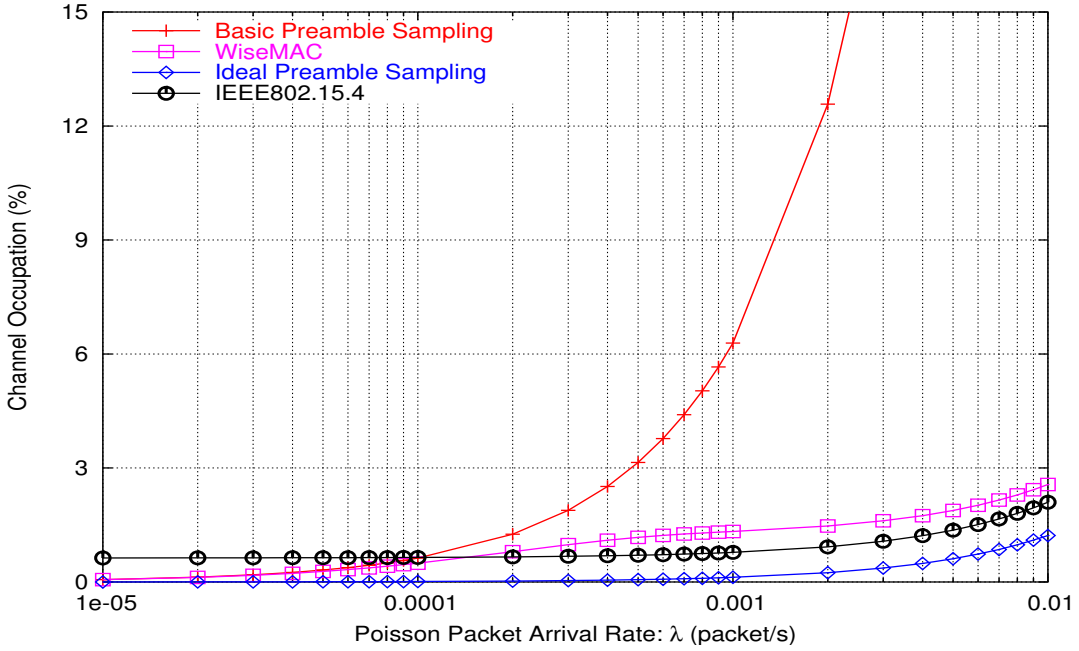


Figure 5.10: Channel Occupation of all the Schemes
(100 slaves, Unicast, Downlink, Analytical)

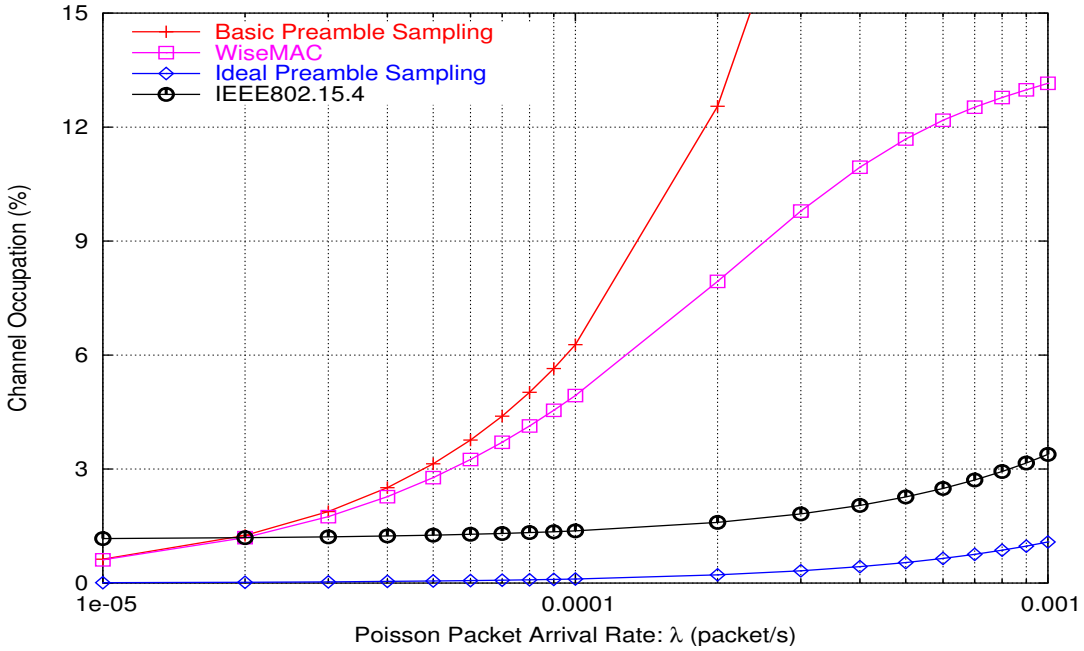


Figure 5.11: Channel Occupation of all the Schemes
(1000 slaves, Unicast, Downlink, Analytical)

From the Fig. 5.9 through Fig. 5.11, we can see that the channel occupation of the WiseMAC is affected by two major facts. First, the smaller is the $\lambda$, the longer is the WUS, so the more is the channel occupied by each data transmission. Second, the bigger is the $\lambda$ and the number of slaves, the more is the overall traffic, so the higher is the channel occupation.

### 5.2.3 Response Time

Another very important system parameter in WSNs is response time, which is a requirement determined by the application. The average response time of the schemes are shown in Fig. 5.12. The response time for a unicast packet is defined as the time difference between the time point that a packet is available in the transmitter's network layer ready for MAC transmission and the time point that the corresponding ACK is received by the transmitter's MAC layer from the receiver.

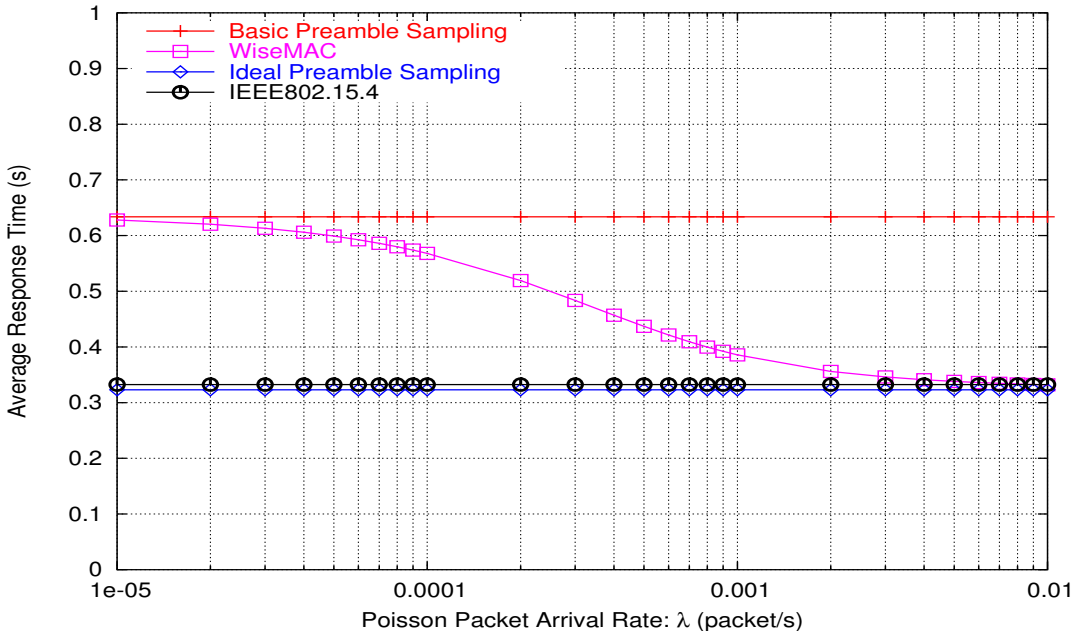


Figure 5.12: Response Time of all the Schemes
(100 slaves, Unicast, Downlink, Analytical)

The average response time of the basic preamble sampling scheme is around one cycle period, and that of the IEEE 802.15.4 MAC is around half cycle period. To this end, one

question may arise whether it is fair to compare the battery lifetime of them when they have different response time. To answer this question, it should be considered that the response time of the IEEE 802.15.4 MAC is uniformly distributed roughly between zero and one cycle period, whereas the response time of the basic preamble sampling scheme is just around one cycle period. It means that both schemes are designed to fulfill the response time of one cycle period, although the IEEE 802.15.4 MAC has a shorter average response time.

The average response time of the WiseMAC is not a constant but a descending curve with the increase of the $\lambda$. It is because that the bigger is $\lambda$, the shorter is the WUS, so the shorter is the response time.

### 5.2.4 Others

Besides the analytical comparison of the performance of the difference schemes, other two issues are also worth being quantitatively discussed as follows.

#### DRD vs. RSSI

As mentioned in Sec. 4.1, the used RF transceiver provides a DRD function during sampling, so that the RF transceiver only can be woken up by a signal with certain data rate and RSSI. Compared to the simple RSSI criterion, strong noise and interference from other systems using the same frequency band cannot wake up the RF transceiver. Thus, energy is saved because of the fewer unnecessary wake-up.

However, the DRD takes the reception time of at least 3 Manchester-coded bits (6 symbols) plus the time needed for setting up the RSSI, which is longer than the time needed by the simple RSSI criteria, i.e. only the RSSI setup time.

If $70$ ksymbol/s data rate is used for the WUS, 6 symbols take $0.086$ ms. However, the measured results show that a practical value is $0.4$ ms because of the disturbance of the channel. Including the setup time of the RF transceiver receive mode ($2.2$ ms) and the setup time of the RSSI ($0.4$ ms), the actual working time of one channel sampling is $3$ ms using DRD. However, using only RSSI takes $2.6$ ms.

As a result, using DRD increases the sampling time as well as the power consumption. Therefore, whether it is worth to use DRD should be decided after further quantitative

analysis, see Appendix A.9 for the detailed analytical computation based on the WUF scheme.

Fig. 5.13 shows the result of the analytical computation. In the no traffic case, using DRD keeps a constant battery lifetime because there is ideally no wake-up caused by noise and interference. When the occupied channel by noise and interference is smaller than $3\%$, using simple RSSI has a slightly longer battery lifetime than using DRD because of the shorter sampling time. However, when the percentage of channel occupation increases, the battery lifetime of using simple RSSI decreases dramatically.

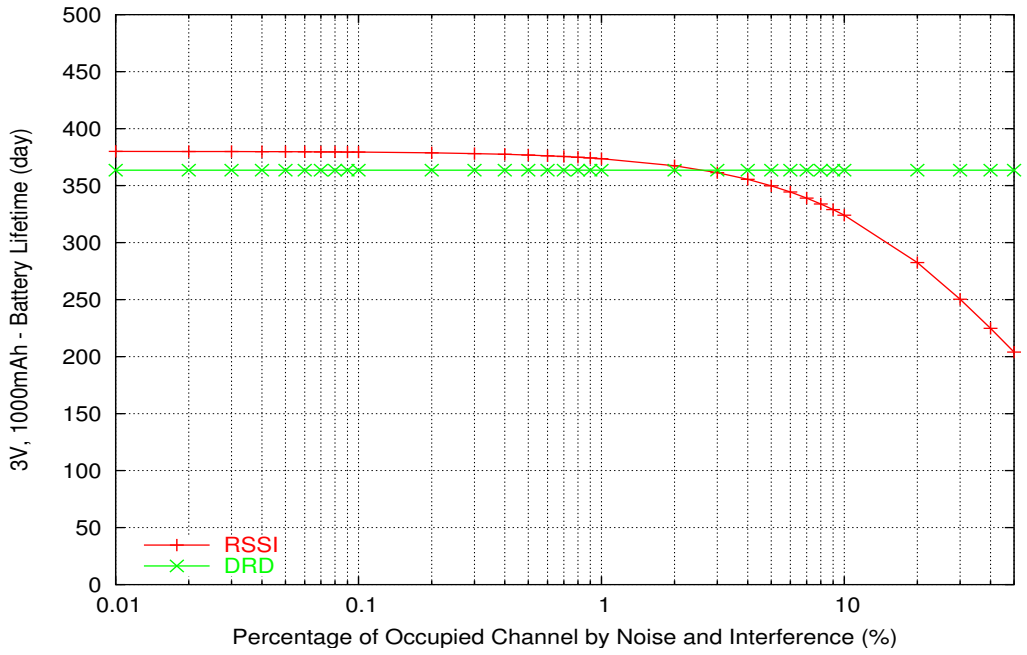


Figure 5.13: RSSI vs. DRD (No Traffic, Analytical)

The ISM-band is shared by a lot of wireless communication systems, and it is also subject to noise caused by electronic appliances such as microwave ovens. Thus, there is typically a substantial amount of interference and noise. Therefore, although using simple RSSI achieves slightly better battery lifetime in the low-noise case, using DRD should be a better choice due to its much better performance for typical channel conditions.

Furthermore, with the DRD it is also possible to use different data rates for the WUS and the data frame respectively, so that the data frame cannot wake up overhears, as mentioned in Sec. 4.1 and Sec. 5.2.1, giving more power-saving potentials.

### Short MAC Address and Header Compression

As aforementioned in Sec. 3.2, *fewer transmission* is supposed to be not an efficient way to save power in a low-traffic WSN. To prove this assumption, header compression and short MAC address are simulated as examples. According to [15], the 40-byte TCP/IP header is compressed to 3 bytes through some negotiation between the source and destination. In [14], short MAC address with minimal variable length is dynamically assigned to each device, which also needs some negotiations.

To simplify the simulation, first, the negotiation is neglected; second, UDP/IP packet with 28-byte header is used instead of the TCP/IP packet, which is also shortened to 3 byte; third, a 1-byte short MAC address is assumed for both source and destination instead of the 8-byte IEEE MAC address [39]. As a result, totally $(28-3)+(8-1)\times 2 = 39$ bytes are saved for each transmitted data packet. Since a normal data packet before ECC and modulation coding is 70-byte long (see Sec. 6.2.5), the length of the data packet is compressed to less than $50\%$ of the original length, which is a quite high compression ratio.

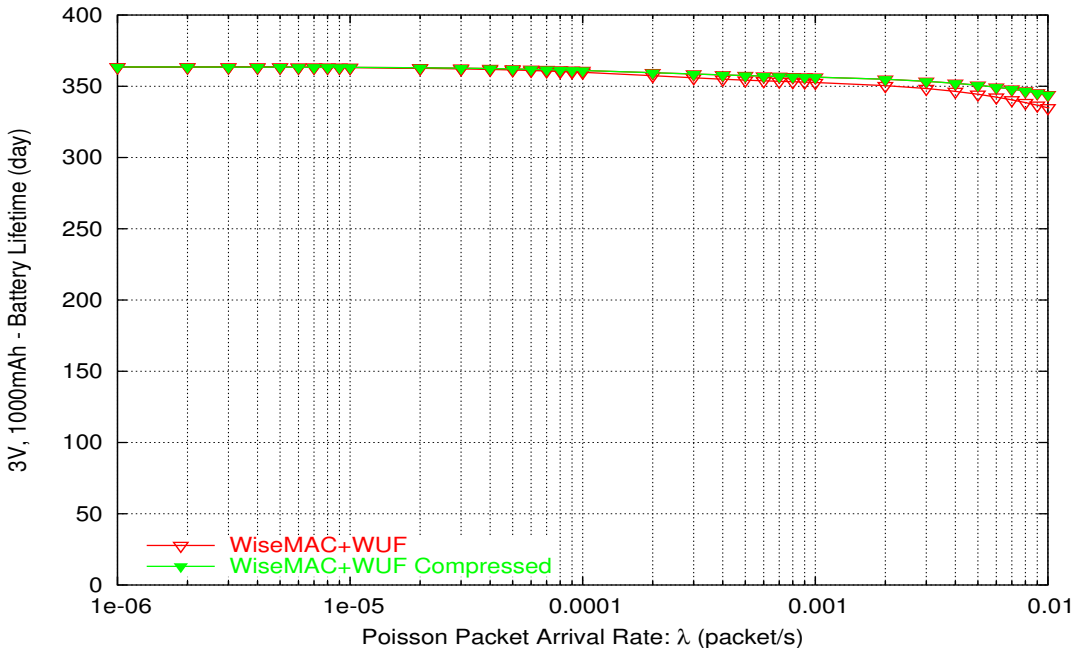


Figure 5.14: Battery Lifetime of WiseMAC+WUF
(Data Compression, 100 Slaves, Unicast, Analytical)

Fig. 5.14 shows the analytical results based on the downlink traffic in star topology with 100 slaves. It clearly illustrates that the battery lifetime gain is very small with such a high compression ratio. If the negotiation overhead is included, the gain will be further smaller or even negative.

From this result, it is also expected that other methods of *fewer transmission* such as information fusion and minimized transmission power have also no significant effect on power-saving in low-traffic WSNs, because most of the system power is consumed by standby mode instead of data transmission.

## 5.3 Analytical Conclusion

From the results of analytical computation, the following conclusions can be made.

In the sense of power consumption, the WUF scheme is a better choice compared to the WUP scheme and the Rep scheme. Its power consumption is also much lower than that of the IEEE 802.15.4 MAC except when the traffic is very high. By combining the WUF scheme and the WiseMAC, the power consumption is further reduced. When the traffic is high, the WiseMAC+WUF scheme is much more power efficient than the WiseMAC+Rep and the IEEE 802.15.4 MAC.

In the sense of channel efficiency, the WUF scheme shows even better results than the IEEE 802.15.4 MAC, when the overall traffic is very low. However, the WUF scheme is not suitable for high traffic networks. To this end, the WiseMAC+WUF has a much lower channel occupation. When the number of slaves is not too big (less than 100), the WiseMAC+WUF is more channel efficient than the IEEE 802.15.4 MAC. When the number of slaves as well as the overall traffic increases to a very high level, the WiseMAC+WUF occupies much more channel than the IEEE 802.15.4 MAC. However, the channel occupation is still limited within a reasonable range compared to the WUF scheme.

Another very important point is that the IEEE 802.15.4 MAC supports power-saving for only one master, so it cannot be used for networks with multiple masters, e.g. the Sindrion.

The above conclusion is according to analytical computation, which is just a rough estimation based on optimistic assumptions. To prove the correctness of the analytical computation, and to further evaluate the performance of the schemes in a more realistic

environment, simulation should be conducted. The simulation models will be introduced in Chp. 6, and simulation schemes and results will be illustrated in Chp. 7.

# Chapter 6

# Simulation Models

## 6.1 Simulation Framework

In Chp. 5, the performance of the proposed WUF scheme and of the WiseMAC+WUF scheme have been compared to all the other schemes by analytical computation. However, the analytical results are based on optimistic assumptions: one master star topology, only downlink traffic, no collision, and no noise or interference. These are not the real working situation for a WSN. Therefore, further evaluation should be conducted to estimate the performance of the schemes based on more realistic traffic and channel models. For this purpose, a simulation platform is established and will be introduced in this chapter.

The simulation platform is programmed by C. The framework of the simulation platform is shown in Fig. 6.1, and the functions of the blocks are as follows.

- **Channel Model:** A new indoor wireless channel model is proposed to support more accurate simulation of the wireless indoor environment, see Sec. 6.2.1.
- **RF Transceiver:** The receive, transmit, sleep, self-polling, and DRD functionality of the RF transceiver is modelled, see Sec. 6.2.2.
- **FPGA:** An individual module is programmed to realize the functionality of the FPGA, e.g. modulation and ECC (de)coding for the PHY layer, and CRC (de)coding, MAC address decoding, and the wake-up logic for the MAC layer, see Sec. 6.2.3.

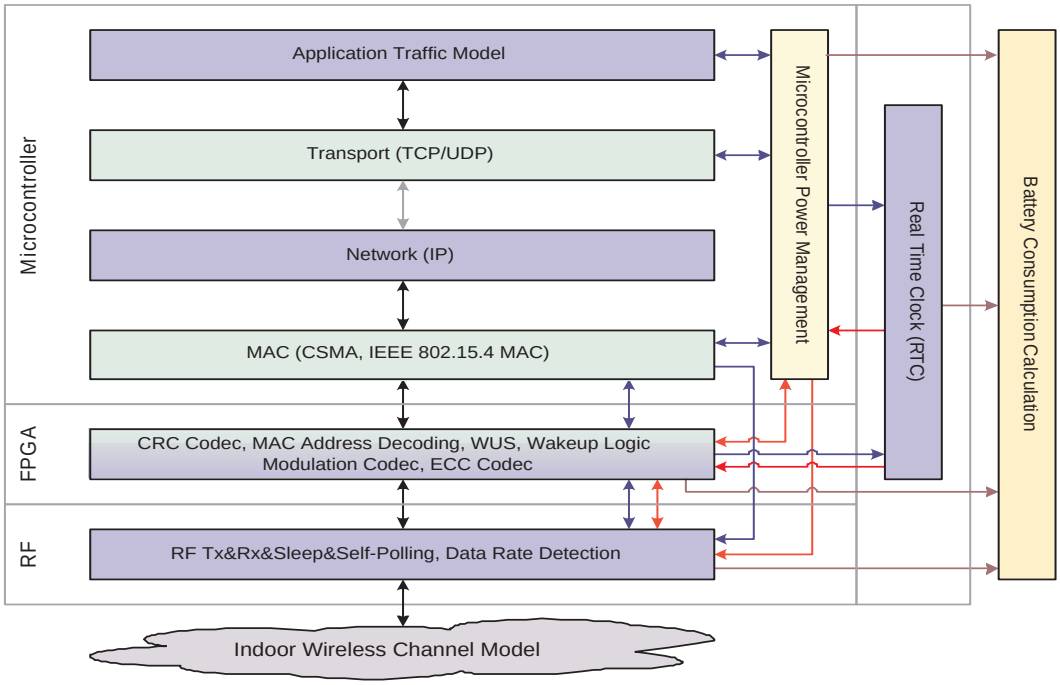


Figure 6.1: The Simulation Framework

- **Microcontroller MAC:** The preamble sampling based schemes and the IEEE 802.15.4 MAC are implemented as a part of microcontroller, see Sec. 6.2.4.
- **Microcontroller TCP/UDP/IP:** The TCP, UDP and IP are implemented.
- **Microcontroller Application:** The Poisson process packet arrival model is implemented, see Sec. 6.2.5.
- **Microcontroller Power Management:** Through this block, the FPGA and the RTC can wake up the microcontroller. It also collects information from the MAC layer and the application layer to decide when the FPGA and the RF transceiver should be woken up.
- **RTC:** The RTC can be set by the FPGA and the microcontroller. It also can wake up the FPGA and the microcontroller.
- **Battery Consumption:** This block collects the power consumption information from the RF transceiver, the FPGA, the RTC, and the microcontroller to calculate the overall system power consumption and battery lifetime.

In the following section, the simulation models of the hardware and the individual protocol layers will be introduced in detail.

## 6.2 Protocol Layers and Models

### 6.2.1 Indoor Channel Model

The channel propagation model determines the quality of the received signal based on the transmitted signal and certain channel characteristics. Accordingly the Bit Error Rate (BER) can be estimated as well as the packet loss rate. A good channel model is very important for protocol simulation, especially in wireless communication, because the dynamic nature of the wireless channel typically causes many bit errors.

The Sindrion network, as well as many other WSNs, is mainly deployed in indoor environments, so that a proper indoor wireless channel model should be used in the simulation. However, there is no very good indoor wireless channel model for networking protocol simulation so far. To this end, we propose to integrate a well-established channel impulse response based indoor channel model, called SIRCIM, into the packet-wise SNR-based wireless channel model that is normally used in the protocol simulation.

#### General Wireless Channel Models

The NS2 [40] and the GloMoSim [41] are two mostly used tools for communication network simulation. Their wireless channel propagation models yield the total power pathloss of the channel. Knowing the transmission power and the noise level on the channel, the received power as well as the Signal-to-Noise Ratio (SNR) for each received packet can be derived. Then the BER is estimated according to the SNR by a certain algorithm. We call this concept as SNR-based channel model. The general wireless channel models used in the NS2 and the GloMoSim are as follows.

- ***Pathloss Model:***
  The large scale effect of the channel is inversely proportional to the antenna separation distance, where this distance is raised to an exponent 2, as given by the Friis Free Space model [40, p187]

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L}, \tag{6.1}$$

where $P_r$ and $P_t$ denote the received and transmitted power, $G_t$ and $G_r$ are the respective gains of the transmitter and the receiver antenna, $\lambda$ is the wavelength of the signal, and $L > 1$ is the system loss that is not related to propagation. The pathloss defined by $\frac{P_r}{P_t}$ can thus be represented in dB by

$$L_p(d) = L_p(d_0) + 10 \log \left( \frac{d_0}{d} \right)^2, \tag{6.2}$$

where $d_0$ is some reference distance and $L_p(d_0)$ is the pathloss at this distance.

- ***Fading Model:***
  To add fading effects into the free space model, a so called log-normal shadowing model is given by

  $$L_p(d) = L_p(d_0) + 10 \log \left( \frac{d_0}{d} \right)^n + X_{dB}, \tag{6.3}$$

  where the exponent becomes $n$ instead of 2, and $X_{dB}$ is a Gaussian random variable with zero mean and standard deviation $\sigma$ [dB]. This log-normal shadowing model is used in NS2 to simulate both the pathloss and the fading effects [40].

  Instead of using the log-normal shadowing model to simulate fading, fading can also be modelled by a random variable that is Rayleigh or Rice distributed added to the pathloss generated by the free space model. This is adopted by the GloMoSim.

  When there is a Line-Of-Sight (LOS) path between the transmitter and the receiver, the pathloss is generated using a Rice distribution:

  $$P(r) = \frac{r}{\sigma^2} \exp\{\frac{-r^2 + v^2}{2\sigma^2}\} I_0(\frac{rv}{\sigma^2}), \qquad r \geq 0, \tag{6.4}$$

  where $r$ is the amplitude gain caused by the fading, $v$ is the mean amplitude gain of the strong LOS path, and $\sigma$ is the deviation.

  For the channel has no LOS path, i.e. it is OBStructed (OBS), the pathloss is generated using a standard deviation $\sigma$ for the Rayleigh distribution:

  $$P(r) = \frac{r}{\sigma^2} \exp\{-\frac{r^2}{2\sigma^2}\}, \qquad r \geq 0, \tag{6.5}$$

  where $r$ is the amplitude gain caused by the fading.

The above general wireless channel models are typically used for outdoor environments. The indoor wireless channel is much more difficult to model than the outdoor channel, because it is susceptible to the changes in the geometry of indoor environment, e.g. a door being shut and a person walking around one of the antennas. Therefore, a special indoor wireless channel model is needed to achieve more accurate simulation. So far, the most widely accepted indoor wireless channel model is called SIRCIM.

### SIRCIM Model [42]

A multipath fading wireless channel can be modelled as a linear time-varying filter and represented by an impulse response

$$h(t) = \sum_{K} A_K e^{j\theta_K} \delta(t - T_K), \tag{6.6}$$

where $A_K$ represents the amplitude, $e^{j\theta_K}$ denotes the phase shift caused by reflection, diffraction and scattering, and $T_K$ is the time delay of the $Kth$ path in the channel with respect to the arrival time of the first arriving component, called excess delay. The received signal can be calculated by applying the convolution of the transmitted signal with this impulse response.

The fluctuations of the amplitudes, phases and multipath delays of a signal can be referred to as fading. There are three types of fading: large scale fading, medium scale fading, and small scale fading. Large scale fading is caused by the general structure and geometry of the environment. Medium scale fading is also referred to as shadowing and is influenced by diffraction because of the absence of LOS between the two antennas. Small scale fading results from reflection and scattering at objects near the receiver.

Although indoor and outdoor channels share the above basic characteristics, the indoor channel cannot be viewed as a scaled down version of the outdoor channel, because it is more susceptible to severe pathloss, non-stationarity, low doppler spread and small excess delay.

There exist two types of general topographies found in the indoor environment: LOS and OBS [43, 44]. Each of the two can be further classified into three smaller groups: *open plan* (OPEN), *hard partitioned* (HARD) and *soft partitioned* (SOFT) [45]. OPEN buildings are those that have large open spaces where exists only few large obstructions or scatterers, e.g. factories. HARD buildings are typical multiple floor buildings partitioned by concrete or drywall, e.g. offices. SOFT buildings are also multiple floor buildings with large open spaces but partitioned into small offices using dividers that do not extend from the floor to the ceiling.

In [42], Seidel and Rappaport present a model that determines a channel impulse response for different types of environments, whether it is LOS or OBS in an OPEN, HARD or SOFT environment. To this end, they provide typical parameters for the impulse response of Equ. 6.6 based on empirical observations. In particular, they characterize the

distribution of the number of multipath components, the probability of receiving a multipath component at a particular excess delay, the distributions of the mean amplitude and phase of each multipath component, and the distribution of the large and small scale fading of each multipath component. The spatial and temporal correlations among the multipath components are also modelled. These parameters gather the information that is necessary to statistically describe a wireless channel. The authors also developed a program to simulate their statistical model called Simulation of Indoor Radio Channel IMpulse response (SIRCIM).

Next, the procedure how the SIRCIM model generates a channel impulse response will be introduced. An OPEN building with LOS case is used as an example. For other cases, the steps are the same except for some differences in the equations and parameters, see [45] for details.

1. *Distribution of the Number of Multipath Components*

   The number of multipath components $N_p$ is taken to be Gaussian distributed with a mean $\overline{N_p}$ and a standard deviation $\sigma_p$. The mean of this distribution, $\overline{N_p}$, is also a random variable being uniformly distributed between 9 to 35. The standard deviation $\sigma_p$ is modelled by

   $$\sigma_p = 0.492 \times (\overline{N_p} - 4.77). \tag{6.7}$$

2. *Probability of the Arrivals of Multipath Components*

   The probability that a multipath component will arrive at a receiver at a particular excess delay is modelled by piecewise linear functions of the excess delay:

   $$P(T_K) = \begin{cases} 1 - \frac{T_K}{367}, & T_K < 110ns \\ 0.65 - \frac{(T_K - 110)}{360}, & 110ns < T_K < 200ns \\ 0.22 - \frac{(T_K - 200)}{1360}, & 200ns < T_K < 500ns, \end{cases} \tag{6.8}$$

   where $T_K$ is the delay and takes values that are integer multiples of $7.8\,ns$.

3. *Distribution of the Phases of the Multipath Components*

   The phases for each multipath component $\theta_K$ are uniformly distributed within $[0, 2\pi)$ [46].

4. *Large Scale Fading of Multipath Components*

   The large scale fading amplitude of each component $K$ is log-normally distributed around a mean amplitude $\overline{A_K}$ with a standard deviation $\sigma_{large-scale}$ of 4 dB. The

mean amplitude $\overline{A_K}$ in dB obeys the exponential law

$$\overline{A_K}(T_K) = 10 \times n(T_K) \times \log\left(\frac{d}{10\lambda}\right), \tag{6.9}$$

where $d$ is the distance between the two antennas, and $\lambda$ denotes the wavelength of the signal. The distribution of $n$ is given by

$$n(T_K) = \begin{cases} 2.5 + \frac{T_K}{39}, & T_K < 15ns \\ 3.0 + \frac{(T_K - 15.6)}{380}, & 15ns < T_K < 250ns \\ 3.6, & 250ns < T_K < 500ns. \end{cases} \tag{6.10}$$

5. *Small Scale Fading of Multipath Components*

   The cumulative distribution function for $\sigma_{small-scale}$ is given by

$$F(\sigma_{small-scale}) = 1 - exp\left(\frac{-(\sigma_{small-scale} - a)^2}{2}\right), \tag{6.11}$$

   where the offset parameter $a$ is $0.25$ dB.

In total, the amplitude of an individual multipath component can be modelled by the distribution

$$A_K(T_K) = N_{log}\left[N_{log}\left[\overline{A_K}, \sigma^2_{large-scale}\right], \sigma^2_{small-scale}\right], \tag{6.12}$$

where $N_{log}[x, \sigma_x^2]$ denotes the log-normal distribution with mean $x$ (dB) and standard deviation $\sigma_x$ (dB). As a result, with Equ. 6.7 to 6.12, a statistical discrete channel impulse response can be generated for a particular channel in an OPEN building with LOS.

### Convert SIRCIM into SNR-based Model

The indoor wireless channel impulse response generated by the SIRCIM model is closer to the reality compared to other indoor channel models. However, the channel impulse response should be convoluted with the transmitted signal symbol by symbol to get the received signal. This is very computation and time consuming. On the other side, the SNR-based model is a much simpler model because it needs only the overall channel power pathloss for each packet. To this end, we propose to derive a total power pathloss from the information given by the SIRCIM channel impulse response, so that the SIRCIM model can be used as an SNR-based model. The way of the conversion is as follows.

First, the total amplitude gain at the receiver can be obtained by summing up all the multipath components generated by the SIRCIM as phasors, which is given by

$$A = \left| \sum_{K=1}^{N_p} A_K e^{j\phi_K} \right|, \tag{6.13}$$

where $N_p$ is the number of multipath components in the channel impulse response, $A_K$ denotes the linear amplitudes of the discrete impulse response, and $\phi_K$ represents the total phase shift. Note that $\phi_K$ is the sum of two phase shifts – the phase shift caused by excess delay $\theta'_K(T_K)$, and the phase shift caused by reflection, diffraction and scattering $\theta_K$. Since $N_p$, $A_K$, $T_K$ and $\theta_K$ are known parameters generated by the SIRCIM, the only parameter left to be determined is $\theta'_K(T_K)$, which is given by

$$\theta'_K(T_K) = \frac{(T_K \cdot c_0)\, mod\, \lambda}{\lambda} \cdot 2\pi, \tag{6.14}$$

where $c_0$ is the velocity of light and $\lambda$ is the wavelength of the carrier frequency. Another simple but reasonable way to get $\phi_K$ is assuming this overall phase shift as a uniformly distributed random number within $[0, 2\pi)$.

The phasorial sum of Equ. 6.13 can be made by breaking each amplitude $A_K e^{j\phi_K}$ into an in-phase component $A_{K-I}$ and a quadrature component $A_{K-Q}$ given by

$$A_{K-I} = A_K \cos\phi_K, \tag{6.15}$$

and

$$A_{K-Q} = A_K \sin\phi_K. \tag{6.16}$$

Then summing up all the in-phase components and the quadrature components separately produces a net in-phase component $A_I$ and a net quadrature component $A_Q$. Hence, the resulting amplitude gain $A$ is calculated as

$$\begin{aligned} A &= \sqrt{A_I^2 + A_Q^2} \\ &= \sqrt{\left(\sum_{K=1}^{N_p} A_{K-I}\right)^2 + \left(\sum_{K=1}^{N_p} A_{K-Q}\right)^2} \\ &= \sqrt{\left(\sum_{K=1}^{N_p} A_K \cos\phi_K\right)^2 + \left(\sum_{K=1}^{N_p} A_K \sin\phi_K\right)^2}. \end{aligned} \tag{6.17}$$

$A$ is the final total channel *amplitude gain*. Thus, the total channel *power pathloss* needed by the SNR-based model is $1/A^2$, including both the propagation and fading effects.

### Advantages of Conversion

Although traditional SNR-based channel models can somehow simulate the indoor channel by properly defining their parameters, e.g. the $n$ and the $X_{dB}$ in the log-normal shadowing model [40, p188] and the K-factor in the Rice fading [47], they still cannot achieve precise

results because they are quite simple and cannot present the dynamic features of complex indoor environments.

The SIRCIM model considers different information regarding the type of environments. This statistical model is based on large amount of measurements. Thus, it is more consolidated. Therefore, converting the SIRCIM model into an SNR-based model gives a more realistic power pathloss for indoor channel.

Furthermore, converting SIRCIM to an SNR-based model also provides the information of the Inter Symbol Interference (ISI), which is a very important channel characteristic missed in the state-of-the-art SNR-based models. However, the conversion of the SIRCIM gives the possibility to do more accurate BER estimation based on both the pathloss and the ISI.

### Simulation Results

To evaluate the proposal, the converted SIRCIM is compared to other channel models.

Fig. 6.2 shows the pathloss Probability Density Function (PDF) of the SIRCIM model in the LOS case. For the OPEN environment, the PDFs for different distances (10m, 20m, 30m) have almost the same shape, while the only difference is the increasing mean attenuation due to the distance. In the HARD/SOFT environment, both the mean and the standard deviation increase with the increasing distance, showing that the HARD/SOFT environment is much more dynamic than the open plan environment. Furthermore, the SOFT environment has a bit larger mean and deviation than the HARD environment. This can be explained by the fact that in the LOS case, the *soft* separators in an office will introduce more unpredictable factors than the *hard* walls.

Fig. 6.3 shows the PDFs of the pathloss according to the SIRCIM model in the OBS case. The first obvious difference from the LOS case is that the means of the pathloss increase more significantly with the distance. The second is that the HARD and SOFT environments have basically the same pathloss PDF. However, the mean attenuation of the SOFT environment is slightly smaller than that of the HARD environment because an obstructed receiver can collect more indirect paths in a SOFT environment than in a HARD one.

In Fig. 6.4 and Fig. 6.5, the mean pathloss of the Friis free space model with Rice or Rayleigh fading, of the shadowing model, and of the SIRCIM model are shown. The mean

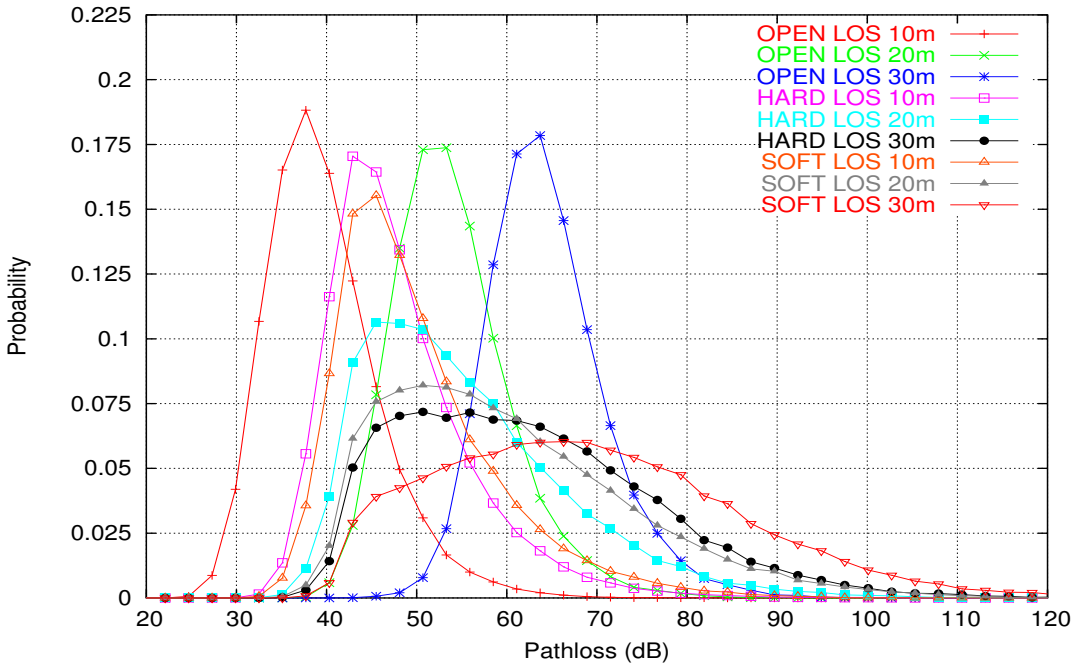


Figure 6.2: Pathloss PDF of SIRCIM (LOS)

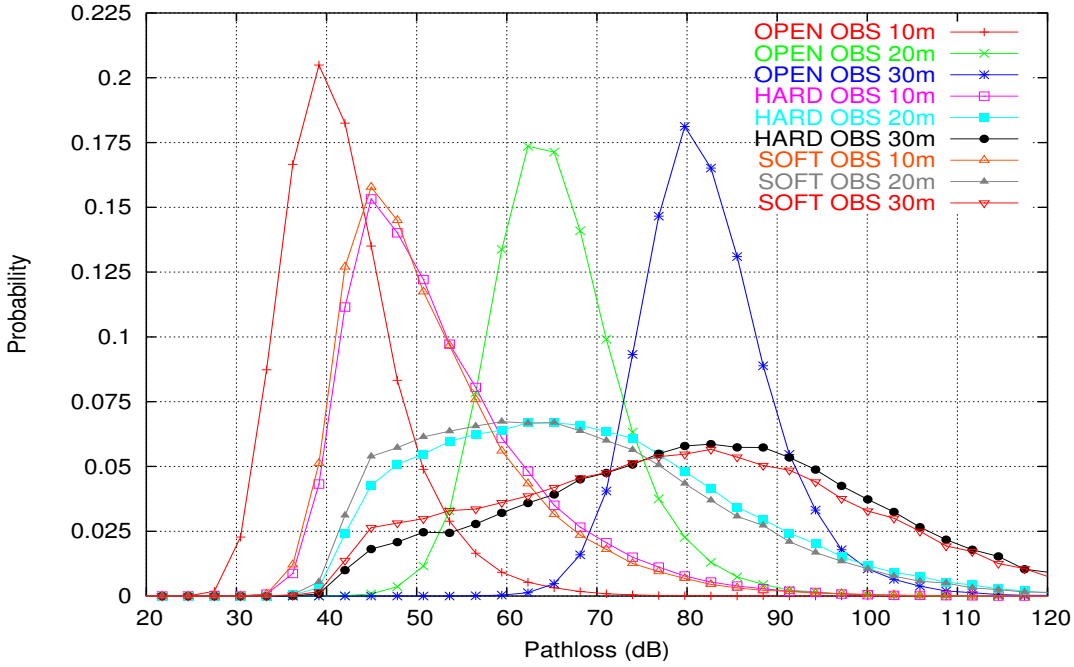


Figure 6.3: Pathloss PDF of SIRCIM (OBS)

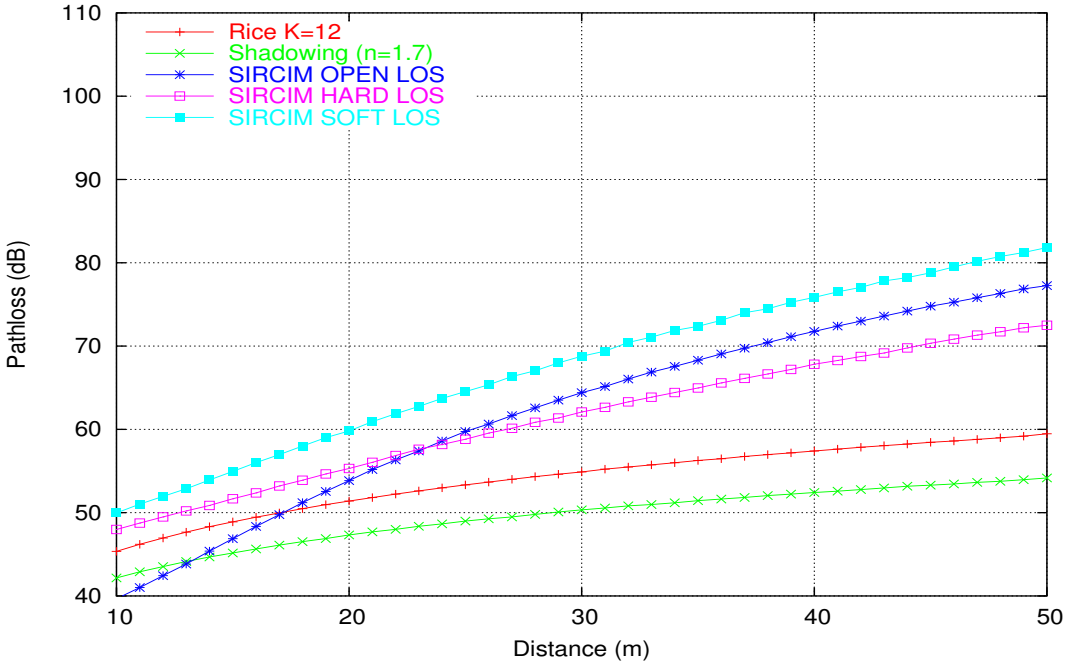


Figure 6.4: Pathloss vs. Distance (LOS)

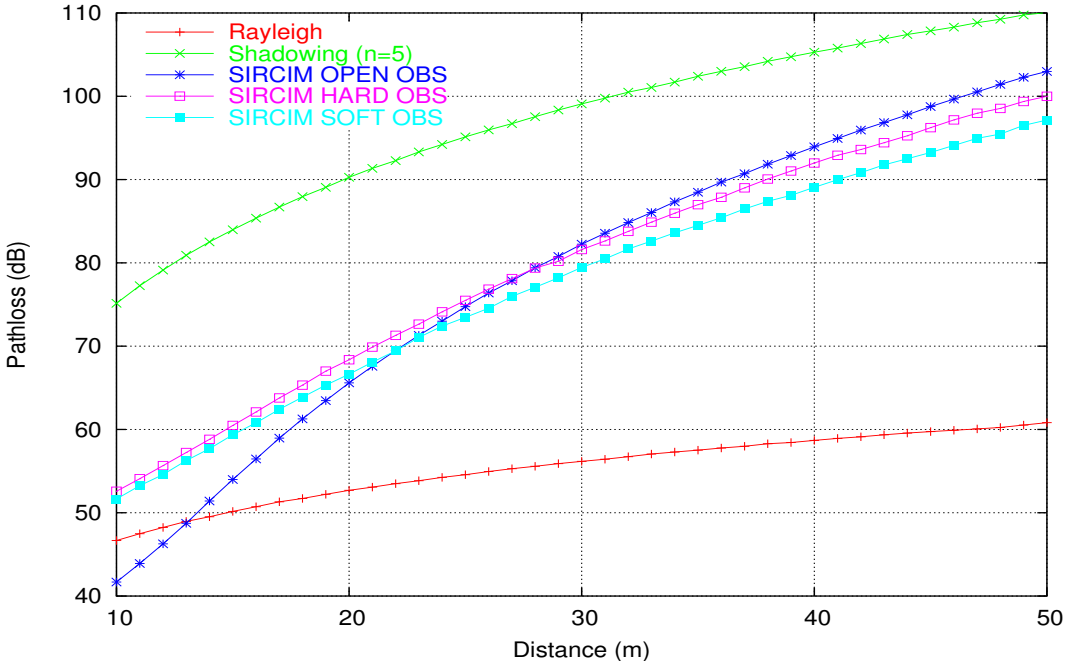


Figure 6.5: Pathloss vs. Distance (OBS)

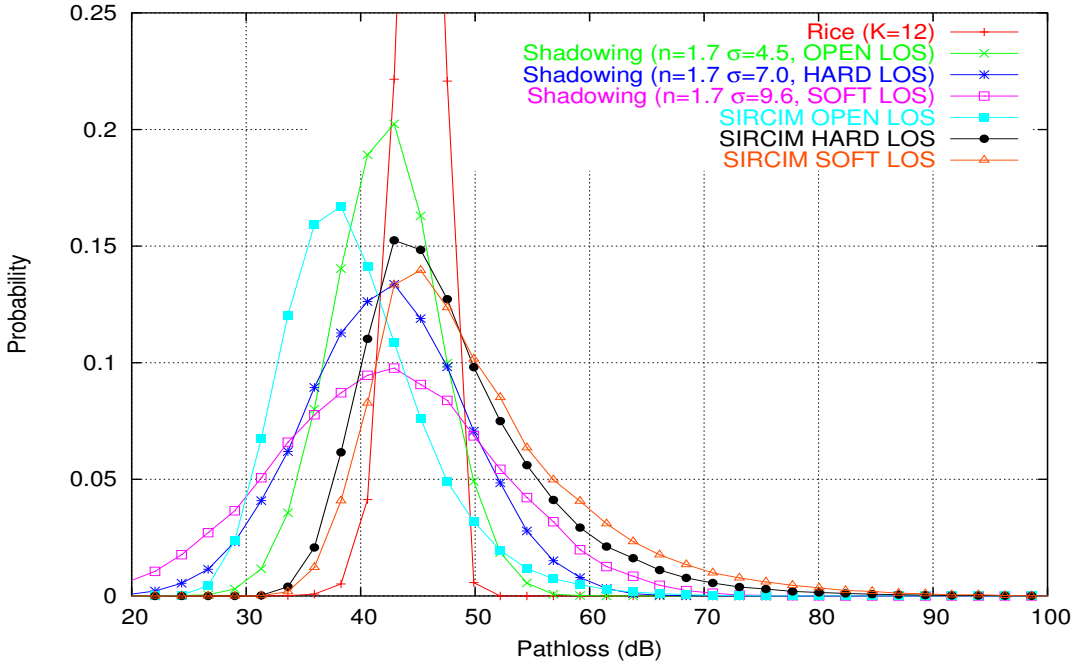


Figure 6.6: Pathloss PDF Comparison (LOS, 10m)

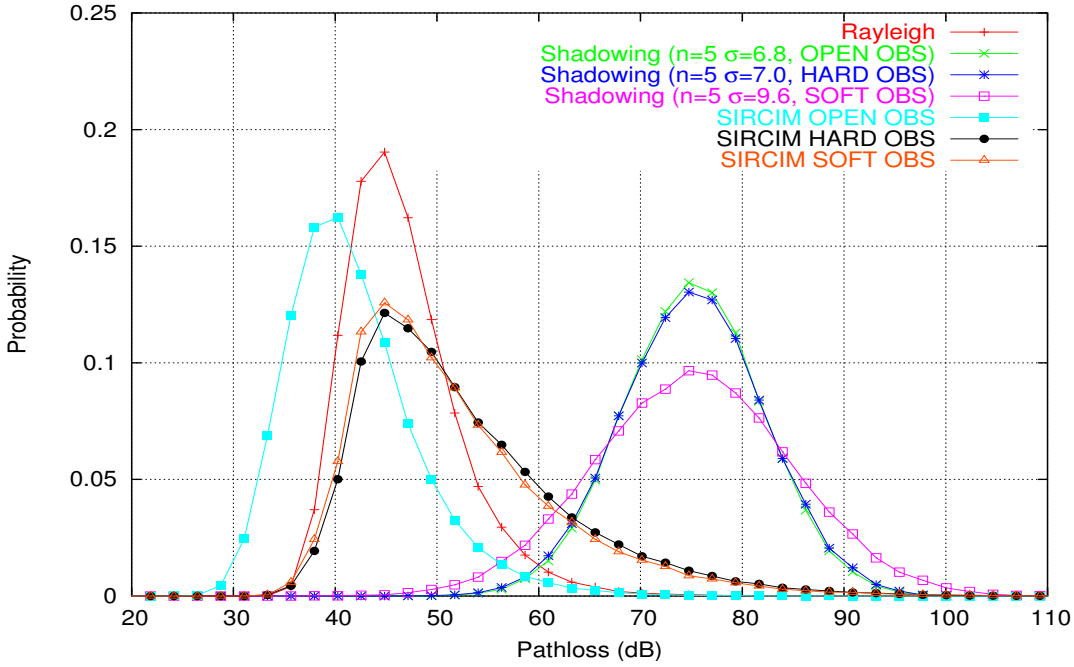


Figure 6.7: Pathloss PDF Comparison (OBS, 10m)

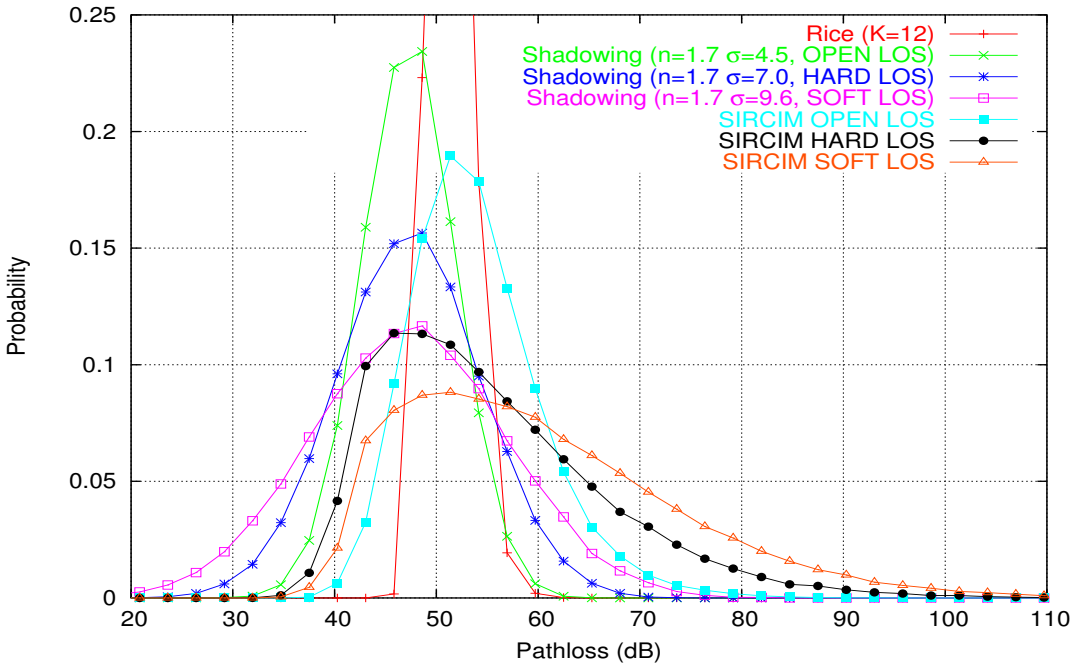


Figure 6.8: Pathloss PDF Comparison (LOS, 20m)

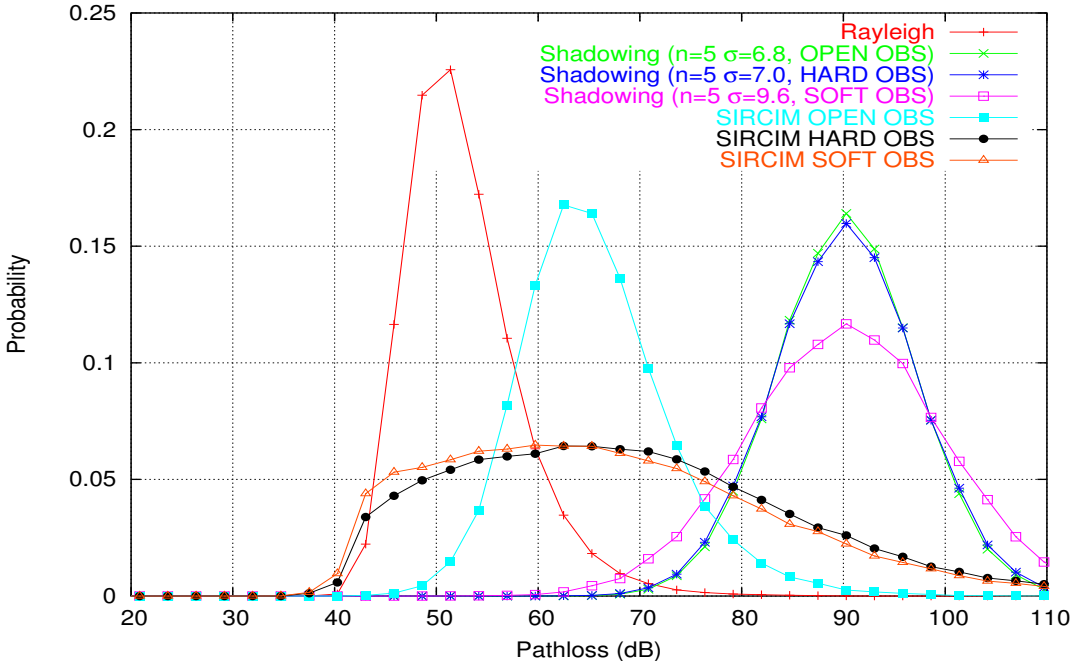


Figure 6.9: Pathloss PDF Comparison (OBS, 20m)

pathloss of the SIRCIM model increases faster with distance than that of the other models, which just indicates that the indoor channel suffers from severe pathloss when the distance increases.

Next, the PDFs of the pathloss of different models will be compared. Fig. 6.6 shows the LOS case with a distance of 10m. The Rice fading with a K-factor of 12 [47] for an indoor channel generates a sharp curve with very small deviation, which is obviously not a good indoor channel model. In the log-normal shadowing model, $n = 1.7$ is for LOS and $\sigma = 4.5, 7, 9.6$ are for the OPEN, HARD and SOFT environments respectively [40, p188]. Because of the same exponent $n$, the log-normal shadowing model has the same mean for the open OPEN, HARD and SOFT environments, which is unrealistic. The other problem of the shadowing model is that although it is true that the deviations of the HARD and SOFT environments are much larger than that of the OPEN environment, an unrealistically small pathloss could be generated with the same probability as a very big one, which is also not true for an indoor channel. However, the SIRCIM model gives a more reasonable distribution that the OPEN environment has relatively smaller mean and deviation than the HARD and SOFT environments. Furthermore, the probability of generating an unrealistically small pathloss is also much smaller than in the log-normal shadowing mode.

In the OBS case as shown in Fig. 6.7, Rayleigh fading provides a more reasonable curve than the Rice fading in the LOS case. However, it can not distinguish the OPEN, HARD and SOFT environments. The log-normal shadowing model with $n = 5$ is for OBS [40, p188], but it seems that the mean pathloss is too big, which is also shown in Fig. 6.5.

Fig. 6.8 and Fig. 6.9 show the same comparison with a distance of 20m, from which we can see that the huge deviation of the SIRCIM model indicates the dynamic characteristics of the indoor channel when the distance is big. Once again, an unrealistically small pathloss is also avoided despite of the big deviation.

### SIRCIM Conversion Conclusion

Since the general SNR-based channel models most widely used do not seem to be adequate for accurate indoor channel simulation, we propose to convert the well-established indoor wireless channel model, SIRCIM, into an SNR-based model. To this end, we derive the total power pathloss from the channel impulse response provided by SIRCIM. Simulation

results show significant differences between the considered state-of-the-art models and the empirically-based SIRCIM, which proves our initial assumption of their inaccuracy. Besides, the new approach also provide ISI information for possibly more accurate BER estimation, which was not available before. The drawback of the proposal is a longer simulation time because of the increased complexity of the model.

### 6.2.2 Physical Layer

The RF transceiver is a nonprogrammable hardware, and its hardware functionality should be modelled by an individual module in the physical layer of the simulation platform. The RF transceiver has two working modes – `SLAVE` mode and `SELF-POLLING` mode. In the `SELF-POLLING` mode, a timer is set to periodically switch the RF transceiver between the states `OFF`, `SETUP`, and `ON`. In the `SLAVE` mode, the RF transceiver has five states – `SLEEP`, `RX_SETUP`, `RX`, `TX_SETUP`, and `TX`, which is controlled by the FPGA or the microcontroller. The RF transceiver can only wake up the FPGA.

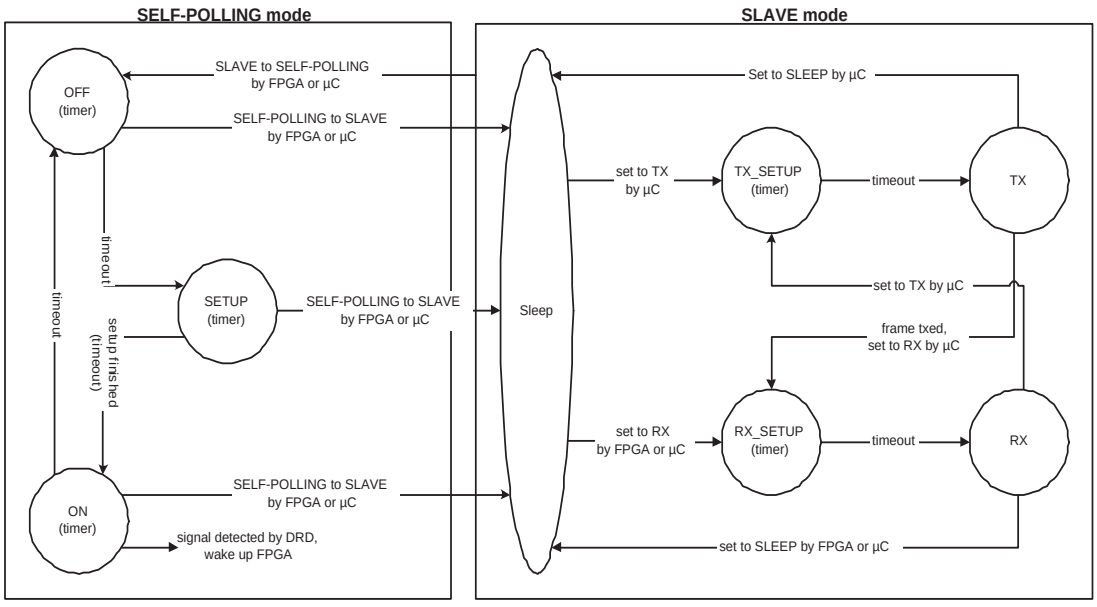


Figure 6.10: State Machine: RF Transceiver

The `SETUP`, `RX_SETUP`, and `TX_SETUP` are used to simulate the state transition delay, e.g. the setup time for the `RX` state is 2.2 ms and for the `TX` state is 1.1 ms. During these

setup states, the RF transceiver can neither receive nor transmit, while only a timer is running until the predefined setup time elapses, and then the RF transceiver is switched to corresponding working state.

How the RF transceiver is switched between different modes and states is illustrated by the state machine as shown in Fig. 6.10.

### 6.2.3 FPGA Module

The state machine for the WUF scheme in the FPGA module is shown in Fig. 6.11. The `OFF` and `SETUP` are hardware states. The `TX`, `RX`, and `RX_WUF` are software states and there is no state transition delay among them.

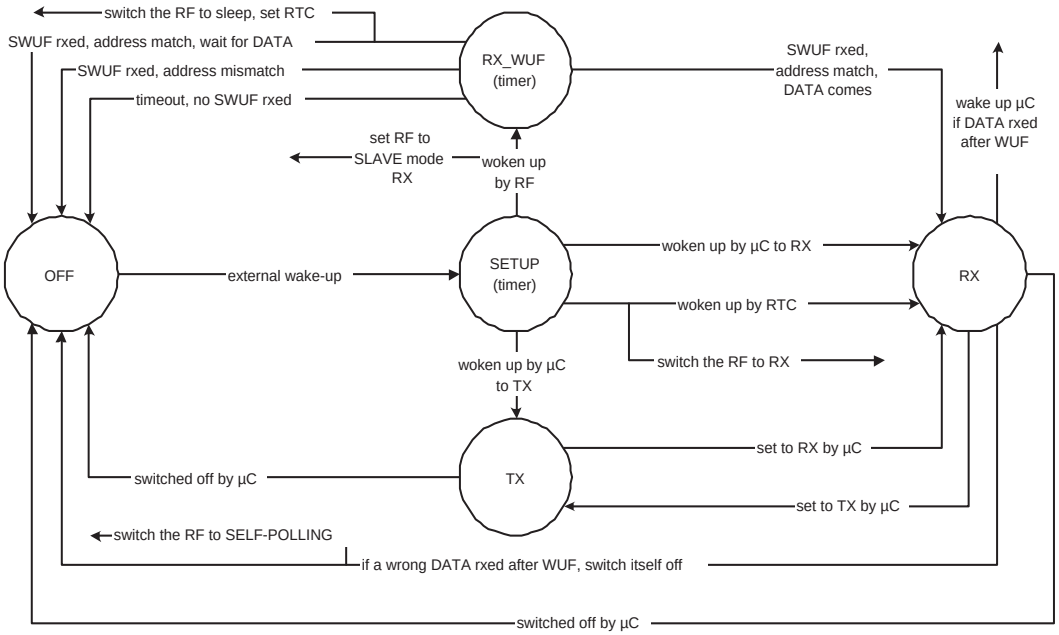


Figure 6.11: State Machine: FPGA

An FPGA in power-down mode can be switched on only by an external wake-up signal. After a setup phase, the FPGA first determines by which component it is woken up – the RF transceiver, the RTC, or the microcontroller. If the FPGA is woken up by the RF transceiver, it goes to `RX_WUF` state, and switches the RF transceiver to `SLAVE` mode `RX` state. If there is no SWUF received or the received SWUF has a wrong destination MAC

address, the FPGA switches the RF transceiver to the sell-polling mode, and switches itself to `OFF` state immediately. If an SWUF with correct destination MAC address is received, the FPGA should calculate the remaining time before the data frame. If there is still enough time left, the FPGA turns the RF transceiver to sleep, sets the RTC, and goes to `OFF` state. If there is no enough time left, the FPGA goes to `RX` state to receive the data frame. If the FPGA is woken up by the RTC, it goes to `RX` state, and switches on the RF transceiver to receive a data frame. If a correct data frame is received after a WUF, the FPGA wakes up the microcontroller to process the data. If a wrong data frame is received after a WUF, the FPGA switches the RF transceiver to self-polling mode, and switches itself off. The FPGA also can be switched between `OFF`, `RX`, and `TX` states by a working microcontroller.

### 6.2.4 MAC Layer

In the MAC layer of the simulation platform, the preamble sampling based schemes and the IEEE 802.15.4 MAC are implemented.

- **The Preamble Sampling Schemes:** The MAC protocol for the preamble sampling schemes is a normal CSMA protocol with exponential backoff. The implementation for a master and a slave is almost the same except that a WUS should be sent before the data frame by a master. The corresponding state machine is shown in Fig. 6.12.

  A sleeping microcontroller can be woken up by the FPGA, the RTC, or a sensor. If the device wants to send a frame, the MAC first does carrier sense to determine whether the channel is busy or idle, i.e. see whether the RSSI is greater than a certain threshold. If the channel is busy, the MAC backs off for a random time period and does the carrier sense again. If the channel is idle, the MAC sends out the frame. If an ACK is required (unicast), the MAC goes to a state waiting for ACK. If no ACK is received after a certain time, the MAC retransmits the frame after another backoff and positive carrier sense. If no ACK is received after a certain times of retry, the MAC stops the transmission and informs the upper layer. If an ACK is received, the MAC goes to a yield state, in which whether the upper layer has more packet is first determined. If there is no more packet, the MAC goes to idle state directly. If there is more packet, the MAC sets a timer and stays at yield state. When the timer expires, the MAC goes to idle and then starts to transmit another frame. Note that

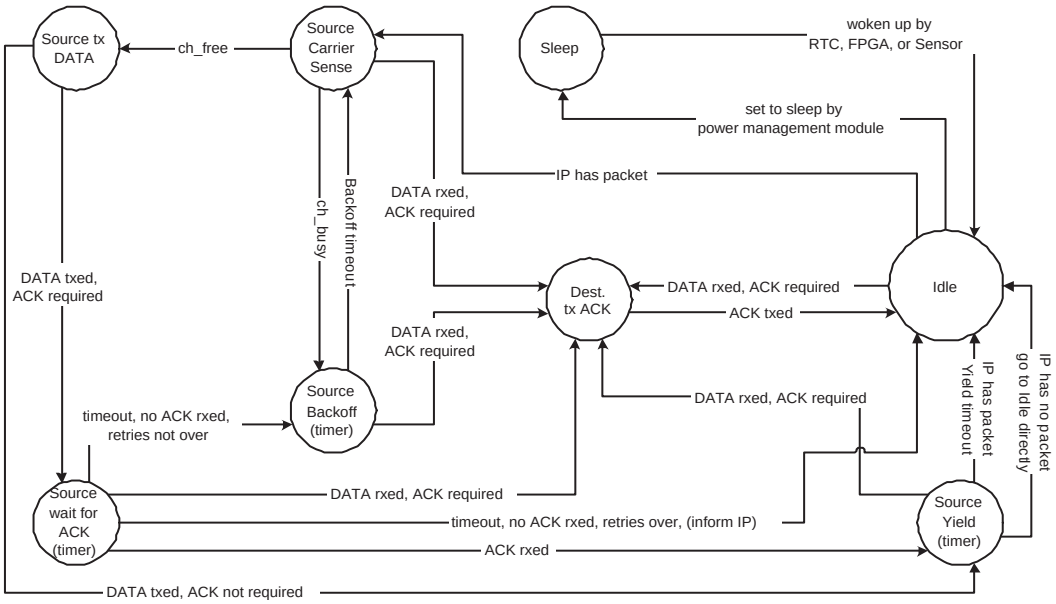


Figure 6.12: State Machine: CSMA

the yield state is for the purpose of fairness. After each successful transmission, the transmitter should let the channel idle for some time to give other devices chance to send frame. If an ACK is not required (broadcast), the MAC directly goes to yield state. If a unicast frame is received during idle, carrier sense, backoff, wait for ACK, or yield states, the MAC goes to the state of answering ACK.

- **The IEEE 802.15.4 MAC:** The IEEE 802.15.4 MAC protocol is a combination of the TDMA and the CSMA random access, supporting low-rate time critical streaming in a master-slave topology, so that the implementation of this protocol is much more complicated than a simple CSMA protocol. Since the master and the slaves in the IEEE 802.15.4 MAC play very different roles, their state machines are also quite different from each other, as shown in Fig. 6.13 and Fig. 6.14, respectively.

  In the master, the initial state of the MAC is `IDLE`. If power-saving should be supported, the MAC first sends a beacon indicating the superframe structure as described in Sec. 3.3.1. Then, the MAC goes to Eclipse I, where both uplink and downlink packets are transmitted using CSMA in the contention access period. The master MAC first waits for frame from slave (`WF_FRAME`). If a data frame is received, the MAC answers with an ACK (`TX_ACK`). If a data request command is received, the

MAC sends the corresponding data frame to the slave (TX_DATA), and then waits for an ACK (WF_ACK). After the time of the contention access period has elapsed, the MAC goes to Eclipse II for transmitting packet in the contention free period, where the uplink and downlink packets are sent in guaranteed time slot(s) without CSMA. Finally, the MAC goes back to idle state.

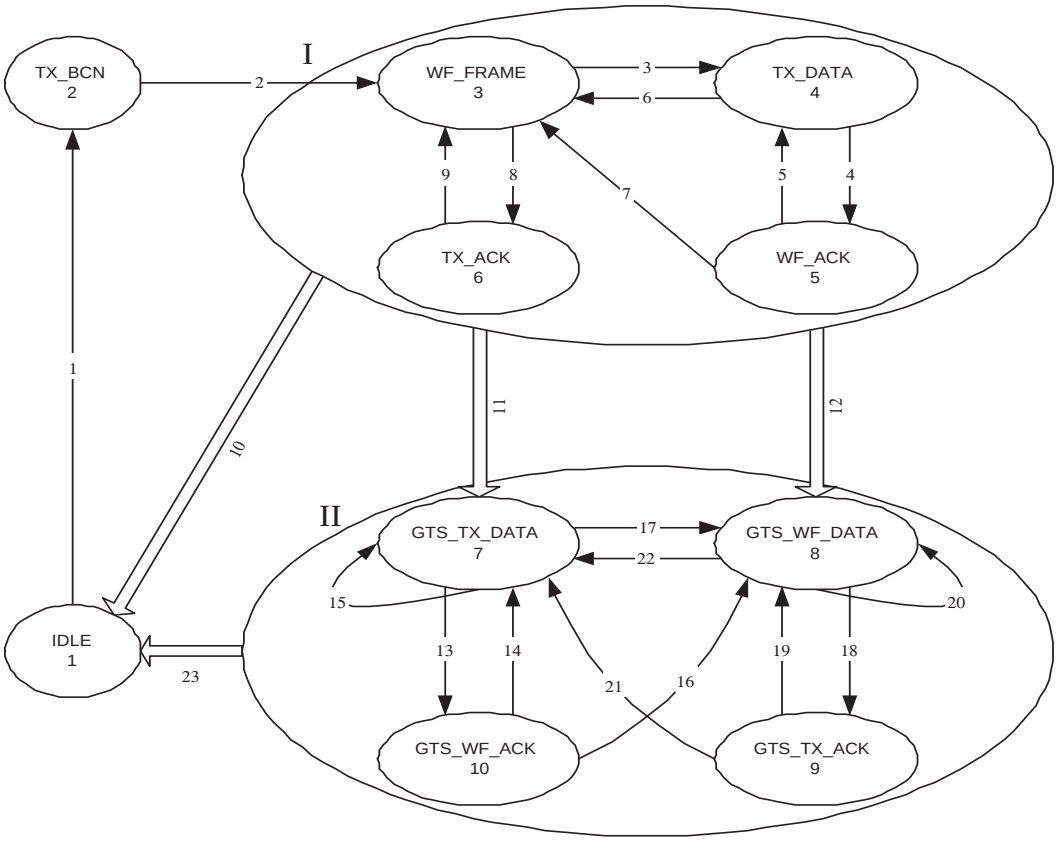


Figure 6.13: State Machine: IEEE 802.15.4 MAC Master

In the slave, the MAC first waits for a beacon (WF_BCN) to get the information of the superframe structure and pending packet. If there is pending downlink packet(s) for the slave or the slave has uplink packet(s) for the master, the MAC goes to Eclipse I for data transmission in the contention access period, where the MAC should do carrier sense (CS) to compete the channel for sending data frame (TX_DATA) or data request command (TX_CMD). After the time of the contention access period has elapsed, the MAC goes to Eclipse II for transmitting time critical packet(s) in the

contention free period. Finally, the MAC enters the sleep state for the rest of the beacon interval until the next beacon comes. For more detailed description of these two state machines, please refer to [48].

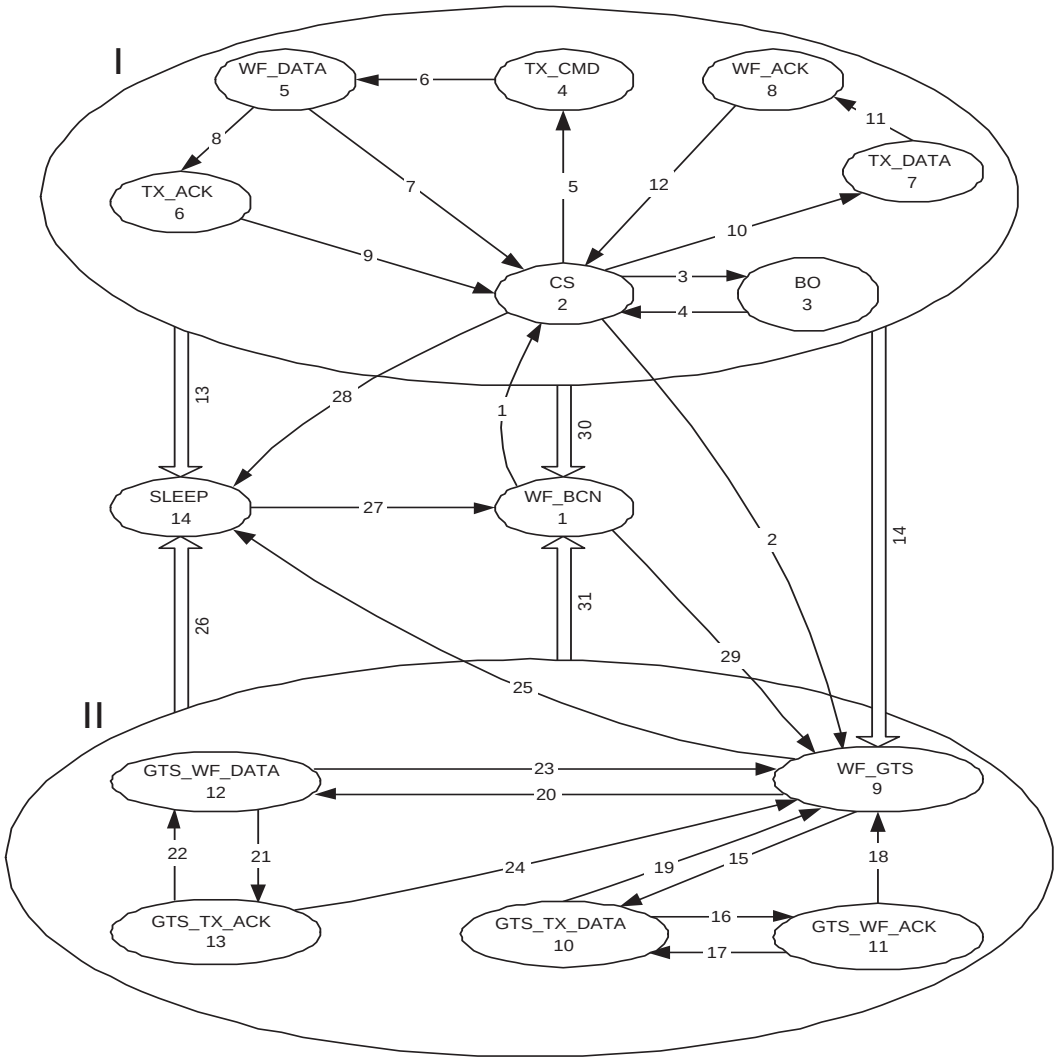


Figure 6.14: State Machine: IEEE 802.15.4 MAC Slave

Note that, from the figures, the complexity of the state machine of the slave is even higher than that of the master, which seems to be unreasonable. However, the master needs definitely more hardware support than the slave, because it has to cope with multiple uplink and downlink frames for a plurality of slaves, which needs much more buffer and more control instructions.

### 6.2.5 Application Layer

The choice of the application traffic model is another very important issue for precise simulation. In general, two parameters must be taken into consideration for an appropriate traffic model: the size of the packets and the arrival of the packets.

#### Packet Size

In the Sindrion network, a packet comprises the following parts:

- **Application Layer:** From 2 up to 500 bytes. However, a large portion of the packets are short control packets in the range of 10 to 30 bytes.
- **Transport Layer:** 20 bytes for TCP header, and 8 bytes for UDP header.
- **IP Layer:** 20 bytes for IP header.
- **MAC Layer:** 22 bytes for MAC header including two 8-byte long MAC addresses, see Fig. C.1.
- **ECC and Modulation Coding:** a coding ratio of 1.5 is assumed for the combination of ECC and modulation coding for the data frame.
- **PHY Layer:** 1 byte is assumed for the start of frame (SOF), and 2 bytes are assumed for the sync field.

For instance, if an application packet of 20 bytes is sent via UDP, the overall length of the frame is $(20 + 8 + 20 + 22) \times 1.5 + 3 = 108$ Bytes. This packet length will be used in most analytical computations and simulations, if it is not stated otherwise.

#### Packet Arrival

Regarding the arrival of the packets, there is so far no general model for WSNs. A possible choice is the Poisson process [49]. It has been used to model telephone traffic and to evaluate the performance of computer networks. It is also used in [35, 36, 38] to evaluate the WiseMAC. Therefore, the Poisson process packet arrival model is adopted in this dissertation to simulate the application traffic.

In the Poisson process [50], the probability of receiving $k$ packets within an interval of $T$ is given by

$$p(k) = (\lambda T)^k \frac{e^{-\lambda T}}{k!}, \tag{6.18}$$

where $\lambda$ is the so-called rate parameter, i.e. the expected number of arrivals or packets per unit of time, given by

$$\lambda = E(k)/T. \tag{6.19}$$

The interarrival time $\tau$ between two packets is a positive random variable, which obeys exponential distribution. Its PDF, and Cumulative Distribution Function (CDF) are given by

$$p(\tau) = \lambda e^{-\lambda \tau}, \tau \geq 0, \tag{6.20}$$

and

$$f(\tau) = 1 - e^{-\lambda \tau}. \tag{6.21}$$

In concordance with Equ. 6.19, the expected interarrival time is given by

$$E(\tau) = \int_0^\infty \tau \, p(\tau) \, d\tau = 1/\lambda. \tag{6.22}$$

The value of $\lambda$ depends on the application. It is quite small for most WSNs. A typical range could be less than $0.001$ packet/s for each sensor device.

# Chapter 7

# Simulation Results and Analysis

This chapter will show and analyze the simulation results based on the simulation models described in Chp. 6.

## 7.1 Downlink Traffic

First, only downlink packets are sent in a star topology with one master. The purpose of this simulation is to prove the correctness of the analytical computation.

### 7.1.1 Battery Lifetime

Fig. 7.1 shows the simulated battery lifetime of the basic preamble sampling schemes including the WUP, Rep, and WUF schemes, and the IEEE 802.15.4 MAC compared to the analytical results that have been presented in Sec. 5.2.1. We can see that the simulation results of the basic preamble sampling schemes are almost exactly the same as the analytical results.

However, the IEEE 802.15.4 MAC shows quite shorter battery lifetime in simulation than in analytical computation when the traffic is high. This is because the analytical computation does not includes the case that multiple frames are sent within one beacon interval, as mentioned in Sec. A.1. When a master indicates more than one frame to multiple slaves in a single beacon, the destination slaves have to contend the channel using CSMA. In this case, first, collisions may happen; second, the one that does not get the channel has to wait.

Both of the two effects waste energy. The higher is the traffic, the higher is the probability of sending multiple frames within one beacon interval, so the more energy is consumed.

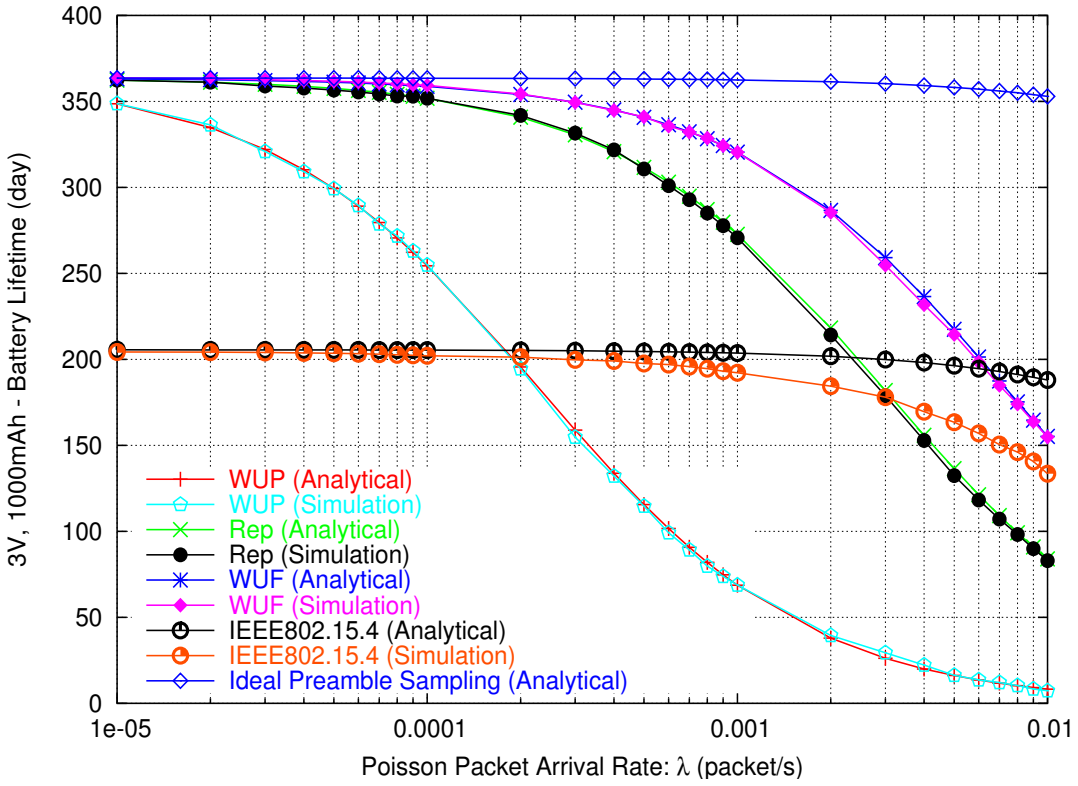


Figure 7.1: Battery Lifetime of Basic Preamble Sampling Schemes
(100 slaves, Unicast, Downlink, Simulation)

Fig. 7.2 shows the same comparison for the WiseMAC+WUP/Rep/WUF schemes. We can see that the shapes of the simulated curves are the same as the analytical ones. The only difference is that the simulated values are slightly smaller than the computed values when the traffic is high. This is caused partially by packet loss due to channel disturbance and collision. However, the main reason is as follows. As mentioned in Sec. 3.4, the first data frame in the WiseMAC should use a long WUS, because the master does not have the sampling schedule of the destination yet. However, the long WUS wakes up all the slaves wasting energy due to overhearing. The analytical computation does not take this into consideration, while the simulation does. If the simulation time is long enough, the extra energy caused by the first data frame could be negligible. However, since the simulation is very time consuming, the shown simulation results are based on simulation time of one

day, causing the inaccuracy of the simulation results. Note that the simulated results are the same as the computed ones when the traffic is low. This is because that all data frames should use long WUS in low traffic load, so the first data frame is the same as the others consuming no extra energy.

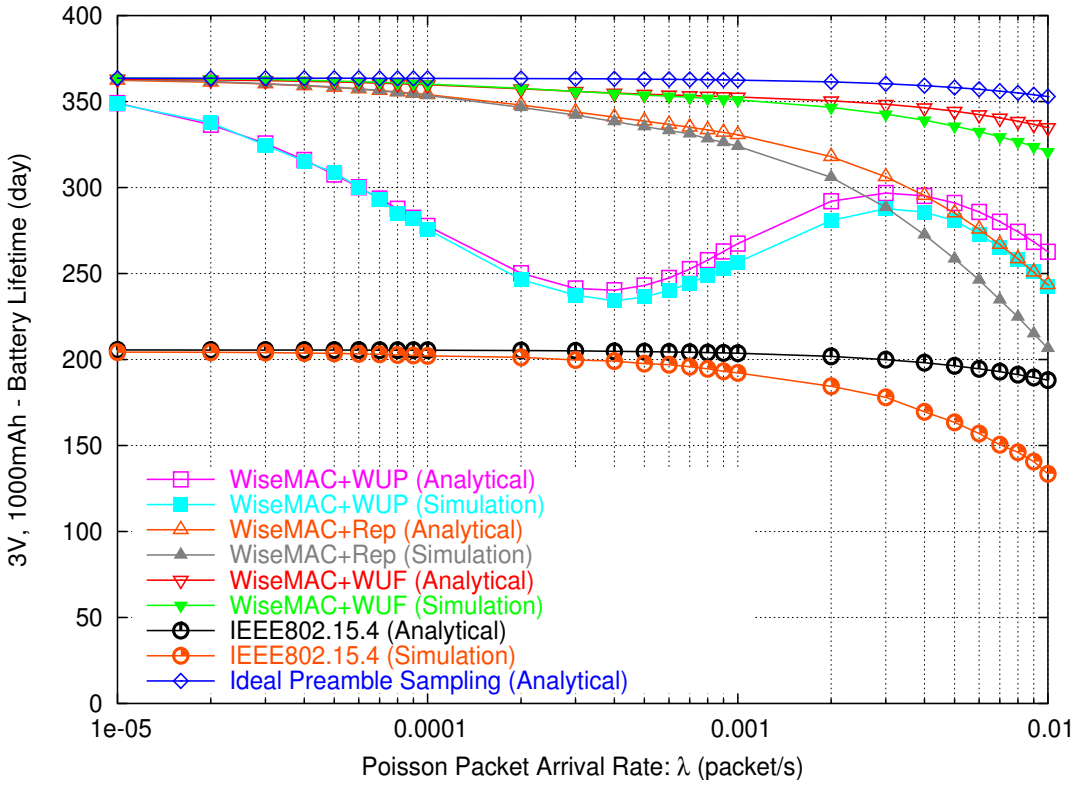


Figure 7.2: Battery Lifetime of WiseMAC Schemes
(100 slaves, Unicast, Downlink, Simulation)

Another point should be noticed is that the difference between the simulation and analytical results of the WiseMAC+Rep is bigger than that of the other two schemes. The reason is that only single data rate can be used for data frame, WUS, and ACK in the data repetition scheme, so that all of the frames can wake up overhearers consuming more energy. However, the analytical computation does not consider the overhearing caused by ACK for the reason of simplicity as described in Sec. A.6, while the simulation does.

Fig. 7.2 also shows that the WiseMAC+WUF scheme achieves a battery lifetime gain of up to $60\%$ compared to the WiseMAC+Rep, and of up to $140\%$ compared to the IEEE 802.15.4 MAC in a network with 100 slaves and only downlink traffic.

### 7.1.2 Channel Occupation

Fig. 7.3 shows the simulated channel occupation results compared to the analytical results, where the basic preamble sampling denotes the WUP, Rep, and WUF schemes, and the WiseMAC denotes the WiseMAC+WUP/Rep/WUF.

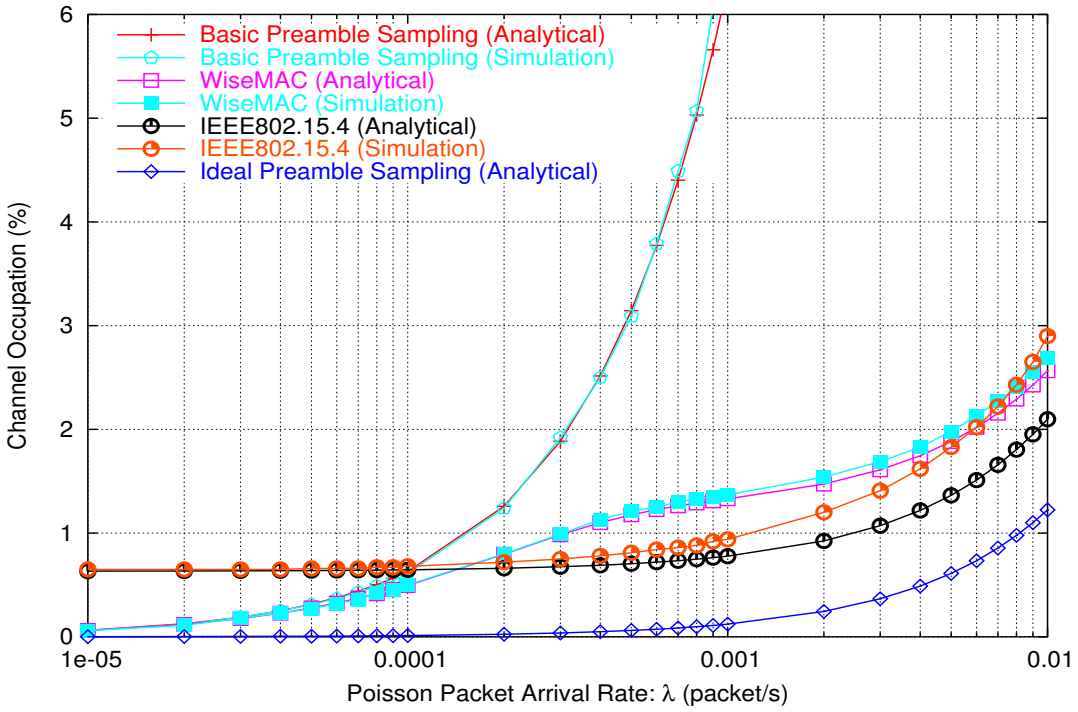


Figure 7.3: Channel Occupation of All the Schemes
(100 slaves, Unicast, Downlink, Simulation)

We can see that the basic preamble sampling scheme still shows the identity between the simulated and computed results. For the WiseMAC, the simulated channel occupation is slightly bigger than the analytical result. It can also be explained by the long WUS caused by the first data packet. The IEEE 802.15.4 MAC occupies more channel in the simulation than in the computation, because collisions may happen if multiple frames are sent within single beacon interval.

From the simulation results, we can see that the channel occupation of the IEEE 802.15.4 MAC is even higher than that of the WiseMAC when the traffic is very high. So with a number of slaves of 100 or less, the IEEE 802.15.4 MAC has actually no advantage in the sense of channel capacity.

### 7.1.3 Response Time

The response time comparison between the simulation and the analytical computation is shown in Fig. 7.4. The simulated results of the basic preamble sampling schemes do not exactly match the computed ones. It is because when a new packet is available in the network layer ready for MAC, it is possible that the transmission of the last packet has not been finished yet. As a result, the new packet has to wait. This phenomenon occurs frequently when traffic is high, so that the response time increases. The same effect also causes the curve of the WiseMAC to go up when the traffic is very high.

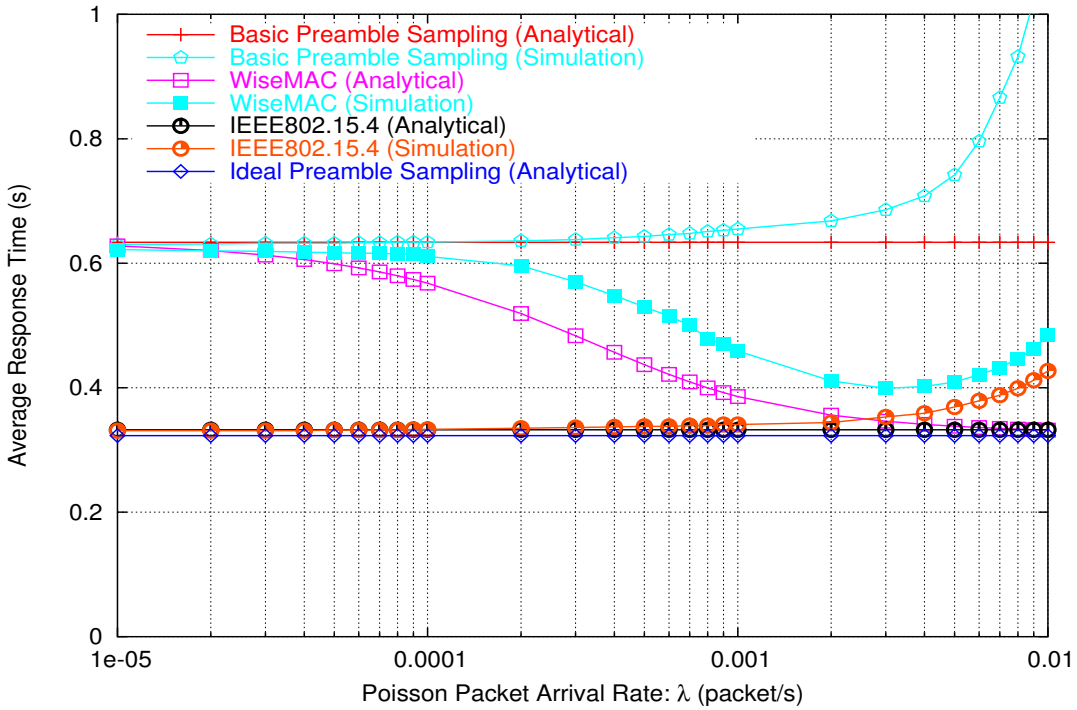


Figure 7.4: Response Time of All the Schemes
(100 slaves, Unicast, Downlink, Simulation)

The response time of the IEEE 802.15.4 MAC also increases in high traffic. However, it cannot be simply explained by the wait time of the network packet, because the MAC layer of the master can indicate multiple packets within a single beacon. So the increased response time is induced by the MAC layer channel contention, when multiple frames are sent within one beacon interval.

## 7.2 Uplink and Downlink Traffic

Since a master-slave WSN has not only downlink but also uplink traffic, a more realistic simulation should also include the uplink traffic. Note that the uplink traffic of the preamble sampling schemes is different from the downlink traffic, because it needs no WUS since the master is always powered-on. We assume that each slave sends packets to the master also with a packet arrival rate $\lambda$. I.e. if the number of the slave is 100, the overall traffic on the network is $2 \cdot 100 \cdot \lambda$.

### 7.2.1 Battery Lifetime

Fig. 7.5 shows the battery lifetimes of the basic preamble sampling schemes in the case with bi-directional traffic and in the case with only downlink traffic. From the figure, we can see the influences on the different schemes caused by the uplink traffic.

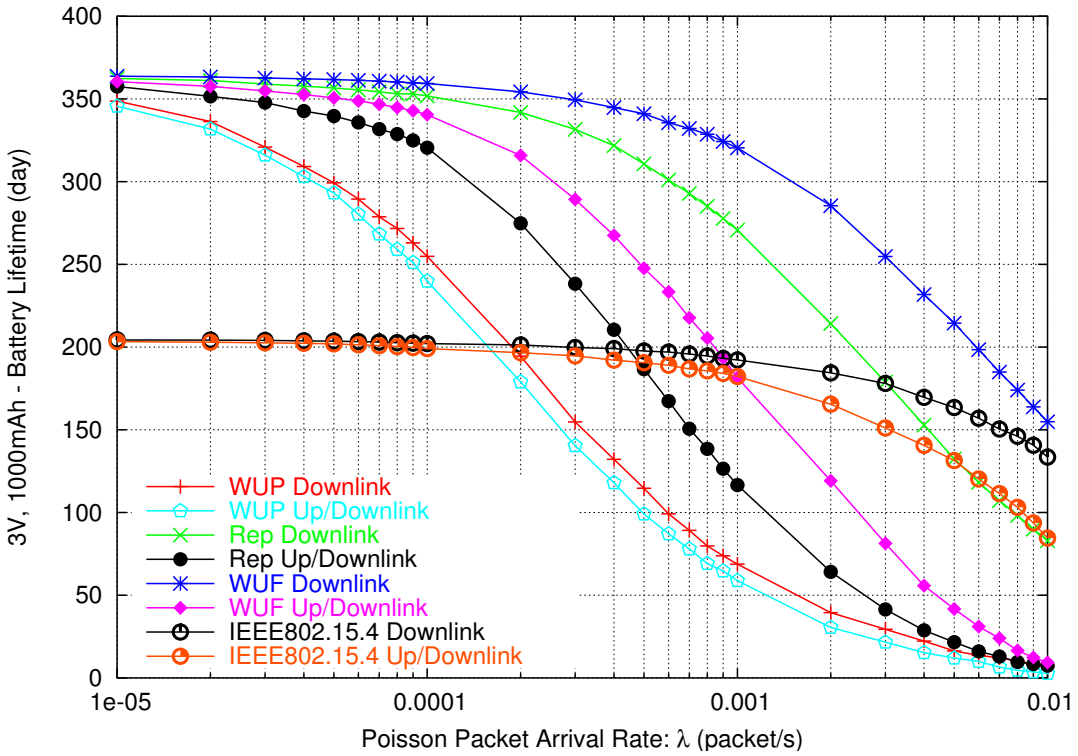


Figure 7.5: Battery Lifetime of Basic Preamble Sampling Schemes
(100 slaves, Unicast, Up/Downlink, Simulation)

Since the uplink packet does not need WUS, the energy consumed by the uplink traffic includes only the energy for transmitting data frames and for receiving the ACKs, which should not be very significant. However, as shown in Fig. 7.5, the degradation caused by the uplink traffic of the Rep scheme and the WUF scheme is significant. It can be explained as follows. When a slave wants to send an uplink data frame, it is possible that a downlink data frame with a long WUS is being transmitted. As a result, the slave has to wait until the end of the downlink transmission. This unnecessary waiting time is about half of the long WUS on the average, wasting a lot of energy. The higher the traffic, the higher the probability that this uplink unnecessary waiting happens, so the more is the energy wasted. When the traffic is very high, the channel is almost saturated by the long WUS, so that there exists a lot of uplink unnecessary waiting as well as collisions. As a result, the battery lifetime is very short. Actually, the Rep scheme and the WUP scheme cannot work in such a high bi-directional traffic.

However, the battery lifetime of the WUP scheme is not decreased so much by the uplink traffic. The reason is as follows. The uplink unnecessary waiting does not occur for each uplink packet, and it affects only the slave that is sending the frame. However, in the WUP scheme, each downlink frame with long WUS causes unnecessary waiting in all slaves. Therefore, the uplink unnecessary waiting is minor compared to the downlink unnecessary waiting.

The battery lifetime of the IEEE 802.15.4 MAC is also significantly shortened by the uplink traffic. As mentioned in Sec. 3.3.1, in the IEEE 802.15.4 MAC, downlink communication (DATA-ACK) and uplink communication (REQUEST-DATA-ACK) have no principle difference, so their energy consumption is similar. Therefore, adding the uplink traffic to the simulation just roughly doubles the traffic load. When the overall traffic is low, the probability of transmitting multiple frames within one beacon interval is very low, so the doubled traffic does not decrease the battery lifetime much because each individual frame costs little energy compared to the periodical beacon. However, when the traffic is high, multiple frames within one beacon interval occurs frequently, in which channel contention as well as collisions waste a lot of energy. As a result, the battery lifetime of the IEEE 802.15.4 MAC is further decreased, especially when the traffic is high.

Fig. 7.6 shows how the battery lifetime of the WiseMAC+WUP/Rep/WUF is affected by the uplink traffic. The degradation caused by the uplink traffic is not so big as in the basic preamble sampling case. The reason is that shortened WUS occupies much less

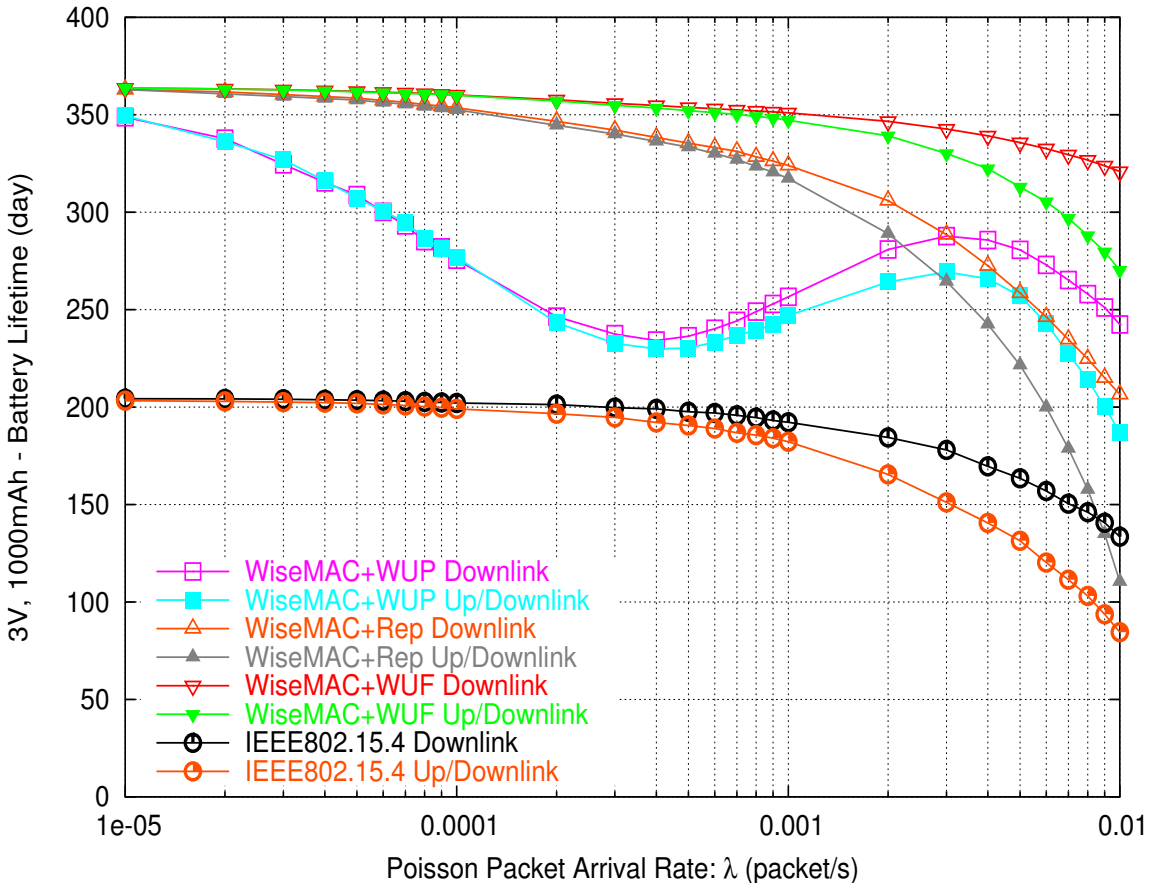


Figure 7.6: Battery Lifetime of WiseMAC Schemes
(100 slaves, Unicast, Up/Downlink, Simulation)

channel causing much fewer uplink unnecessary waiting.

Note that the WiseMAC+Rep is more negatively affected by the uplink traffic than the WiseMAC+WUP and the WiseMAC+WUF. This is because the uplink data frame as well as the corresponding ACK can also wake up overhearers since only one datarate is assigned to all types of frames.

Fig. 7.6 also compares the battery lifetime of the different schemes with bi-directional traffic. The result shows that the WiseMAC+WUF scheme achieves a battery lifetime gain of up to $140\%$ compared to the WiseMAC+Rep, and of up to $220\%$ compared to the IEEE 802.15.4 MAC a in network with 100 slaves.

### 7.2.2 Channel Occupation

Fig. 7.7 shows the influence on the channel occupation caused by the uplink traffic.

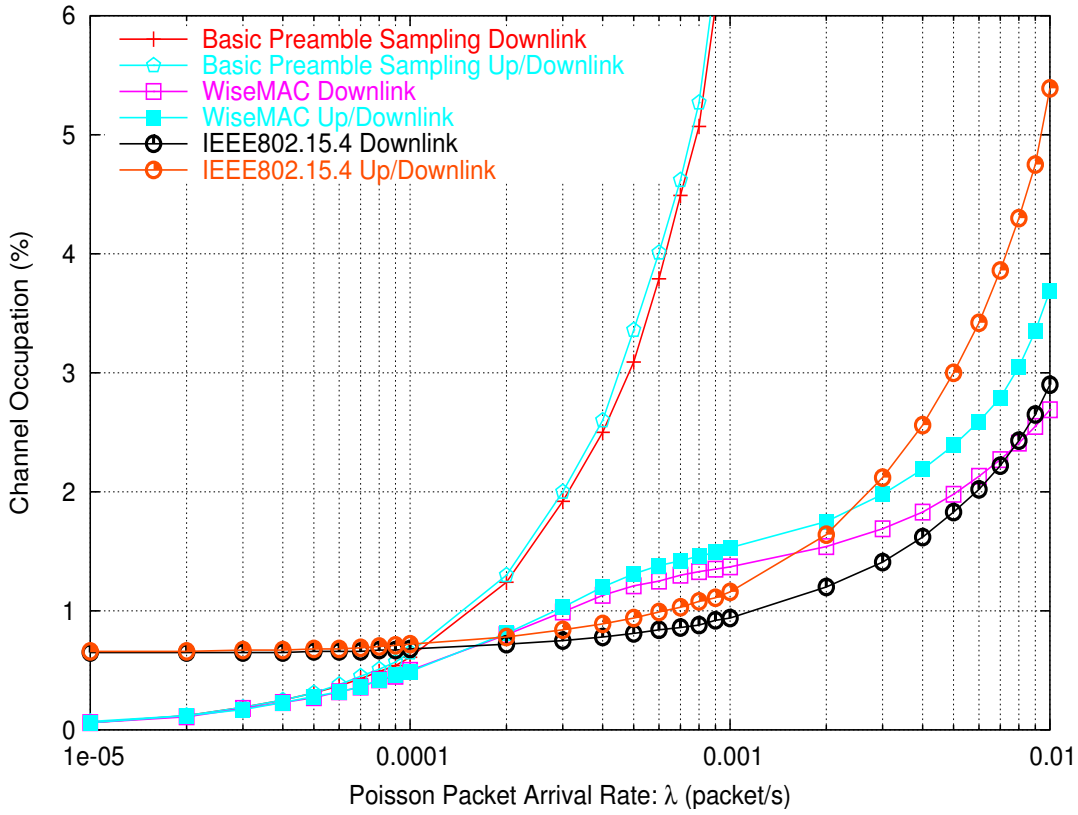


Figure 7.7: Channel Occupation of All the Schemes
(100 slaves, Unicast, Up/Downlink, Simulation)

The channel occupation of the basic preamble sampling schemes for the bi-directional traffic is just slightly higher than that for downlink only traffic. The reason is that the uplink traffic without long WUS is too short compared to the downlink traffic with long WUS, so that its channel occupation can almost be neglected.

Note that collisions in the bi-directional traffic cannot significantly increase the channel occupation. The reason is as follows. First, there are few collisions between the downlink and the uplink frame, because the master and slaves can hear each other. Second, if collision happens, the uplink frame collides most likely with the downlink WUS instead of

the data frame, which causes only a retransmission of the uplink frame, but not a downlink frame. Third, the collision between uplink frames from different slaves happens with a little bit higher probability, causing the retransmission of the both uplink frames. All in all, there are only small number of retransmissions for short uplink frames and very few retransmission for long downlink frames, which causes negligible additional channel occupation.

For the WiseMAC, the channel occupation caused by the uplink traffic cannot be neglected because the uplink traffic is not so short compared to the downlink traffic with shortened WUS.

In the IEEE 802.15.4 MAC, the uplink traffic occupies just slightly less channel than the downlink traffic because uplink traffic has no data request command. As a result, adding the uplink traffic nearly doubles the overall traffic, so that the channel occupation is increased a lot.

Fig. 7.7 also compares the channel occupation of the WiseMAC and the IEEE 802.15.4 MAC with bi-directional traffic, showing that the IEEE 802.15.4 MAC has no advantage of higher channel capacity anymore.

### 7.2.3 Response Time

The average response time for bi-directionally traffic is shown in Fig. 7.8.

The response time of uplink packets for the preamble sampling based schemes is ideally just the transmission time of the DATA and ACK frame because of the absence of the WUS. However, both the uplink and downlink average response times of the basic preamble sampling schemes increase drastically when the traffic is very high because the almost saturated channel causes a lot of waiting and collisions. It again shows that the basic preamble sampling scheme works only for low traffic networks.

For the WiseMAC, first the uplink response time keeps within a very small range even for very high traffic; second the downlink response time is almost the same as the corresponding curve for the downlink only traffic as shown in Fig. 7.4. Therefore, the average response time of the WiseMAC is not significantly affected by the bi-directional traffic because the low channel occupation induces few waiting and collisions.

In the IEEE 802.15.4 MAC, the average response time of the uplink traffic is slightly shorter than that of the downlink traffic, because the uplink traffic needs no data request

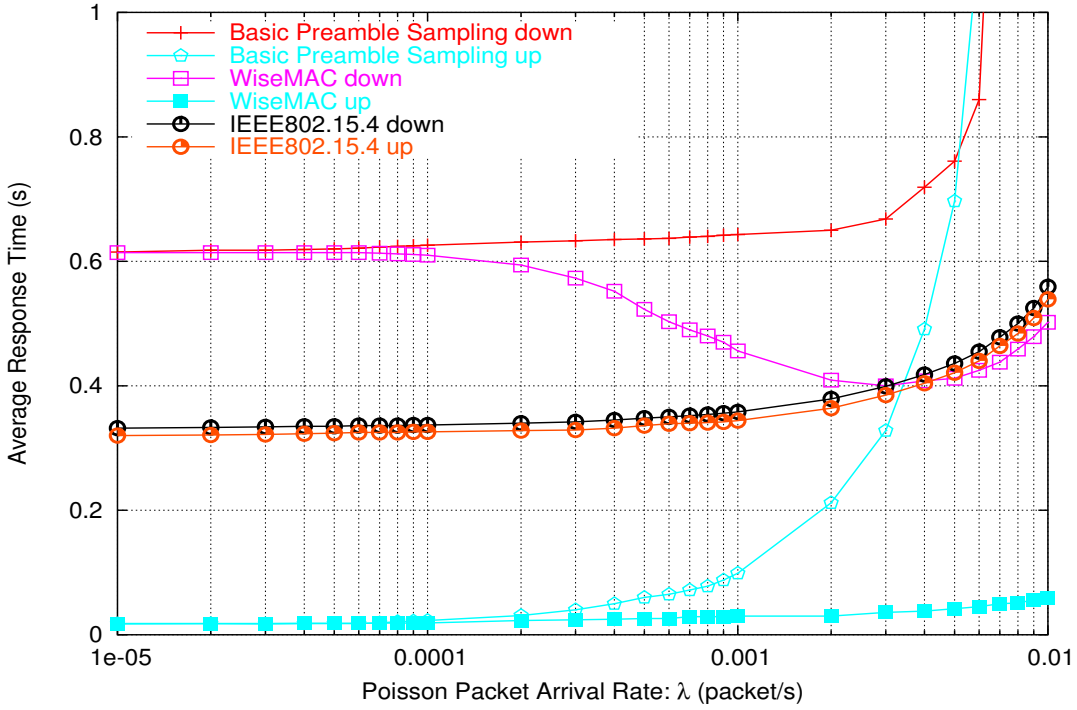


Figure 7.8: Response Time of All the Schemes
(100 slaves, Unicast, Up/Downlink, Simulation)

command. Both the average response times of the uplink and downlink increase slowly with the traffic, because multiple frames within one beacon interval causes more waiting and collisions.

Fig. 7.8 also shows that the average downlink response time of the IEEE 802.15.4 MAC has actually no advantage compared to the WiseMAC in a high traffic load. Furthermore, the WiseMAC has a much shorter uplink response time than the IEEE 802.15.4 MAC.

# Chapter 8

# Conclusion

This dissertation addresses the topic of ultra low-power MAC layer duty cycle scheduling schemes for low-cost and low-traffic wireless sensor networks.

Since large number of wireless sensor devices in a wireless sensor network are battery powered, power consumption becomes one of the most important design issues. To this end, the MAC layer duty cycle scheduling is supposed to be the most efficient way to enable the RF transceiver to work with a very low duty cycle, which has been proved to be able to save power by a factor of more than 50.

A conventional MAC layer duty cycle scheduling scheme is the preamble sampling scheme, which uses a wake-up-preamble that is sent in front of the actual data. It allows a very low duty cycle of less than $1\%$ to be achieved for a receiver. However, the long wake-up-preamble used for transmission causes too long unnecessary waiting time for both the destination and overhearing devices, consuming a large amount of energy. It makes the preamble sampling scheme practically unusable.

To solve the problem, this dissertation first proposes to use a data rate detection function instead of only the received signal strength indicator as the sampling criteria, and to use different data rates for the wake-up-signal and other normal frames, respectively. As a result, a device cannot be woken up by noise, interference, and normal frames, which drastically reduces the probability of overhearing and correspondingly decreases the power consumption.

Second, this dissertation proposes to send a wake-up-frame (WUF), which contains the transmitter's schedule, instead of the meaningless wake-up-preamble to minimize the

unnecessary waiting time, which drastically decreases the system power consumption.

However, the WUF scheme cannot solve the channel inefficiency problem of the preamble sampling scheme due to the long wake-up-signal sent by the transmitter. As a result, it is only proper for low-traffic wireless sensor networks.

To this end, another new approach is proposed in this dissertation to combine the WUF scheme and another preamble-sampling-based scheme, called WiseMAC. The WiseMAC enables a loose synchronization between the transmitter and the receiver, so that the transmitter can roughly know the schedule of the receiver and use a much shorter wake-up-signal, which reduces the power as well as the channel occupation. The WUF scheme and the WiseMAC use two complementary methods to optimize the preamble sampling scheme respectively, so a combination of them is supposed to be a further optimized solution.

To evaluate the proposed WUF scheme and the combination of the WUF and the WiseMAC (WiseMAC+WUF), analytical computations and simulations have been conducted to estimate their power consumption, channel occupation, and response time in a master-slave star topology. From the analytical and simulation results, the following conclusions can be made.

For a small scale network of around 10 nodes with a very low overall traffic load of less than $0.01$ packet/s, where the idea of the WiseMAC takes no effect, the WUF scheme alone works much better than the standardized IEEE 802.15.4 low-power MAC in the sense of power-saving and channel efficiency.

For a moderate scale network of around 100 nodes with a high overall traffic load of up to $1$ packet/s, the proposed WiseMAC+WUF scheme shows more than $100\%$ battery lifetime gain compared to the original WiseMAC protocol and the IEEE 802.15.4 MAC, while it still keeps similar channel efficiency as the IEEE 802.15.4 MAC.

For a large scale network of around 1000 nodes with a high overall traffic load of up to $1$ packet/s, the WiseMAC+WUF scheme still gives significant battery lifetime gain compared to the IEEE 802.15.4 MAC. However, the channel occupation of the WiseMAC+WUF is much higher than that of the IEEE 802.15.4 MAC in this case.

Another important point is that the IEEE 802.15.4 MAC supports power-saving only in star topology with one master, which highly limits its range of usage. However, the WUF scheme and the WiseMAC+WUF are not restricted by any topology.

# Appendix A

# Analytical Computation Details

Note that the meanings and values of parameters used in the analytical computation are listed in Tab. B.1.

To simplify the computation, the power consumption differences between the sleep and working mode of different components are defined by

$$\hat{P}_{rf\_rx} = P_{rf\_rx} - P_{rf\_sleep}, \tag{A.1}$$

$$\hat{P}_{rf\_tx} = P_{rf\_tx} - P_{rf\_sleep}, \tag{A.2}$$

$$\hat{P}_{fpga} = P_{fpga} - P_{fpga\_sleep}, \tag{A.3}$$

$$\hat{P}_{\mu c} = P_{\mu c} - P_{\mu c\_sleep}. \tag{A.4}$$

The power consumption of the Sindrion Transceiver with all components in sleep mode is

$$P_{sleep} = P_{rf\_sleep} + P_{fpga\_sleep} + P_{\mu c\_sleep} + P_{others}. \tag{A.5}$$

Therefore, if only the RF transceiver samples the channel periodically, the power consumption of a Sindrion Transceiver in the no traffic case is

$$P_{no\_traffic} = \hat{P}_{rf\_rx} \frac{T_{s\_rf} + T_{sample}}{T_{cycle}} + P_{sleep}, \tag{A.6}$$

which is the power consumption for all the preamble sampling based schemes in the no traffic case. Note that the power consumption of a component during setup time is assumed to be the normal power consumption of the component, e.g. $\hat{P}_{rf\_rx}$ is used in this case as the power consumption of RF setup.

**Note:** All the following computations of the power consumption exclude the power that has already been included in the no traffic case, i.e. the sleep power and the power consumed by $T_{s\_rf}$ and $T_{sample}$ in each cycle.

## A.1 The IEEE 802.15.4 MAC

To make the comparison to the preamble sampling based schemes using the hardware prototype given in Sec. 4.1 fair, it is assumed that the IEEE 802.15.4 MAC uses the same hardware structure, in which only the RF transceiver and the MAC module (FPGA) are woken up periodically to receive and parse the beacon, and the microcontroller can be switched on smartly on demand. The detailed protocol procedures are shown in Fig. A.1.

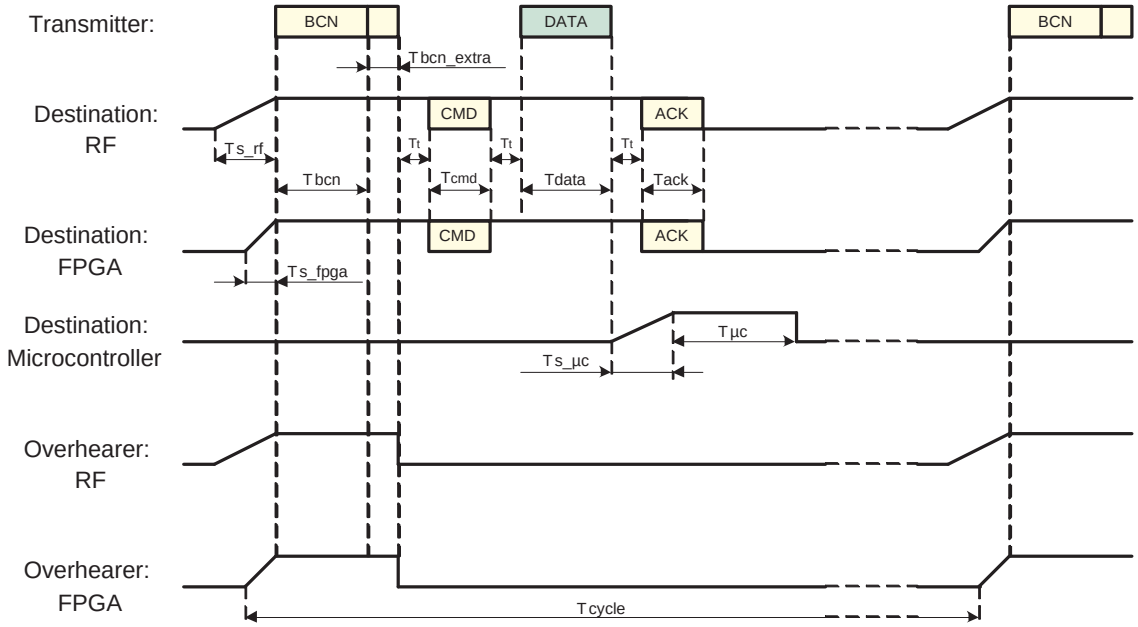


Figure A.1: The IEEE802.15.4 MAC Detail

The power consumption of the IEEE 802.15.4 MAC in the no traffic case is

$$P_{no\_traffic\_15.4} = \hat{P}_{rf\_rx}\frac{T_{s\_rf} + T_{bcn}}{T_{cycle}} + \hat{P}_{fpga}\frac{T_{s\_fpga} + T_{bcn}}{T_{cycle}} + P_{sleep}. \tag{A.7}$$

Note that the following computations do not include the situation that two or more frames are sent in one beacon interval, in which first collisions may happen, and second the slave that does not get the channel immediately should wait for the channel consuming more energy. Therefore, it is the optimistic computation, and the real case should be worse, especially when the traffic load is high.

### A.1.1 Unicast

The power consumption of receiving one unicast packet for the destination slave device is

$$\begin{aligned} P_{dst_{15.4\_u}} &= \hat{P}_{rf\_rx}(T_{bcn\_extra}) \\ &\quad +\hat{P}_{rf\_tx}(T_t + T_{cmd}) \\ &\quad +\hat{P}_{rf\_rx}(T_t + T_{data}) \\ &\quad +\hat{P}_{rf\_tx}(T_t + T_{ack}) \\ &\quad +\hat{P}_{fpga}(T_{bcn\_extra} + T_t + T_{cmd} + T_t + T_{data} + T_t + T_{ack}) \\ &\quad +\hat{P}_{\mu c}(T_{s\_\mu c} + T_{\mu c}), \end{aligned} \tag{A.8}$$

and the power consumption for an overhearer is

$$P_{ovh_{15.4\_u}} = \hat{P}_{rf\_rx} \cdot T_{bcn\_extra} + \hat{P}_{fpga} \cdot T_{bcn\_extra}, \tag{A.9}$$

where $T_{bcn\_extra}$ denotes the transmission time of one frame indication field in the beacon. The frame indication field is just the destination address of the slave, which is 64-bit long plus the modulation and ECC coding.

For each packet, a slave has the probability of $\frac{1}{N}$ to become the destination and the probability of $\frac{N-1}{N}$ to become an overhearer. Therefore, the average power consumption of receiving one unicast packet for each slave is

$$\overline{P_{pkt_{15.4\_u}}} = P_{dst_{15.4\_u}} \frac{1}{N} + P_{ovh_{15.4\_u}} \frac{N-1}{N}. \tag{A.10}$$

So the power consumption of the device for a Poisson arrival rate of $\lambda$ to each slave is

$$P_{15.4\_U} = \overline{P_{pkt_{15.4\_u}}} \cdot \lambda N + P_{no\_traffic\_15.4}. \tag{A.11}$$

The percentage of channel occupation is

$$C_{15.4\_U} = \left( \frac{T_{bcn}}{T_{cycle}} + (T_{bcn\_extra} + T_{cmd} + T_{data} + T_{ack}) \cdot \lambda N \right) \cdot 100\,[\%]. \tag{A.12}$$

The average response time of a unicast packet is

$$\overline{D_{15.4\_U}} = 0.5T_{cycle} + T_{bcn} + T_{bcn\_extra} + T_t + T_{cmd} + T_t + T_{data} + T_t + T_{ack}. \tag{A.13}$$

### A.1.2 Broadcast

The power consumption of receiving one broadcast packet for each slave is

$$\begin{aligned} P_{pkt_{15.4\_b}} &= \hat{P}_{rf\_rx} \cdot T_{data-mac} + \hat{P}_{fpga}(T_{data-mac} + T_{s\_\mu c}) \\ &\quad +\hat{P}_{\mu c}(T_{s\_\mu c} + T_{\mu c}), \end{aligned} \tag{A.14}$$

where $T_{data-mac}$ is the time needed to transmit the data frame excluding the MAC header. It is because that the broadcast data is transmitted as the payload of the beacon frame in the IEEE 802.15.4 MAC as mentioned Sec. 3.3.1.

So the power consumption of the device for a Poisson arrival rate of $\lambda$ is

$$P_{15.4\_B} = P_{pkt_{15.4\_b}} \cdot \lambda + P_{no\_traffic\_15.4}. \tag{A.15}$$

Note that not as in the unicast case, the $\lambda$ is the overall packet arrival rate from the master to all slaves. The percentage of channel occupation is

$$C_{15.4\_B} = \left( \frac{T_{bcn}}{T_{cycle}} + T_{data-mac} \cdot \lambda \right) \cdot 100\,[\%]. \tag{A.16}$$

The average response time of a broadcast packet is

$$\overline{D_{15.4\_B}} = 0.5 T_{cycle} + T_{bcn} + T_{data-mac}. \tag{A.17}$$

Note that the response time of broadcast is different from that of unicast because of the absence of the ACK.

## A.2 The WUP Scheme

The detailed procedures of the WUP scheme are shown in Fig. A.2.

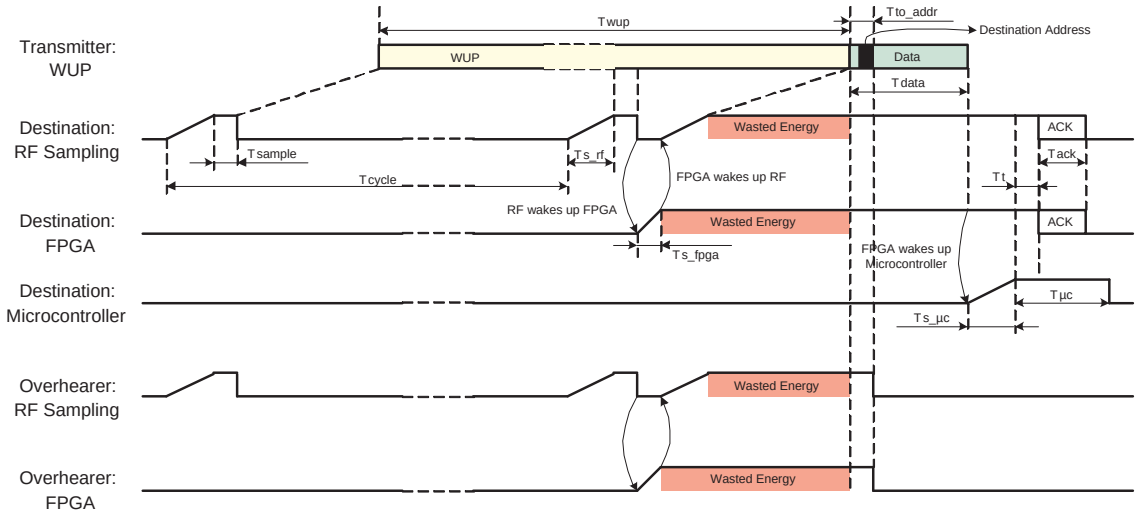


Figure A.2: The WUP Scheme Detail

The length of the WUP is

$$T_{pre_{wup}} = T_{cycle} + T_{sample} + T_{s\_fpga} + T_{s\_rf}. \tag{A.18}$$

### A.2.1 Unicast

The RF samples the channel and detects the WUP at the middle of $T_{cycle}$ on the average. Therefore, the average power consumption of receiving one unicast packet for the destination device is

$$\begin{aligned}\overline{P_{dst_{wup\_u}}} &= \hat{P}_{rf\_rx}(0.5T_{cycle} + T_{s\_fpga} + T_{s\_rf} - T_{s\_fpga} + T_{data} + T_{s\_\mu c}) \\ &\quad + \hat{P}_{rf\_tx}(T_t + T_{ack}) \\ &\quad + \hat{P}_{fpga}(0.5T_{cycle} + T_{s\_fpga} + T_{s\_rf} + T_{data} + T_{s\_\mu c} + T_t + T_{ack}) \\ &\quad + \hat{P}_{\mu c}(T_{s\_\mu c} + T_{\mu c}),\end{aligned} \tag{A.19}$$

and the average power consumption of receiving one unicast packet for an overhearer is

$$\begin{aligned}\overline{P_{ovh_{wup\_u}}} &= \hat{P}_{rf\_rx}(0.5T_{cycle} + T_{s\_fpga} + T_{s\_rf} - T_{s\_fpga} + T_{to\_addr}) \\ &\quad + \hat{P}_{fpga}(0.5T_{cycle} + T_{s\_fpga} + T_{s\_rf} + T_{to\_addr}).\end{aligned} \tag{A.20}$$

Note that an overhearer does not have to receive the whole data frame, because it can determine that this frame is not for itself as long as the destination address is received. For this reason, $T_{to\_addr}$ is used here instead of $T_{data}$. If the destination address is disturbed, the destination device works also as an overhearer, because a disturbed frame will be thrown anyway.

The average power consumption of receiving one unicast packet for each slave is

$$\overline{P_{pkt_{wup\_u}}} = \overline{P_{dst_{wup\_u}}}\frac{1}{N} + \overline{P_{ovh_{wup\_u}}}\frac{N-1}{N}. \tag{A.21}$$

So the power consumption of the device for a Poisson arrival rate of $\lambda$ to each slave is

$$P_{WUP\_U} = \overline{P_{pkt_{wup\_u}}} \cdot \lambda N + P_{no\_traffic}. \tag{A.22}$$

The percentage of channel occupation is

$$C_{WUP\_U} = (T_{pre_{wup}} + T_{data} + T_{ack}) \cdot \lambda N \cdot 100\,[\%]. \tag{A.23}$$

The response time of a unicast packet is

$$D_{WUP\_U} = T_{pre_{wup}} + T_{data} + T_{s\_\mu c} + T_t + T_{ack}. \tag{A.24}$$

### A.2.2 Broadcast

The average power consumption of receiving one broadcast packet for each slave is

$$\begin{aligned}\overline{P_{pkt_{wup\_b}}} &= \hat{P}_{rf\_rx}(0.5T_{cycle} + T_{s\_fpga} + T_{s\_rf} - T_{s\_fpga} + T_{data}) \\ &\quad + \hat{P}_{fpga}(0.5T_{cycle} + T_{s\_fpga} + T_{s\_rf} + T_{data} + T_{s\_\mu c}) \\ &\quad + \hat{P}_{\mu c}(T_{s\_\mu c} + T_{\mu c}).\end{aligned} \tag{A.25}$$

So the power consumption of the device for a Poisson arrival rate of $\lambda$ is

$$P_{WUP\_B} = \overline{P_{pkt_{wup\_b}}} \cdot \lambda + P_{no\_traffic}. \tag{A.26}$$

The percentage of channel occupation is

$$C_{WUP\_B} = (T_{pre_{wup}} + T_{data}) \cdot \lambda \cdot 100\,[\%]. \tag{A.27}$$

The response time of a broadcast packet is

$$D_{WUP\_B} = T_{pre_{wup}} + T_{data}. \tag{A.28}$$

## A.3 The Rep Scheme

The detailed procedures of the Rep scheme are shown in Fig. A.3.

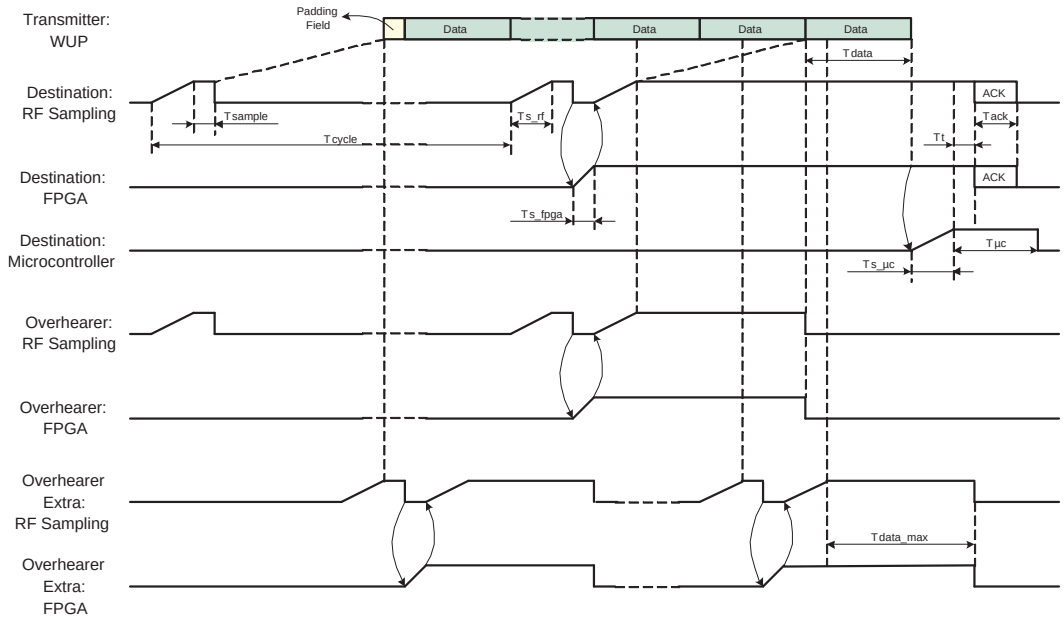


Figure A.3: The Rep Scheme Detail

In the Rep scheme, there is no difference between the preamble and data. However, to make the computation the same as the others, the total length of the repeated data excluding the last data frame is taken as the preamble. So the length of the preamble is

$$T_{pre_{rep}} = T_{cycle} + T_{sample} + T_{s\_fpga} + T_{s\_rf}. \tag{A.29}$$

Note that $(T_{cycle} + T_{sample} + T_{s\_fpga} + T_{s\_rf})$ normally cannot be exactly divided by $T_{data}$, so an extra padding field with sync bits is used at the beginning, as shown in Fig. A.3.

### A.3.1 Unicast

The average power consumption of receiving one unicast packet for the destination device is

$$\begin{aligned}\overline{P_{dst_{rep\_u}}} &= \hat{P}_{rf\_rx}(0.5T_{cycle} + T_{s\_fpga} + T_{s\_rf} - T_{s\_fpga} + T_{data} + T_{s\_\mu c}) \\ &\quad + \hat{P}_{rf\_tx}(T_t + T_{ack}) \\ &\quad + \hat{P}_{fpga}(0.5T_{cycle} + T_{s\_fpga} + T_{s\_rf} + T_{data} + T_{s\_\mu c} + T_t + T_{ack}) \\ &\quad + \hat{P}_{\mu c}(T_{s\_\mu c} + T_{\mu c}). \end{aligned} \tag{A.30}$$

The average power consumption for a normal overhearer is

$$\begin{aligned}\overline{P_{ovh\_normal}} &= \hat{P}_{rf\_rx}(T_{s\_rf} + 1.5T_{data}) \\ &\quad + \hat{P}_{fpga}(T_{s\_fpga} + T_{s\_rf} + 1.5T_{data}). \end{aligned} \tag{A.31}$$

Note that not as in the WUP scheme, a receiver in the Rep scheme should receive a complete data frame to decide whether it is an overhear. If the destination address is disturbed, the receiver can know it according to the CRC, then it should try to receive the next repetition of the data. As a result, an overhearer needs $1.5T_{data}$ on the average to do the decision instead of $(0.5T_{data} + T_{to\_addr})$ as in the WUP scheme.

Furthermore, one important point for the Rep scheme is that only one data rate can be used for all repetitions of the data frame, so not only the so-called preamble but also the last repetition of the data frame can wake up a receiver. So it is also possible that an overhearer that is woken up at the beginning will be woken up again at the end, as the overhearer extra shown in Fig. A.3. The probability of this situation is $\frac{T_{sample}+T_{s\_fpga}+T_{s\_rf}+T_{data}}{T_{cycle}}$. In this situation, the overhearer cannot receive the last repetition of the data, so it has to wait for at least one $T_{data\_max}$. The power consumption of this extra overhearing is

$$\begin{aligned}P_{ovh\_extra} &= \hat{P}_{rf\_rx}(T_{s\_rf} + T_{data\_max}) \\ &\quad + \hat{P}_{fpga}(T_{s\_fpga} + T_{s\_rf} + T_{data\_max}). \end{aligned} \tag{A.32}$$

So the average power consumption for an overhearer is

$$\overline{P_{ovh_{rep\_u}}} = \overline{P_{ovh\_normal}} + P_{ovh\_extra}\frac{T_{sample} + T_{s\_fpga} + T_{s\_rf} + T_{data}}{T_{cycle}}. \tag{A.33}$$

The average power consumption of receiving one packet for each slave is

$$\overline{P_{pkt_{rep\_u}}} = \overline{P_{dst_{rep\_u}}}\frac{1}{N} + \overline{P_{ovh_{rep\_u}}}\frac{N-1}{N}. \tag{A.34}$$

So the power consumption of the device for a Poisson arrival rate of $\lambda$ to each slave is

$$P_{REP\_U} = \overline{P_{pkt_{rep\_u}}} \cdot \lambda N + P_{no\_traffic}. \tag{A.35}$$

The percentage of channel occupation is

$$C_{REP\_U} = (T_{pre_{rep}} + T_{data} + T_{ack}) \cdot \lambda N \cdot 100\,[\%]. \tag{A.36}$$

The response time of a unicast packet is

$$D_{REP\_U} = T_{pre_{rep}} + T_{data} + T_{s\_\mu c} + T_t + T_{ack}. \tag{A.37}$$

### A.3.2 Broadcast

The average power consumption of receiving one broadcast packet for each slave is

$$\begin{aligned} \overline{P_{pkt_{rep\_b}}} &= \hat{P}_{rf\_rx}(T_{s\_rf} + 1.5T_{data}) \\ &\quad + \hat{P}_{fpga}(T_{s\_fpga} + T_{s\_rf} + 1.5T_{data} + T_{s\_\mu c}) \\ &\quad + \hat{P}_{\mu c}(T_{s\_\mu c} + T_{\mu c}). \end{aligned} \tag{A.38}$$

So the power consumption of the device for a Poisson arrival rate of $\lambda$ is

$$P_{REP\_B} = \overline{P_{pkt_{rep\_b}}} \cdot \lambda + P_{no\_traffic}. \tag{A.39}$$

The percentage of channel occupation is

$$C_{REP\_B} = (T_{pre_{rep}} + T_{data}) \cdot \lambda \cdot 100\,[\%]. \tag{A.40}$$

The average response time of a broadcast packet is

$$\overline{D_{REP\_B}} = 0.5T_{cycle} + T_{sample} + T_{s\_fpga} + T_{s\_rf} + 1.5T_{data}. \tag{A.41}$$

## A.4 The WUF Scheme

The detailed procedures of the WUF scheme are shown in Fig. A.4. The length of the WUF is

$$T_{pre_{wuf}} = T_{cycle} + T_{sample} + T_{s\_fpga} + T_{s\_rf} + T_{swuf}. \tag{A.42}$$

### A.4.1 Unicast

The average power consumption of receiving one packet for the destination device is

$$\begin{aligned} \overline{P_{dst_{wuf\_u}}} &= \hat{P}_{rf\_rx}(T_{s\_rf} + 1.5T_{swuf} + T_{s\_rf} + T_{data} + T_{s\_\mu c}) \\ &\quad + \hat{P}_{rf\_tx}(T_t + T_{ack}) \\ &\quad + \hat{P}_{fpga}(T_{s\_fpga} + T_{s\_rf} + 1.5T_{swuf} + T_{s\_fpga} + T_{s\_rf} + T_{data} + T_{s\_\mu c} + T_t + T_{ack}) \\ &\quad + \hat{P}_{\mu c}(T_{s\_\mu c} + T_{\mu c}). \end{aligned} \tag{A.43}$$

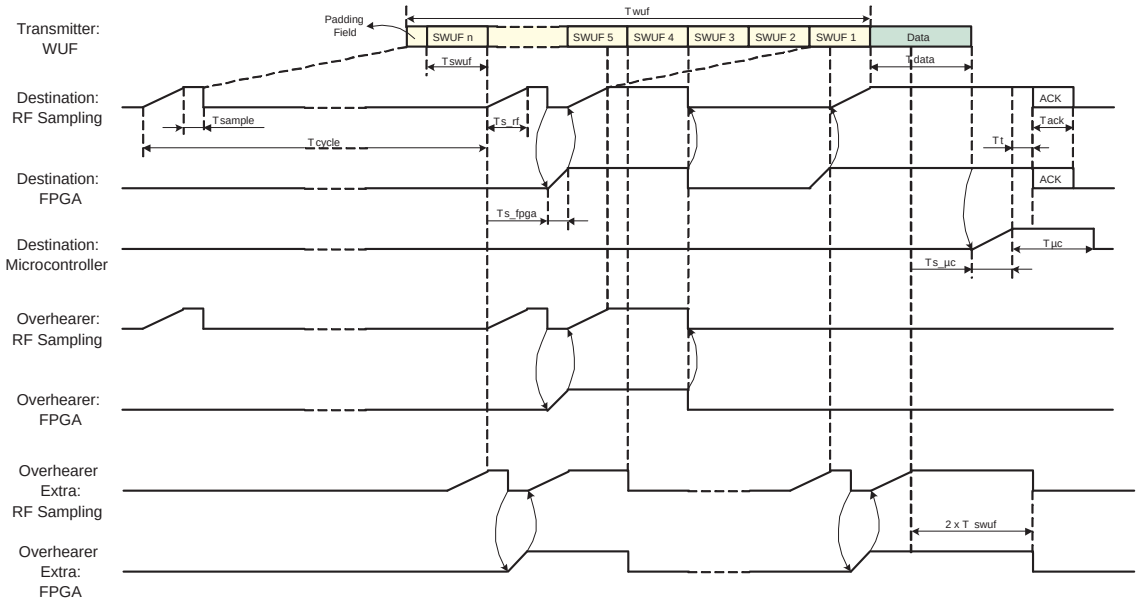


Figure A.4: The WUF Scheme Detail

The average power consumption for an normal overhearer is

$$\begin{aligned}\overline{P_{ovh\_normal}} &= \hat{P}_{rf\_rx}(T_{s\_rf} + 1.5T_{swuf}) \\ &\quad + \hat{P}_{fpga}(T_{s\_fpga} + T_{s\_rf} + 1.5T_{swuf}).\end{aligned} \tag{A.44}$$

However, since the length of the WUF is longer than one cycle period, it is possible that an overhearer that is woken up at the beginning of the WUF is woken up again at the end of the WUF, as the overhearer extra shown in Fig. A.4. The probability of this situation is $\frac{T_{sample}+T_{s\_fpga}+T_{s\_rf}+T_{swuf}}{T_{cycle}}$. If the overhear is woken up at the end of the WUF, it cannot hear any SWUF, so it should wait for at least $2T_{swuf}$ before returns to sleep, so extra power consumption is induced

$$\begin{aligned}P_{ovh\_extra} &= \hat{P}_{rf\_rx}(T_{s\_rf} + 2T_{swuf}) \\ &\quad + \hat{P}_{fpga}(T_{s\_fpga} + T_{s\_rf} + 2T_{swuf}).\end{aligned} \tag{A.45}$$

Note that the WUP scheme has no such situation because an overhearer that is woken up at the beginning of the preamble should keep awake until it receives the data frame.

Therefore, the average of the power consumption of an overhearer is

$$\overline{P_{ovh_{wuf\_u}}} = \overline{P_{ovh\_normal}} + P_{ovh\_extra}\frac{T_{sample} + T_{s\_fpga} + T_{s\_rf} + T_{swuf}}{T_{cycle}}. \tag{A.46}$$

The average power consumption of receiving one packet for each slave is

$$\overline{P_{pkt_{wuf\_u}}} = \overline{P_{dst_{wuf\_u}}}\frac{1}{N} + \overline{P_{ovh_{wuf\_u}}}\frac{N-1}{N}. \tag{A.47}$$

So the power consumption of the device for a Poisson arrival rate of $\lambda$ for each slave is

$$P_{WUF\_U} = \overline{P_{pkt_{wuf\_u}}} \cdot \lambda N + P_{no\_traffic}. \tag{A.48}$$

The percentage of channel occupation is

$$C_{WUF\_U} = (T_{pre_{wuf}} + T_{data} + T_{ack}) \cdot \lambda N \cdot 100\,[\%]. \tag{A.49}$$

The response time of a unicast packet is

$$D_{WUF\_U} = T_{pre_{wuf}} + T_{data} + T_{s\_\mu c} + T_t + T_{ack}. \tag{A.50}$$

### A.4.2 Broadcast

The average power consumption of receiving one broadcast packet for each slave is

$$\begin{aligned} \overline{P_{pkt_{wuf\_b}}} \quad = \quad & \hat{P}_{rf\_rx}(T_{s\_rf} + 1.5T_{swuf} + T_{s\_rf} + T_{data}) \\ & + \hat{P}_{fpga}(T_{s\_fpga} + T_{s\_rf} + 1.5T_{swuf} + T_{s\_fpga} + T_{s\_rf} + T_{data} + T_{s\_\mu c}) \\ & + \hat{P}_{\mu c}(T_{s\_\mu c} + T_{\mu c}), \end{aligned} \tag{A.51}$$

So the power consumption of the device for a Poisson arrival rate of $\lambda$ is

$$P_{WUF\_B} = \overline{P_{pkt_{wuf\_b}}} \cdot \lambda + P_{no\_traffic}. \tag{A.52}$$

The percentage of channel occupation is

$$C_{WUF\_B} = (T_{pre_{wuf}} + T_{data}) \cdot \lambda \cdot 100\,[\%]. \tag{A.53}$$

The response time of a broadcast packet is

$$D_{WUF\_B} = T_{pre_{wuf}} + T_{data}. \tag{A.54}$$

## A.5 The WiseMAC+WUP Scheme

### A.5.1 Unicast

The interarrival interval of a Poisson process traffic model is exponentially distributed with a probability density function

$$p(t) = \lambda e^{-\lambda t}, \tag{A.55}$$

and a cumulative distribution function

$$f(t) = 1 - e^{-\lambda t}. \tag{A.56}$$

As mentioned in Sec. 3.3.3, under a certain threshold of the interval, the WiseMAC enables a shortened WUS, which is denoted by

$$T_{th} = \frac{T_{cycle}}{4\theta}, \tag{A.57}$$

where $\theta$ is the clock drift of the device.

The probability to generate an interval that is less than $T_{th}$ (case 1) is

$$p_1 = p(t < T_{th}) = f(T_{th}) = 1 - e^{-\lambda T_{th}}, \tag{A.58}$$

and that is larger than $T_{th}$ (case 2) is

$$p_2 = p(t \geq T_{th}) = 1 - f(T_{th}) = e^{-\lambda T_{th}}. \tag{A.59}$$

**Case 1:** $interval < T_{th}$

In this case, the detailed procedures of the WiseMAC+WUP scheme are shown in Fig. A.5.

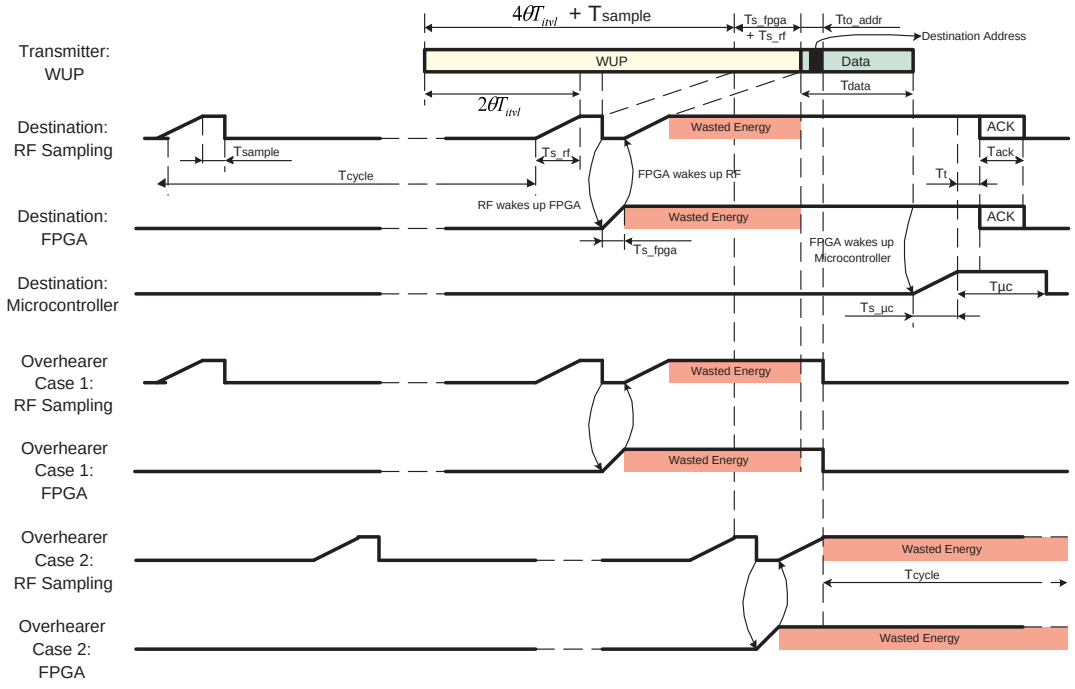


Figure A.5: The WiseMAC+WUP Detail

The length of the shortened WUP is

$$T_{pre1} = 4\theta T_{itvl} + T_{sample} + T_{s\_fpga} + T_{s\_rf}, \tag{A.60}$$

where $T_{itvl}$ is the Poisson interarrival interval and its expectation is

$$\begin{aligned}
\overline{T_{itvl}} &= E\left(T_{itvl}\right) \\
&= \int_0^{T_{th}} t \cdot \frac{\lambda}{p_1} e^{-\lambda t} dt \\
&= \frac{\lambda}{p_1}\left(\frac{1}{-\lambda} t e^{-\lambda t}\Big|_0^{T_{th}} - \frac{1}{-\lambda}\int_0^{T_{th}} e^{-\lambda t} dt\right) \\
&= \frac{\lambda}{p_1}\left(\frac{1}{-\lambda} T_{th} e^{-\lambda T_{th}} - \frac{1}{-\lambda}\frac{1}{-\lambda}\, e^{-\lambda t}\Big|_0^{T_{th}}\right) \\
&= \frac{\lambda}{p_1}\left(\frac{1}{-\lambda} T_{th} e^{-\lambda T_{th}} - \frac{1}{\lambda^2}\left(e^{-\lambda T_{th}} - 1\right)\right) \\
&= \frac{1}{p_1}\left(-T_{th} p_2 + \frac{1}{\lambda} p_1\right) \\
&= \frac{1}{\lambda} - \frac{1-p_1}{p_1} T_{th}
\end{aligned} \tag{A.61}$$

So the mean of $T_{pre1}$ is

$$\overline{T_{pre1}} = 4\theta\left(\frac{1}{\lambda} - \frac{1-p_1}{p_1} T_{th}\right) + T_{sample} + T_{s\_fpga} + T_{s\_rf}. \tag{A.62}$$

The average power consumption of receiving one packet for the destination device is

$$\begin{aligned}
\overline{P_{dst1}} &= \hat{P}_{rf_rx}(2\theta T_{itvl} + T_{s_fpga} + T_{s_rf} - T_{s_fpga} + T_{data} + T_{s_\mu c}) \\
&\quad + \hat{P}_{rf_tx}(T_t + T_{ack}) \\
&\quad + \hat{P}_{fpga}(2\theta T_{itvl} + T_{s_fpga} + T_{s_rf} + T_{data} + T_{s_\mu c} + T_t + T_{ack}) \\
&\quad + \hat{P}_{\mu c}(T_{s_\mu c} + T_{\mu c}) \\
&= \underbrace{2\theta(\hat{P}_{rf_rx} + \hat{P}_{fpga})}_{A} T_{itvl} \\
&\quad + \underbrace{\hat{P}_{rf_rx}(T_{s_rf} + T_{data} + T_{s_\mu c}) + \hat{P}_{rf_tx}(T_t + T_{ack})}_{P_a} \\
&\quad + \underbrace{\hat{P}_{fpga}(T_{s_fpga} + T_{s_rf} + T_{data} + T_{s_\mu c} + T_t + T_{ack})}_{P_b} \\
&\quad + \underbrace{\hat{P}_{\mu c}(T_{s_\mu c} + T_{\mu c})}_{P_c} \\
&= AT_{itvl} + \underbrace{P_a + P_b + P_c}_{B} \\
&= AT_{itvl} + B.
\end{aligned} \tag{A.63}$$

For an overhearer, there are two cases. In the first case, the overhearer is woken up by the first $(4\theta T_{itvl} + T_{sample})$ part of $T_{pre1}$, so the overhearer can receive the following data

frame and make a decision according to the destination address, as the overhearer case 1 shown in Fig. A.5. The probability of this case is

$$p_{ovh1\_1} = \frac{4\theta T_{itvl} + T_{sample}}{T_{pre1}}, \tag{A.64}$$

and the average power consumption in this case is

$$\begin{aligned}
\overline{P_{ovh1_1}} &= \hat{P}_{rf_rx}(2\theta T_{itvl} + T_{s_fpga} + T_{s_rf} - T_{s_fpga} + T_{to_addr}) \\
&\quad + \hat{P}_{fpga}(2\theta T_{itvl} + T_{s_fpga} + T_{s_rf} + T_{to_addr}) \\
&= \underbrace{2\theta(\hat{P}_{rf_rx} + \hat{P}_{fpga})}_{A} T_{itvl} \\
&\quad + \underbrace{\hat{P}_{rf_rx}(T_{s_rf} + T_{to_addr}) + \hat{P}_{fpga}(T_{s_fpga} + T_{s_rf} + T_{to_addr})}_{C} \\
&= AT_{itvl} + C.
\end{aligned} \tag{A.65}$$

In the second case, the overhearer is woken up by the last $(T_{s\_fpga} + T_{s\_rf})$ part of $T_{pre1}$, so the overhearer has no time to setup to receive the following data frame. As a result, it has to wait one $T_{cycle}$ if nothing is received after being woken up, because overhearers do not know the length of the WUP, as the overhearer case 2 shown in Fig. A.5. The probability of this case is

$$p_{ovh1\_2} = \frac{T_{s\_fpga} + T_{s\_rf}}{T_{pre1}}, \tag{A.66}$$

and the power consumption in this case is

$$P_{ovh1\_2} = \hat{P}_{rf\_rx}(T_{s\_rf} + T_{cycle}) + \hat{P}_{fpga}(T_{s\_fpga} + T_{s\_rf} + T_{cycle}), \tag{A.67}$$

which is not a function of $T_{itvl}$. So the average power consumption for an overhearer is

$$\begin{aligned}
\overline{P_{ovh1}} &= \overline{P_{ovh1_1}} \cdot p_{ovh1_1} + P_{ovh1_2} \cdot p_{ovh1_2} \\
&= (AT_{itvl} + C)\frac{4\theta T_{itvl} + T_{sample}}{T_{pre1}} + P_{ovh1_2}\frac{T_{s_fpga} + T_{s_rf}}{T_{pre1}}.
\end{aligned} \tag{A.68}$$

Since the preamble is shortened, only a part of the slaves is woken up to become overhearers. The probability that a slave is woken up is $\frac{T_{pre1}}{T_{cycle}}$. So the average power consumption of receiving one packet for each slave in case 1 ($T_{itvl} < T_{th}$) is

$$
\begin{aligned}
\overline{P'_{pkt1}} &= \overline{P_{dst1}}\frac{1}{N} + \overline{P_{ovh1}}\frac{T_{pre1}}{T_{cycle}}\frac{N-1}{N} \\
&= (AT_{itvl}+B)\frac{1}{N} + \left((AT_{itvl}+C)\frac{4\theta T_{itvl}+T_{sample}}{T_{pre1}} + \overline{P_{ovh1\_2}}\frac{T_{s\_fpga}+T_{s\_rf}}{T_{pre1}}\right)\frac{T_{pre1}}{T_{cycle}}\frac{N-1}{N} \\
&= (AT_{itvl}+B)\frac{1}{N} + \left((AT_{itvl}+C)\frac{4\theta T_{itvl}+T_{sample}}{T_{cycle}} + \overline{P_{ovh1\_2}}\frac{T_{s\_fpga}+T_{s\_rf}}{T_{cycle}}\right)\frac{N-1}{N} \\
&= \frac{A}{N}T_{itvl} + \frac{B}{N} + \left(\frac{A}{N}T_{itvl}+\frac{C}{N}\right)\underbrace{4\theta\frac{N-1}{T_{cycle}}}_{D}T_{itvl} + A\frac{T_{sample}}{T_{cycle}}\frac{N-1}{N}T_{itvl} \\
&\quad + \underbrace{\left(C\frac{T_{sample}}{T_{cycle}} + \overline{P_{ovh1\_2}}\frac{T_{s\_fpga}+T_{s\_rf}}{T_{cycle}}\right)\frac{N-1}{N}}_{E} \\
&= \frac{A}{N}T_{itvl} + \frac{B}{N} + \frac{AD}{N}T_{itvl}^2 + \frac{CD}{N}T_{itvl} + A\frac{T_{sample}}{T_{cycle}}\frac{N-1}{N}T_{itvl} + E \\
&= \underbrace{\frac{AD}{N}}_{F}T_{itvl}^2 + \underbrace{\left(\frac{A+CD}{N} + A\frac{T_{sample}}{T_{cycle}}\frac{N-1}{N}\right)}_{G}T_{itvl} + \underbrace{\frac{B}{N}+E}_{H} \\
&= FT_{itvl}^2 + GT_{itvl} + H. \qquad \text{(A.69)}
\end{aligned}
$$

The expectation of $\overline{P'_{pkt1}}$ is the average power consumption of receiving one packet for each slave

$$
\begin{aligned}
\overline{P_{pkt1}} &= E\left(\overline{P'_{pkt1}}\right) \\
&= \int_0^{T_{th}} (Ft^2+Gt+H)\frac{p(t)}{p_1}dt \\
&= \int_0^{T_{th}} (Ft^2+Gt+H)\frac{\lambda}{p_1}e^{-\lambda t}dt \\
&= \frac{\lambda F}{p_1}\int_0^{T_{th}} t^2e^{-\lambda t}dt + \frac{\lambda G}{p_1}\int_0^{T_{th}} te^{-\lambda t}dt + \frac{\lambda H}{p_1}\int_0^{T_{th}} e^{-\lambda t}dt \\
&= \frac{\lambda F}{p_1}\left(\frac{1}{-\lambda}t^2e^{-\lambda t}\Big|_0^{T_{th}} - \frac{2}{-\lambda}\int_0^{T_{th}} te^{-\lambda t}dt\right) + \frac{\lambda G}{p_1}\int_0^{T_{th}} te^{-\lambda t}dt + \frac{\lambda H}{p_1}\int_0^{T_{th}} e^{-\lambda t}dt \\
&= \frac{-F}{p_1}\,t^2e^{-\lambda t}\Big|_0^{T_{th}} + \frac{2F+\lambda G}{p_1}\int_0^{T_{th}} te^{-\lambda t}dt + \frac{\lambda H}{p_1}\int_0^{T_{th}} e^{-\lambda t}dt \\
&= \frac{-F}{p_1}\,t^2e^{-\lambda t}\Big|_0^{T_{th}} + \frac{2F+\lambda G}{p_1}\left(\frac{1}{-\lambda}te^{-\lambda t}\Big|_0^{T_{th}} - \frac{1}{-\lambda}\int_0^{T_{th}} e^{-\lambda t}dt\right) + \frac{\lambda H}{p_1}\int_0^{T_{th}} e^{-\lambda t}dt \\
&= \frac{-F}{p_1}\,t^2e^{-\lambda t}\Big|_0^{T_{th}} + \frac{2F+\lambda G}{-\lambda p_1}\,te^{-\lambda t}\Big|_0^{T_{th}} + \frac{2F+\lambda G+\lambda^2 H}{\lambda p_1}\int_0^{T_{th}} e^{-\lambda t}dt \\
&= \frac{-F}{p_1}\,t^2e^{-\lambda t}\Big|_0^{T_{th}} + \frac{2F+\lambda G}{-\lambda p_1}\,te^{-\lambda t}\Big|_0^{T_{th}} + \frac{2F+\lambda G+\lambda^2 H}{-\lambda^2 p_1}\,e^{-\lambda t}\Big|_0^{T_{th}} \\
&= \frac{-F}{p_1}T_{th}^2e^{-\lambda T_{th}} + \frac{2F+\lambda G}{-\lambda p_1}T_{th}e^{-\lambda T_{th}} + \frac{2F+\lambda G+\lambda^2 H}{-\lambda^2 p_1}(e^{-\lambda T_{th}}-1) \\
&= \frac{-F}{p_1}T_{th}^2p_2 + \frac{2F+\lambda G}{-\lambda p_1}T_{th}p_2 + \frac{2F+\lambda G+\lambda^2 H}{-\lambda^2 p_1}(-p_1) \\
&= 2F\frac{1}{\lambda^2} + \left(G - 2FT_{th}\frac{1-p_1}{p_1}\right)\frac{1}{\lambda} + \left(H - GT_{th}\frac{1-p_1}{p_1} - FT_{th}^2\frac{1-p_1}{p_1}\right). \qquad \text{(A.70)}
\end{aligned}
$$

The average time of channel occupation is

$$\overline{C_1} = \overline{T_{pre1}} + T_{data} + T_{ack} \tag{A.71}$$

The average response time of a unicast packet is

$$\overline{D_1} = 0.5T_{cycle} + 2\theta\left(\frac{1}{\lambda} - \frac{1-p_1}{p_1}T_{th_1}\right) + T_{sample} + T_{s\_fpga} + T_{s\_rf} + T_{data} + T_{s\_\mu c} + T_t + T_{ack}. \tag{A.72}$$

**Case 2:** $interval \geq T_{th}$

When the interval is greater than $T_{th}$, the length of the WUP is the same as in the normal WUP scheme. So the corresponding equations are the same as Equ. A.18, A.19, A.20, i.e.

$$T_{pre2} = T_{pre_{wup}} = T_{cycle} + T_{sample} + T_{s\_fpga} + T_{s\_rf}, \tag{A.73}$$

$$\begin{aligned}\overline{P_{dst2}} &= \overline{P_{dst_{wup\_u}}} \\ &= \hat{P}_{rf\_rx}(0.5T_{cycle} + T_{s\_fpga} + T_{s\_rf} - T_{s\_fpga} + T_{data} + T_{s\_\mu c}) \\ &\quad + \hat{P}_{rf\_tx}(T_t + T_{ack}) \\ &\quad + \hat{P}_{fpga}(0.5T_{cycle} + T_{s\_fpga} + T_{s\_rf} + T_{data} + T_{s\_\mu c} + T_t + T_{ack}) \\ &\quad + \hat{P}_{\mu c}(T_{s\_\mu c} + T_{\mu c}),\end{aligned} \tag{A.74}$$

$$\begin{aligned}\overline{P_{ovh2}} &= \overline{P_{ovh_{wup\_u}}} \\ &= \hat{P}_{rf\_rx}(0.5T_{cycle} + T_{s\_fpga} + T_{s\_rf} - T_{s\_fpga} + T_{to\_addr}) \\ &\quad + \hat{P}_{fpga}(0.5T_{cycle} + T_{s\_fpga} + T_{s\_rf} + T_{to\_addr}).\end{aligned} \tag{A.75}$$

The average power consumption of receiving one packet for each slave in case 2 ($T_{itvl} \geq T_{th}$) is

$$\overline{P_{pkt2}} = \overline{P_{dst2}}\frac{1}{N} + \overline{P_{ovh2}}\frac{N-1}{N}. \tag{A.76}$$

The time of channel occupation is

$$C_2 = T_{pre2} + T_{data} + T_{ack}. \tag{A.77}$$

The response time of a unicast packet is

$$D_2 = T_{pre2} + T_{data} + T_{s\_\mu c} + T_t + T_{ack}. \tag{A.78}$$

**Overall**

All in all, the average power consumption of receiving one packet for each slave in the WiseMAC+WUP is

$$\overline{P_{pkt_{WiseMAC+WUP\_U}}} = \overline{P_{pkt1}} \cdot p_1 + \overline{P_{pkt2}} \cdot p_2. \tag{A.79}$$

So the power consumption of the device for a Poisson arrival rate of $\lambda$ to each slave is

$$P_{WiseMAC+WUP\_U} = \overline{P_{pkt_{WiseMAC+WUP\_U}}} \cdot \lambda N + P_{no\_traffic}. \tag{A.80}$$

The overall percentage of channel occupation is

$$C_{WiseMAC+WUP\_U} = (\overline{C_1} \cdot p_1 + C_2 \cdot p_2) \cdot \lambda N \cdot 100\,[\%]. \tag{A.81}$$

The average response time of a unicast packet is

$$\overline{D_{WiseMAC+WUP\_U}} = \overline{D_1} \cdot p_1 + D_2 \cdot p_2. \tag{A.82}$$

### A.5.2 Broadcast

The WiseMAC+WUP cannot provide improvement for broadcast packet, so it is the same as the broadcast case in the WUP scheme, see Sec. A.2.2.

## A.6 The WiseMAC+Rep Scheme

This WiseMAC+Rep scheme repeats the data frame when the WUS is longer than one data frame.

### A.6.1 Unicast

When the length of the WUS is less than the length of one data frame, the Rep scheme makes no difference from the WUP, so a threshold of the Poisson interarrival interval is defined by

$$T_{th_1} = \frac{T_{data}}{4\theta}. \tag{A.83}$$

Under a certain threshold of the interval, the WiseMAC enables a shortened WUS, which is denoted by

$$T_{th_2} = \frac{T_{cycle}}{4\theta}. \tag{A.84}$$

The probability to generate an interval that is less than $T_{th_1}$ (case 1) is

$$p_1 = p(t < T_{th_1}) = f(T_{th_1}) = 1 - e^{-\lambda T_{th_1}}. \tag{A.85}$$

The probability that it is larger than $T_{th_1}$ but less than $T_{th_2}$ (case 2) is

$$p_2 = p(T_{th_1} \leq t < T_{th_2}) = f(T_{th_2}) - f(T_{th_1}) = e^{-\lambda T_{th_1}} - e^{-\lambda T_{th_2}}. \tag{A.86}$$

Finally, the probability that the interval is larger than $T_{th_2}$ (case 3) is

$$p_3 = p(t \geq T_{th_2}) = 1 - f(T_{th_2}) = e^{-\lambda T_{th_2}}. \tag{A.87}$$

**Case 1:** $interval < T_{th_1}$

In this case, the detailed procedures of the WiseMAC+Rep scheme are shown in Fig. A.6.

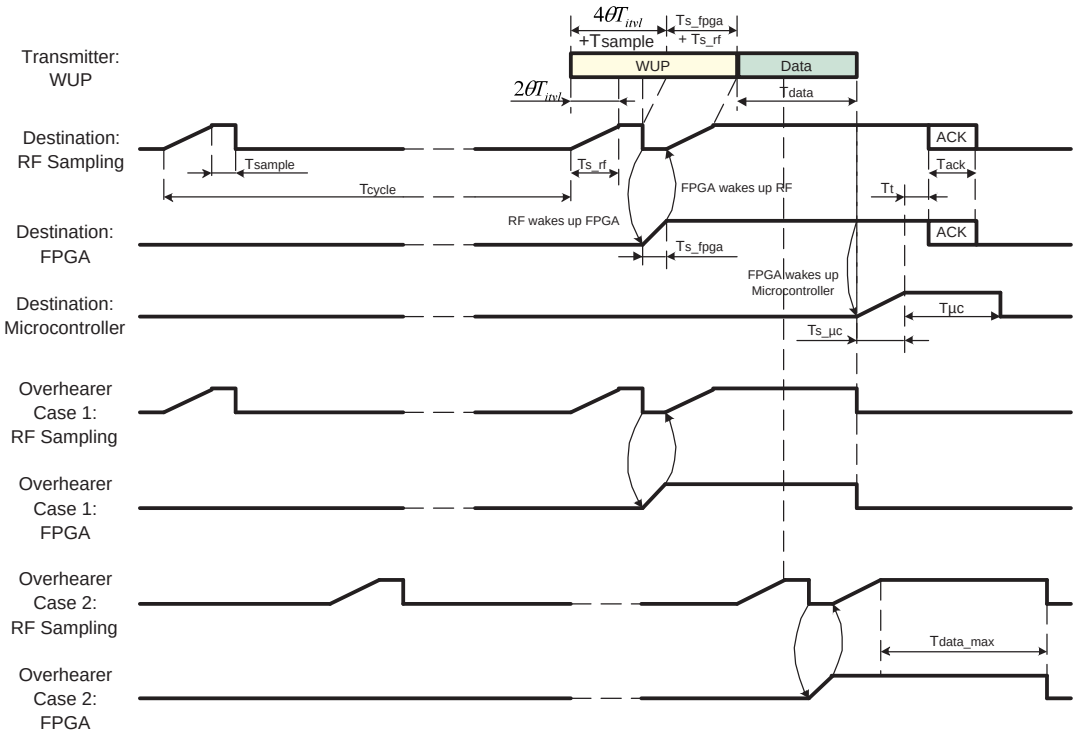


Figure A.6: The WiseMAC+Rep Detail (Case 1)

The length of the shortened preamble is

$$T_{pre1} = 4\theta T_{itvl} + T_{sample} + T_{s\_fpga} + T_{s\_rf}, \tag{A.88}$$

where $T_{itvl}$ is the Poisson interarrival interval and its expectation is given by Equ. A.61

$$\overline{T_{itvl}} = \frac{1}{\lambda} - \frac{1-p_1}{p_1} T_{th_1}. \tag{A.89}$$

So the mean of $T_{pre1}$ is

$$\overline{T_{pre1}} = 4\theta \left( \frac{1}{\lambda} - \frac{1-p_1}{p_1} T_{th_1} \right) + T_{sample} + T_{s\_fpga} + T_{s\_rf}. \tag{A.90}$$

The average power consumption of receiving one unicast packet for the destination device

is given by Equ. A.63

$$\begin{aligned}\overline{P_{dst1}} &= \hat{P}_{rf\_rx}(2\theta T_{itvl} + T_{s\_fpga} + T_{s\_rf} - T_{s\_fpga} + T_{data} + T_{s\_\mu c}) \\ &\quad + \hat{P}_{rf\_tx}(T_t + T_{ack}) \\ &\quad + \hat{P}_{fpga}(2\theta T_{itvl} + T_{s\_fpga} + T_{s\_rf} + T_{data} + T_{s\_\mu c} + T_t + T_{ack}) \\ &\quad + \hat{P}_{\mu c}(T_{s\_\mu c} + T_{\mu c}) \\ &= AT_{itvl} + B. \end{aligned} \tag{A.91}$$

Note that, in the Rep scheme, the so-called preamble is also the repeated data, so both the data and the preamble have the same data rate. Therefore, they both can wake up a slave that hears them.

For an overhearer, there are two cases. In the first case, the overhearer is woken up by the first $(4\theta T_{itvl} + T_{sample})$ part of $T_{pre1}$, so the overhearer can receive the data frame and make decision according to the destination address, as the overhearer case 1 shown in Fig. A.6. The probability of this case is

$$p_{ovh1\_1} = \frac{4\theta T_{itvl} + T_{sample}}{T_{pre1} + T_{data}}, \tag{A.92}$$

and the average power consumption in this case is

$$\begin{aligned}\overline{P_{ovh1\_1}} &= \hat{P}_{rf\_rx}(2\theta T_{itvl} + T_{s\_fpga} + T_{s\_rf} - T_{s\_fpga} + T_{data}) \\ &\quad + \hat{P}_{fpga}(2\theta T_{itvl} + T_{s\_fpga} + T_{s\_rf} + T_{data}) \\ &= \underbrace{2\theta(\hat{P}_{rf\_rx} + \hat{P}_{fpga})}_{A} T_{itvl} \\ &\quad + \underbrace{\hat{P}_{rf\_rx}(T_{s\_rf} + T_{data}) + \hat{P}_{fpga}(T_{s\_fpga} + T_{s\_rf} + T_{data})}_{I} \\ &= AT_{itvl} + I. \end{aligned} \tag{A.93}$$

Note that, unlike Equ. A.65, an overhear should receive the complete data frame in the Rep scheme instead of just until the destination address in the WUP scheme.

In the second case, the overhearer is woken up by the last $(T_{s\_fpga} + T_{s\_rf})$ part of $T_{pre1}$ or by $T_{data}$, so the overhearer cannot receive the data frame. As a result, it has to wait for at least one $T_{data\_max}$ if nothing is received after being woken up, as the overhearer case 2 shown in Fig. A.6. The probability of this case is

$$p_{ovh1\_2} = \frac{T_{s\_fpga} + T_{s\_rf} + T_{data}}{T_{pre1} + T_{data}}, \tag{A.94}$$

and the power consumption in this case is

$$P_{ovh1\_2} = \hat{P}_{rf\_rx}(T_{s\_rf} + T_{data\_max}) + \hat{P}_{fpga}(T_{s\_fpga} + T_{s\_rf} + T_{data\_max}), \tag{A.95}$$

which is not a function of $T_{itvl}$.

So the average power consumption for an overhearer is

$$\begin{aligned}\overline{P_{ovh1}} &= \overline{P_{ovh1\_1}} \cdot p_{ovh1\_1} + P_{ovh1\_2} \cdot p_{ovh1\_2} \\ &= \overline{P_{ovh1\_1}} \frac{4\theta T_{itvl} + T_{sample}}{T_{pre1} + T_{data}} + P_{ovh1\_2} \frac{T_{s\_fpga} + T_{s\_rf} + T_{data}}{T_{pre1} + T_{data}}. \end{aligned} \tag{A.96}$$

Since the preamble is shortened, only a part of the slaves is woken up to become overhearers. The probability that a slave is woken up is $\frac{T_{pre1}+T_{data}}{T_{cycle}}$. So the power consumption of receiving one packet for each slave in case 1 ($T_{itvl} < T_{th_1}$) is

$$\begin{aligned}\overline{P'_{pkt1}} &= \overline{P_{dst1}} \frac{1}{N} + \overline{P_{ovh1}} \frac{T_{pre1} + T_{data}}{T_{cycle}} \frac{N-1}{N} \\ &= (AT_{itvl} + B)\frac{1}{N} + \left((AT_{itvl} + I)\frac{4\theta T_{itvl} + T_{sample}}{T_{cycle}} + \overline{P_{ovh1\_2}} \frac{T_{s\_fpga} + T_{s\_rf} + T_{data}}{T_{cycle}}\right)\frac{N-1}{N} \\ &= \frac{A}{N}T_{itvl} + \frac{B}{N} + \left(\frac{A}{N}T_{itvl} + \frac{I}{N}\right)\underbrace{4\theta\frac{N-1}{T_{cycle}}}_{D} T_{itvl} + A\frac{T_{sample}}{T_{cycle}}\frac{N-1}{N}T_{itvl} \\ &\quad + \underbrace{\left(I\frac{T_{sample}}{T_{cycle}} + \overline{P_{ovh1\_2}}\frac{T_{s\_fpga} + T_{s\_rf} + T_{data}}{T_{cycle}}\right)\frac{N-1}{N}}_{J} \\ &= \frac{A}{N}T_{itvl} + \frac{B}{N} + \frac{AD}{N}T_{itvl}^2 + \frac{ID}{N}T_{itvl} + A\frac{T_{sample}}{T_{cycle}}\frac{N-1}{N}T_{itvl} + J \\ &= \underbrace{\frac{AD}{N}}_{F}T_{itvl}^2 + \underbrace{\left(\frac{A+ID}{N} + A\frac{T_{sample}}{T_{cycle}}\frac{N-1}{N}\right)}_{K}T_{itvl} + \underbrace{\frac{B}{N} + J}_{L} \\ &= FT_{itvl}^2 + KT_{itvl} + L. \end{aligned} \tag{A.97}$$

The average power consumption of receiving one packet for each slave is the expectation of $\overline{P'_{pkt1}}$. According to Equ. A.70,

$$\begin{aligned}\overline{P_{pkt1}} &= E\left(\overline{P'_{pkt1}}\right) \\ &= \int_0^{T_{th_1}} (Ft^2 + Kt + L)\frac{p(t)}{p_1}dt \\ &= \int_0^{T_{th_1}} (Ft^2 + Kt + L)\frac{\lambda}{p_1}e^{-\lambda t}dt \\ &= 2F\frac{1}{\lambda^2} + \left(K - 2FT_{th_1}\frac{1-p_1}{p_1}\right)\frac{1}{\lambda} + \left(L - KT_{th_1}\frac{1-p_1}{p_1} - FT_{th_1}^2\frac{1-p_1}{p_1}\right) \end{aligned} \tag{A.98}$$

The average time of channel occupation is

$$\overline{C_1} = \overline{T_{pre1}} + T_{data} + T_{ack}. \tag{A.99}$$

The average response time of a unicast packet is

$$\overline{D_1} = 0.5T_{cycle} + 2\theta\left(\frac{1}{\lambda} - \frac{1-p_1}{p_1}T_{th_1}\right) + T_{sample} + T_{s\_fpga} + T_{s\_rf} + T_{data} + T_{s\_\mu c} + T_t + T_{ack}. \tag{A.100}$$

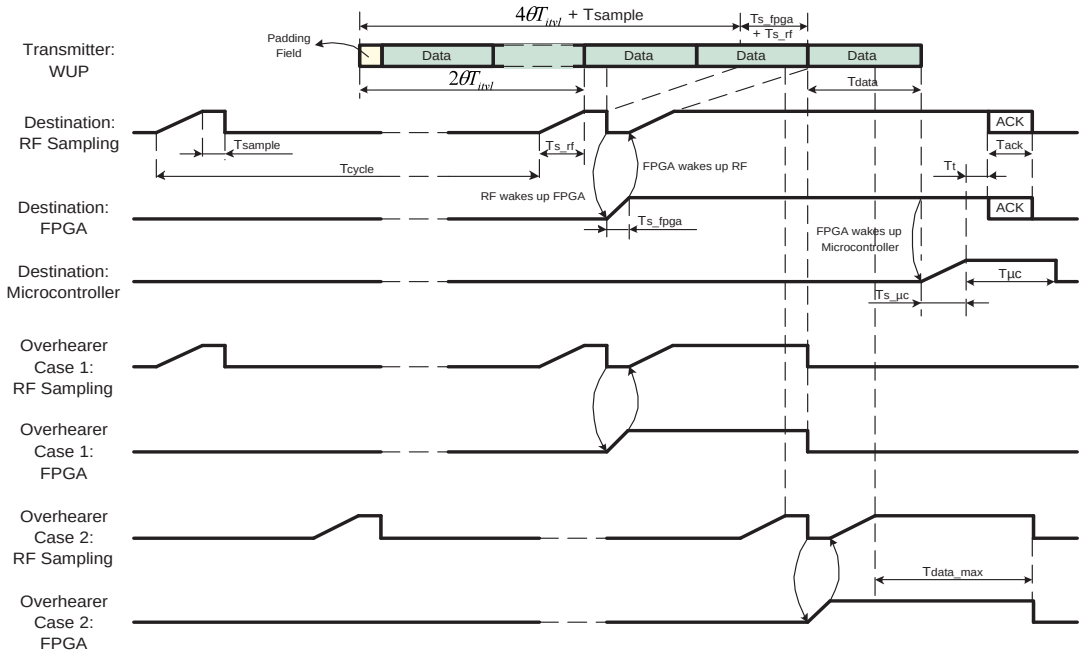


Figure A.7: The WiseMAC+Rep Detail (Case 2)

**Case 2:** $T_{th_1} \leq interval < T_{th_2}$

In this case, the detailed procedures of the WiseMAC+Rep scheme are shown in Fig. A.7.

The length of the shortened preamble is still

$$T_{pre2} = 4\theta T_{itvl} + T_{sample} + T_{s\_fpga} + T_{s\_rf}, \tag{A.101}$$

where $T_{itvl}$ is the Poisson interarrival interval and its expectation is

$$\begin{aligned}
\overline{T_{itvl}} &= E\left(T_{itvl}\right) \\
&= \int_{T_{th_1}}^{T_{th_2}} \frac{\lambda}{p_2} t e^{-\lambda t} dt \\
&= \frac{\lambda}{p_2}\left(\frac{1}{-\lambda} t e^{-\lambda t}\Big|_{T_{th_1}}^{T_{th_2}} - \frac{1}{-\lambda}\int_{T_{th_1}}^{T_{th_2}} e^{-\lambda t} dt\right) \\
&= \frac{\lambda}{p_2}\left(\frac{1}{-\lambda}\left(T_{th_2} e^{-\lambda T_{th_2}} - T_{th_1} e^{-\lambda T_{th_1}}\right) - \frac{1}{-\lambda}\frac{1}{-\lambda}\, e^{-\lambda t}\Big|_{T_{th_1}}^{T_{th_2}}\right) \\
&= \frac{\lambda}{p_2}\left(\frac{1}{-\lambda}\left(T_{th_2} e^{-\lambda T_{th_2}} - T_{th_1} e^{-\lambda T_{th_1}}\right) - \frac{1}{\lambda^2}\left(e^{-\lambda T_{th_2}} - e^{-\lambda T_{th_1}}\right)\right) \\
&= \frac{1}{p_2}\left(T_{th_1}(1-p_1) - T_{th_2} p_3 + \frac{1}{\lambda} p_2\right) \\
&= \frac{1}{\lambda} + \frac{1-p_1}{p_2} T_{th_1} - \frac{p_3}{p_2} T_{th_2}.
\end{aligned} \tag{A.102}$$

So the mean of $T_{pre2}$ is

$$\overline{T_{pre2}} = 4\theta\left(\frac{1}{\lambda} + \frac{1-p_1}{p_2}T_{th_1} - \frac{p_3}{p_2}T_{th_2}\right) + T_{sample} + T_{s\_fpga} + T_{s\_rf}. \tag{A.103}$$

The average power consumption of receiving one unicast packet for the destination device is the same as Equ. A.63

$$\begin{aligned}\overline{P_{dst2}} &= \hat{P}_{rf\_rx}(2\theta T_{itvl} + T_{s\_fpga} + T_{s\_rf} - T_{s\_fpga} + T_{data} + T_{s\_\mu c}) \\ &\quad + \hat{P}_{rf\_tx}(T_t + T_{ack}) \\ &\quad + \hat{P}_{fpga}(2\theta T_{itvl} + T_{s\_fpga} + T_{s\_rf} + T_{data} + T_{s\_\mu c} + T_t + T_{ack}) \\ &\quad + \hat{P}_{\mu c}(T_{s\_\mu c} + T_{\mu c}) \\ &= AT_{itvl} + B.\end{aligned} \tag{A.104}$$

For an overhearer, there are two cases. In the first case, the overhearer is woken up by the first $(4\theta T_{itvl} + T_{sample})$ part of $T_{pre2}$, so the overhearer can receive the data frame and make decision according to the destination address, as the overhearer case 1 shown in Fig. A.7. The probability of this case is

$$p_{ovh2\_1} = \frac{4\theta T_{itvl} + T_{sample}}{T_{pre2} + T_{data}}, \tag{A.105}$$

and the average power consumption in this case is

$$\begin{aligned}\overline{P_{ovh2\_1}} &= \hat{P}_{rf\_rx}(T_{s\_rf} + 1.5T_{data}) \\ &\quad + \hat{P}_{fpga}(T_{s\_fpga} + T_{s\_rf} + 1.5T_{data}),\end{aligned} \tag{A.106}$$

which is not a function of $T_{itvl}$.

In the second case, the overhearer is woken up by the last $(T_{s\_fpga} + T_{s\_rf})$ part of $T_{pre2}$ or by $T_{data}$, so the overhearer cannot receive the data frame. As a result, it has to wait for at least one $T_{data\_max}$ if nothing is received after being woken up, as the overhearer case 2 shown in Fig. A.7. The probability of this case is

$$p_{ovh2\_2} = \frac{T_{s\_fpga} + T_{s\_rf} + T_{data}}{T_{pre2} + T_{data}}, \tag{A.107}$$

and the power consumption in this case is

$$P_{ovh2\_2} = \hat{P}_{rf\_rx}(T_{s\_rf} + T_{data\_max}) + \hat{P}_{fpga}(T_{s\_fpga} + T_{s\_rf} + T_{data\_max}), \tag{A.108}$$

which is also not a function of $T_{itvl}$.

So the average power consumption of receiving one packet for an overhearer is

$$\begin{aligned}\overline{P_{ovh2}} &= \overline{P_{ovh2\_1}} \cdot p_{ovh2\_1} + P_{ovh2\_2} \cdot p_{ovh2\_2} \\ &= \overline{P_{ovh2\_1}}\frac{4\theta T_{itvl} + T_{sample}}{T_{pre2} + T_{data}} + P_{ovh2\_2}\frac{T_{s\_fpga} + T_{s\_rf} + T_{data}}{T_{pre2} + T_{data}}.\end{aligned} \tag{A.109}$$

Since the preamble is shortened, only a part of the slaves is woken up to become overhearers. The probability that a slave is woken up is $\frac{T_{pre2}+T_{data}}{T_{cycle}}$. So the average power consumption of receiving one packet for each slave in case 2 is

$$\begin{aligned}
\overline{P'_{pkt2}} &= \overline{P_{dst2}}\frac{1}{N} + \overline{P_{ovh2}}\frac{T_{pre2}+T_{data}}{T_{cycle}}\frac{N-1}{N} \\
&= (AT_{itvl}+B)\frac{1}{N} + \left(\overline{P_{ovh2_1}}\frac{4\theta T_{itvl}+T_{sample}}{T_{cycle}} + \overline{P_{ovh2_2}}\frac{T_{s_fpga}+T_{s_rf}+T_{data}}{T_{cycle}}\right)\frac{N-1}{N} \\
&= \underbrace{\left(\frac{A}{N} + \overline{P_{ovh2_1}}\frac{4\theta(N-1)}{NT_{cycle}}\right)}_{M} T_{itvl} + \underbrace{\frac{B}{N} + \left(\overline{P_{ovh2_1}}\frac{T_{sample}}{T_{cycle}} + \overline{P_{ovh2_2}}\frac{T_{s_fpga}+T_{s_rf}+T_{data}}{T_{cycle}}\right)\frac{N-1}{N}}_{O} \\
&= MT_{itvl} + O, \qquad \text{(A.110)}
\end{aligned}$$

and its expectation is the average power consumption of receiving one packet for each slave

$$\begin{aligned}
\overline{P_{pkt2}} &= E\left(\overline{P'_{pkt2}}\right) \\
&= M\overline{T_{itvl}} + O \\
&= M\left(\frac{1}{\lambda} + \frac{1-p_1}{p_2}T_{th_1} - \frac{p_3}{p_2}T_{th_2}\right) + O. \qquad \text{(A.111)}
\end{aligned}$$

The average time of channel occupation is

$$\overline{C_2} = \overline{T_{pre2}} + T_{data} + T_{ack}. \qquad \text{(A.112)}$$

The average response time of a unicast packet is

$$\overline{D_2} = 0.5T_{cycle} + 2\theta\left(\frac{1}{\lambda} + \frac{1-p_1}{p_2}T_{th_1} - \frac{p_3}{p_2}T_{th_2}\right) + T_{sample} + T_{s\_fpga} + T_{s\_rf} + T_{data} + T_{s\_\mu c} + T_t + T_{ack}. \qquad \text{(A.113)}$$

### Case 3: $interval \geq T_{th_2}$

When the interval is larger than $T_{th_2}$, the length of the preamble is the same as in the noraml Rep scheme, so Equ. A.29 and Equ. A.34 from the Rep scheme can be reused.

$$T_{pre3} = T_{pre_{rep}} = T_{cycle} + T_{sample} + T_{s\_fpga} + T_{s\_rf}. \qquad \text{(A.114)}$$

The average power consumption of receiving one packet for each slave in case 3 is

$$\overline{P_{pkt3}} = \overline{P_{pkt_{rep\_u}}}. \qquad \text{(A.115)}$$

The time of channel occupation is

$$C_3 = T_{pre3} + T_{data} + T_{ack}. \qquad \text{(A.116)}$$

The response time of a unicast packet is

$$D_3 = T_{pre3} + T_{data} + T_{s\_\mu c} + T_t + T_{ack}. \qquad \text{(A.117)}$$

### Overall

All in all, the average power consumption of receiving one packet for each slave in the WiseMAC+Rep is

$$\overline{P_{pkt_{WiseMAC+REP\_U}}} = \overline{P_{pkt1}} \cdot p_1 + \overline{P_{pkt2}} \cdot p_2 + \overline{P_{pkt3}} \cdot p_3. \tag{A.118}$$

So the power consumption of the device for a Poisson arrival rate of $\lambda$ for each slave is

$$P_{WiseMAC+REP\_U} = \overline{P_{pkt_{WiseMAC+REP\_U}}} \cdot \lambda N + P_{no\_traffic}. \tag{A.119}$$

The overall percentage of channel occupation is

$$C_{WiseMAC+REP\_U} = (\overline{C_1} \cdot p_1 + \overline{C_2} \cdot p_2 + C_3 \cdot p_3) \cdot \lambda N. \tag{A.120}$$

The average response time of a unicast packet is

$$\overline{D_{WiseMAC+REP\_U}} = \overline{D_1} \cdot p_1 + \overline{D_2} \cdot p_2 + D_3 \cdot p_3. \tag{A.121}$$

Note that an ACK can also wake up overhearers in the WiseMAC+Rep consuming more power, which is not taken into consideration in the analytical calculation. However, the degradation caused by the ACK is supposed to be not very significant, because first the ACK is very short, and second not each slave can hear an ACK from another slave. So it is reasonable to neglect this effect in the analytical computation for the reason of simplicity.

## A.6.2 Broadcast

The same as the broadcast in the Rep scheme, see Sec. A.3.2.

# A.7 The WiseMAC+WUF Scheme

## A.7.1 Unicast

When the length of the WUS is less than the length of an SWUF, a threshold for the Poisson interarrival interval is defined by

$$T_{th_1} = \frac{T_{swuf}}{4\theta}. \tag{A.122}$$

Under a certain threshold of the interval, the WiseMAC enables a shortened wake-up-preamble, which is denoted by

$$T_{th_2} = \frac{T_{cycle}}{4\theta}. \tag{A.123}$$

The probability to generate an interval that is less than $T_{th_1}$ (case 1) is

$$p_1 = p(t < T_{th_1}) = f(T_{th_1}) = 1 - e^{-\lambda T_{th_1}}. \tag{A.124}$$

The probability that it is larger than $T_{th_1}$ but less than $T_{th_2}$ (case 2) is

$$p_2 = p(T_{th_1} \leq t < T_{th_2}) = f(T_{th_2}) - f(T_{th_1}) = e^{-\lambda T_{th_1}} - e^{-\lambda T_{th_2}}. \tag{A.125}$$

Finally, the probability that the interval is larger than $T_{th_2}$ (case 3) is

$$p_3 = p(t \geq T_{th_2}) = 1 - f(T_{th_2}) = e^{-\lambda T_{th_2}}. \tag{A.126}$$

**Case 1:** $interval < T_{th_1}$

In this case, the detailed procedures of the WiseMAC+WUF scheme are shown in Fig. A.8.

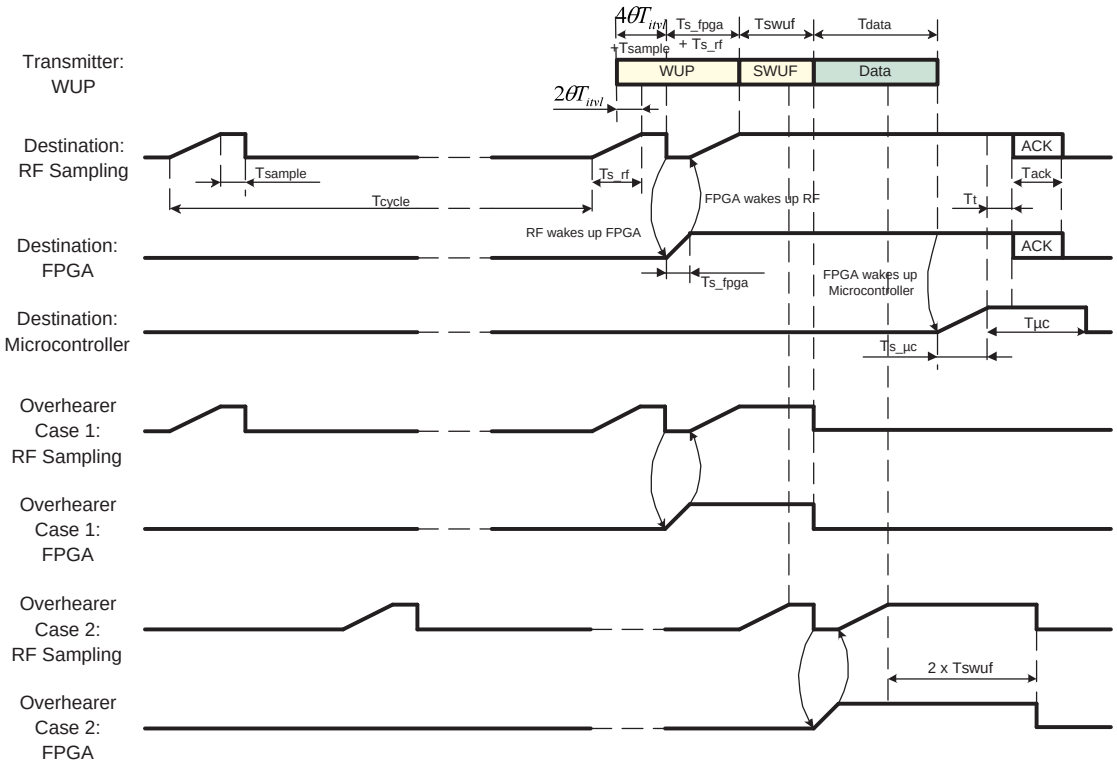


Figure A.8: The WiseMAC+WUF Scheme Detail (Case 1)

The length of the shortened preamble is

$$T_{pre1} = 4\theta T_{itvl} + T_{sample} + T_{s\_fpga} + T_{s\_rf} + T_{swuf}, \tag{A.127}$$

where $T_{itvl}$ is the Poisson inter-arrival interval and its expectation is given by Equ. A.61

$$\overline{T_{itvl}} = \frac{1}{\lambda} - \frac{1-p_1}{p_1} T_{th_1}. \tag{A.128}$$

So the mean of $T_{pre1}$ is

$$\overline{T_{pre1}} = 4\theta \left( \frac{1}{\lambda} - \frac{1-p_1}{p_1} T_{th_1} \right) + T_{sample} + T_{s\_fpga} + T_{s\_rf} + T_{swuf}. \tag{A.129}$$

The average power consumption of receiving one unicast packet for the destination device is similar to Equ. A.63 except for some small modifications

$$\begin{aligned} \overline{P_{dst1}} &= \hat{P}_{rf\_rx}(2\theta T_{itvl} + T_{s\_fpga} + T_{s\_rf} - T_{s\_fpga} + T_{swuf} + T_{data} + T_{s\_\mu c}) \\ &\quad + \hat{P}_{rf\_tx}(T_t + T_{ack}) \\ &\quad + \hat{P}_{fpga}(2\theta T_{itvl} + T_{s\_fpga} + T_{s\_rf} + T_{swuf} + T_{data} + T_{s\_\mu c} + T_t + T_{ack}) \\ &\quad + \hat{P}_{\mu c}(T_{s\_\mu c} + T_{\mu c}) \\ &\doteq AT_{itvl} + R. \end{aligned} \tag{A.130}$$

For an overhearer, there are two cases. In the first case, the overhearer is woken up by the first $(4\theta T_{itvl} + T_{sample})$ part of $T_{pre1}$, so the overhearer can receive the SWUF and make decision according to the destination address, as the overhearer case 1 shown in Fig. A.8. The probability of this case is

$$p_{ovh1\_1} = \frac{4\theta T_{itvl} + T_{sample}}{T_{pre1}}, \tag{A.131}$$

and the average power consumption in this case is

$$\begin{aligned} \overline{P_{ovh1\_1}} &= \hat{P}_{rf\_rx}(2\theta T_{itvl} + T_{s\_fpga} + T_{s\_rf} - T_{s\_fpga} + T_{swuf}) \\ &\quad + \hat{P}_{fpga}(2\theta T_{itvl} + T_{s\_fpga} + T_{s\_rf} + T_{swuf}) \\ &= \underbrace{2\theta(\hat{P}_{rf\_rx} + \hat{P}_{fpga})}_{A} T_{itvl} \\ &\quad + \underbrace{\hat{P}_{rf\_rx}(T_{s\_rf} + T_{swuf}) + \hat{P}_{fpga}(T_{s\_fpga} + T_{s\_rf} + T_{swuf})}_{S} \\ &= AT_{itvl} + S. \end{aligned} \tag{A.132}$$

In the second case, the overhearer is woken up by the last $(T_{s\_fpga} + T_{s\_rf} + T_{swuf})$ part of $T_{pre1}$, so the overhearer cannot receive the SWUF but has to wait for at least $2T_{swuf}$ before returns to sleep, as the overhearer case 2 shown in Fig. A.8. The probability of this case is

$$p_{ovh1\_2} = \frac{T_{s\_fpga} + T_{s\_rf} + T_{swuf}}{T_{pre1}}, \tag{A.133}$$

and the power consumption in this case is

$$P_{ovh1\_2} = \hat{P}_{rf\_rx}(T_{s\_rf} + 2T_{swuf}) + \hat{P}_{fpga}(T_{s\_fpga} + T_{s\_rf} + 2T_{swuf}), \tag{A.134}$$

which is not a function of $T_{itvl}$.

So the average power consumption for an overhearer is

$$\begin{aligned} \overline{P_{ovh1}} &= \overline{P_{ovh1\_1}} \cdot p_{ovh1\_1} + P_{ovh1\_2} \cdot p_{ovh1\_2} \\ &= \overline{P_{ovh1\_1}} \frac{4\theta T_{itvl} + T_{sample}}{T_{pre1}} + P_{ovh1\_2} \frac{T_{s\_fpga} + T_{s\_rf} + T_{swuf}}{T_{pre1}}. \end{aligned} \tag{A.135}$$

Since the preamble is shortened, only a part of the slaves is woken up to become overhearers. The probability that a slave is woken up is $\frac{T_{pre1}}{T_{cycle}}$. So the power consumption of receiving one packet for each slave in case 1 is

$$\begin{aligned} \overline{P'_{pkt1}} &= \overline{P_{dst1}} \frac{1}{N} + \overline{P_{ovh1}} \frac{T_{pre1}}{T_{cycle}} \frac{N-1}{N} \\ &= (AT_{itvl} + R)\frac{1}{N} + \left((AT_{itvl} + S)\frac{4\theta T_{itvl} + T_{sample}}{T_{cycle}} + \overline{P_{ovh1\_2}} \frac{T_{s\_fpga} + T_{s\_rf} + T_{swuf}}{T_{cycle}}\right) \frac{N-1}{N} \\ &= \frac{A}{N} T_{itvl} + \frac{R}{N} + \left(\frac{A}{N} T_{itvl} + \frac{S}{N}\right) \underbrace{4\theta \frac{N-1}{T_{cycle}}}_{D} T_{itvl} + A \frac{T_{sample}}{T_{cycle}} \frac{N-1}{N} T_{itvl} \\ &\quad + \underbrace{\left(S \frac{T_{sample}}{T_{cycle}} + \overline{P_{ovh1\_2}} \frac{T_{s\_fpga} + T_{s\_rf} + T_{swuf}}{T_{cycle}}\right) \frac{N-1}{N}}_{U} \\ &= \frac{A}{N} T_{itvl} + \frac{R}{N} + \frac{AD}{N} T_{itvl}^2 + \frac{SD}{N} T_{itvl} + A \frac{T_{sample}}{T_{cycle}} \frac{N-1}{N} T_{itvl} + U \\ &= \underbrace{\frac{AD}{N}}_{F} T_{itvl}^2 + \underbrace{\left(\frac{A + SD}{N} + A \frac{T_{sample}}{T_{cycle}} \frac{N-1}{N}\right)}_{V} T_{itvl} + \underbrace{\frac{R}{N} + U}_{W} \\ &= FT_{itvl}^2 + VT_{itvl} + W. \end{aligned} \tag{A.136}$$

The average power consumption of receiving one packet for each slave is the expectation of $\overline{P'_{pkt1}}$. According to Equ. A.70,

$$\begin{aligned} \overline{P_{pkt1}} &= E\left(\overline{P'_{pkt1}}\right) \\ &= \int_0^{T_{th_1}} (Ft^2 + Vt + W) \frac{p(t)}{p_1} dt \\ &= \int_0^{T_{th_1}} (Ft^2 + Vt + W) \frac{\lambda}{p_1} e^{-\lambda t} dt \\ &= 2F\frac{1}{\lambda^2} + \left(V - 2FT_{th_1}\frac{1-p_1}{p_1}\right)\frac{1}{\lambda} + \left(W - VT_{th_1}\frac{1-p_1}{p_1} - FT_{th_1}^2\frac{1-p_1}{p_1}\right). \end{aligned} \tag{A.137}$$

The average time of channel occupation is

$$\overline{C_1} = \overline{T_{pre1}} + T_{data} + T_{ack}. \tag{A.138}$$

The average response time of a unicast packet is

$$\overline{D_1} = 0.5T_{cycle} + 2\theta\left(\frac{1}{\lambda} - \frac{1-p_1}{p_1} T_{th_1}\right) + T_{sample} + T_{s\_fpga} + T_{s\_rf} + T_{data} + T_{s\_\mu c} + T_t + T_{ack}. \tag{A.139}$$

**Case 2:** $T_{th_1} \leq interval < T_{th_2}$

In this case, the detailed procedures of the WiseMAC+WUF scheme are shown in Fig. A.9.

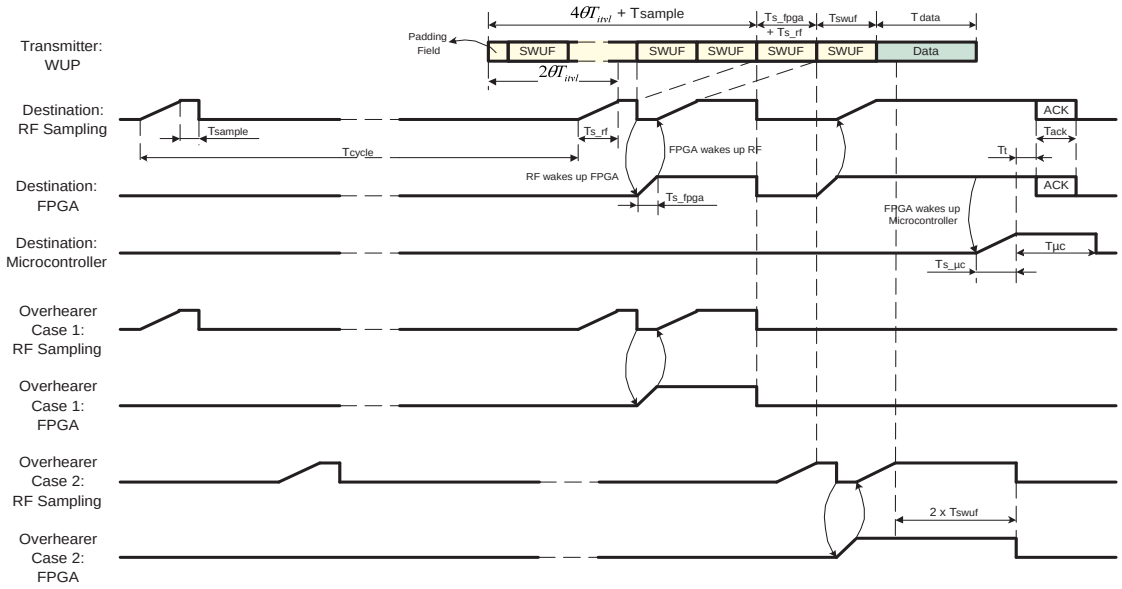


Figure A.9: The WiseMAC+WUF Scheme Detail (Case 2)

The length of the shortened preamble is

$$T_{pre2} = 4\theta T_{itvl} + T_{sample} + T_{s\_fpga} + T_{s\_rf} + T_{swuf}. \tag{A.140}$$

where $T_{itvl}$ is the Poisson inter-arrival interval and its expectation is given by Equ. A.102

$$\overline{T_{itvl}} = E\left(T_{itvl}\right) = \frac{1}{\lambda} + \frac{1-p_1}{p_2} T_{th_1} - \frac{p_3}{p_2} T_{th_2}. \tag{A.141}$$

So the mean of $T_{pre2}$ is

$$\overline{T_{pre2}} = 4\theta \left( \frac{1}{\lambda} + \frac{1-p_1}{p_2} T_{th_1} - \frac{p_3}{p_2} T_{th_2} \right) + T_{sample} + T_{s\_fpga} + T_{s\_rf} + T_{swuf}. \tag{A.142}$$

The average power consumption of receiving one unicast packet for the destination is

$$\begin{aligned} \overline{P_{dst2}} \quad = \quad & \hat{P}_{rf\_rx}(T_{s\_rf} + 1.5T_{swuf} + T_{s\_rf} + T_{data} + T_{s\_\mu c}) \\ & + \hat{P}_{rf\_tx}(T_t + T_{ack}) \\ & + \hat{P}_{fpga}(T_{s\_fpga} + T_{s\_rf} + 1.5T_{swuf} + T_{s\_fpga} + T_{s\_rf} + T_{data} + T_{s\_\mu c} + T_t + T_{ack}) \\ & + \hat{P}_{\mu c}(T_{s\_\mu c} + T_{\mu c}) \end{aligned} \tag{A.143}$$

which is not a function of $T_{itvl}$.

For an overhearer, there are two cases. In the first case, the overhearer is woken up by the first $(4\theta T_{itvl} + T_{sample})$ part of $T_{pre2}$, so the overhearer can receive the SWUF and make decision according to the destination address, as the overhearer case 1 shown in Fig. A.9. The probability of this case is

$$p_{ovh2\_1} = \frac{4\theta T_{itvl} + T_{sample}}{T_{pre2}}, \tag{A.144}$$

and the average power consumption in this case is

$$\overline{P_{ovh2\_1}} = \hat{P}_{rf\_rx}(T_{s\_rf} + 1.5T_{swuf}) + \hat{P}_{fpga}(T_{s\_fpga} + T_{s\_rf} + 1.5T_{swuf}), \tag{A.145}$$

which is not a function of $T_{itvl}$.

In the second case, the overhearer is woken up by the last $(T_{s\_fpga} + T_{s\_rf} + T_{swuf})$ part of $T_{pre2}$, so the overhearer cannot receive the SWUF. As a result, it has to wait at leat two $T_{swuf}$ if nothing is received after being woken up, as the overhearer case 2 shown in Fig. A.9. The probability of this case is

$$p_{ovh2\_2} = \frac{T_{s\_fpga} + T_{s\_rf} + T_{swuf}}{T_{pre2}}, \tag{A.146}$$

and the power consumption in this case is

$$P_{ovh2\_2} = \hat{P}_{rf\_rx}(T_{s\_rf} + 2T_{swuf}) + \hat{P}_{fpga}(T_{s\_fpga} + T_{s\_rf} + 2T_{swuf}), \tag{A.147}$$

which is also not a function of $T_{itvl}$.

So the average power consumption for an overhearer is

$$\begin{aligned}\overline{P_{ovh2}} &= \overline{P_{ovh2\_1}} \cdot p_{ovh2\_1} + P_{ovh2\_2} \cdot p_{ovh2\_2} \\ &= \overline{P_{ovh2\_1}} \frac{4\theta T_{itvl} + T_{sample}}{T_{pre2}} + P_{ovh2\_2} \frac{T_{s\_fpga} + T_{s\_rf} + T_{swuf}}{T_{pre2}}.\end{aligned} \tag{A.148}$$

Since the preamble is shortened, only a part of the slaves is woken up to become overhearers. The probability that a slave is woken up is $\frac{T_{pre2}}{T_{cycle}}$. So the power consumption of receiving one packet for each slave in case 2 is

$$\begin{aligned}\overline{P'_{pkt2}} &= \overline{P_{dst2}} \frac{1}{N} + \overline{P_{ovh2}} \frac{T_{pre2}}{T_{cycle}} \frac{N-1}{N} \\ &= \overline{P_{dst2}} \frac{1}{N} + \left( \overline{P_{ovh2\_1}} \frac{4\theta T_{itvl} + T_{sample}}{T_{cycle}} + \overline{P_{ovh2\_2}} \frac{T_{s\_fpga} + T_{s\_rf} + T_{swuf}}{T_{cycle}} \right) \frac{N-1}{N} \\ &= \underbrace{\overline{P_{ovh2\_1}} \frac{4\theta(N-1)}{NT_{cycle}}}_{P} T_{itvl} + \underbrace{\overline{P_{dst2}} \frac{1}{N} + \left( \overline{P_{ovh2\_1}} \frac{T_{sample}}{T_{cycle}} + \overline{P_{ovh2\_2}} \frac{T_{s\_fpga} + T_{s\_rf} + T_{swuf}}{T_{cycle}} \right) \frac{N-1}{N}}_{Q} \\ &= PT_{itvl} + Q,\end{aligned} \tag{A.149}$$

and its expectation is the average power consumption of receiving one packet for each slave

$$\begin{aligned}\overline{P_{pkt2}} &= E\left(\overline{P'_{pkt2}}\right) \\ &= P\overline{T_{itvl}} + Q \\ &= P\left(\frac{1}{\lambda} + \frac{1-p_1}{p_2} T_{th_1} - \frac{p_3}{p_2} T_{th_2}\right) + Q.\end{aligned} \tag{A.150}$$

The average time of channel occupation is

$$\overline{C_2} = \overline{T_{pre2}} + T_{data} + T_{ack}. \tag{A.151}$$

The average response time of a unicast packet is

$$\overline{D_2} = 0.5T_{cycle} + 2\theta\left(\frac{1}{\lambda} + \frac{1-p_1}{p_2}T_{th_1} - \frac{p_3}{p_2}T_{th_2}\right) + T_{sample} + T_{s\_fpga} + T_{s\_rf} + T_{data} + T_{s\_\mu c} + T_t + T_{ack}. \tag{A.152}$$

**Case 3:** $interval \geq T_{th_2}$

When the interval is larger than $T_{th_2}$, the length of the preamble is the same as in the normal WUF scheme, so Equ. A.42 and Equ. A.47 from the WUF scheme can be reused.

$$T_{pre3} = T_{pre_{wuf}} = T_{cycle} + T_{sample} + T_{s\_fpga} + T_{s\_rf} + T_{swuf}. \tag{A.153}$$

The average power consumption of receiving one packet for each slave in case 3 is

$$\overline{P_{pkt3}} = \overline{P_{pkt_{wuf\_u}}}. \tag{A.154}$$

The time of channel occupation is

$$C_3 = T_{pre3} + T_{data} + T_{ack}. \tag{A.155}$$

The response time of a unicast packet is

$$D_3 = T_{pre3} + T_{data} + T_{s\_\mu c} + T_t + T_{ack}. \tag{A.156}$$

**Overall**

All in all, the average power consumption of receiving one packet for each slave in the WiseMAC+WUF is

$$\overline{P_{pkt_{WiseMAC+WUF\_U}}} = \overline{P_{pkt1}} \cdot p_1 + \overline{P_{pkt2}} \cdot p_2 + \overline{P_{pkt3}} \cdot p_3. \tag{A.157}$$

So the power consumption of the device for a Poisson arrival rate of $\lambda$ for each slave is

$$P_{WiseMAC+WUF\_U} = \overline{P_{pkt_{WiseMAC+WUF\_U}}} \cdot \lambda N + P_{no\_traffic}. \tag{A.158}$$

The overall percentage of channel occupation is

$$C_{WiseMAC+WUF\_U} = (\overline{C_1} \cdot p_1 + \overline{C_2} \cdot p_2 + C_3 \cdot p_3) \cdot \lambda N \cdot 100\,[\%]. \tag{A.159}$$

The average response time of a unicast packet is

$$\overline{D_{WiseMAC+WUF\_U}} = \overline{D_1} \cdot p_1 + \overline{D_2} \cdot p_2 + D_3 \cdot p_3. \tag{A.160}$$

### A.7.2 Broadcast

The same as the broadcast in the WUF scheme, see Sec. A.4.2.

## A.8 The Ideal Preamble Sampling

In an ideal case, the master always knows exactly the sampling time of each slave. So the master can send a very short WUP, e.g. the length of the sampling time $T_{sample}$. This ideal case is used to show the best situation that a preamble sampling based scheme can reach. The length of the WUP is

$$T_{pre_{ideal}} = T_{sample}. \tag{A.161}$$

### A.8.1 Unicast

The power consumption of receiving one unicast packet for the destination device is

$$\begin{aligned} P_{dst_{ideal\_u}} &= \hat{P}_{rf\_rx}(T_{s\_rf} + T_{data} + T_{s\_\mu c}) \\ &\quad + \hat{P}_{rf\_tx}(T_t + T_{ack}) \\ &\quad + \hat{P}_{fpga}(T_{s\_fpga} + T_{s\_rf} + T_{data} + T_{s\_\mu c} + T_t + T_{ack}) \\ &\quad + \hat{P}_{\mu c}(T_{s\_\mu c} + T_{\mu c}), \end{aligned} \tag{A.162}$$

and the power consumption of an overhearer is

$$P_{ovh_{ideal\_u}} = \hat{P}_{rf\_rx}(T_{s\_rf} + T_{to\_addr}) + \hat{P}_{fpga}(T_{s\_fpga} + T_{s\_rf} + T_{to\_addr}). \tag{A.163}$$

The average power consumption of receiving one unicast packet for each slave is

$$\overline{P_{pkt_{ideal\_u}}} = P_{dst_{ideal\_u}} \frac{1}{N} + P_{ovh_{ideal\_u}} \frac{T_{pre_{ideal}}}{T_{cycle}} \frac{N-1}{N}. \tag{A.164}$$

So the power consumption of the device for a Poisson arrival rate of $\lambda$ to each slave is

$$P_{Ideal\_U} = \overline{P_{pkt_{ideal\_u}}} \cdot \lambda N + P_{no\_traffic}. \tag{A.165}$$

The percentage of channel occupation is

$$C_{Ideal\_U} = (T_{pre_{ideal}} + T_{data} + T_{ack}) \cdot \lambda N \cdot 100\,[\%]. \tag{A.166}$$

The average response time of a unicast packet is

$$\overline{D_{Ideal\_U}} = 0.5T_{cycle} + T_{sample} + T_{s\_fpga} + T_{s\_rf} + T_{data} + T_{s\_\mu c} + T_t + T_{ack}. \tag{A.167}$$

### A.8.2 Broadcast

There is no ideal case in broadcast, because sending a very long WUS is the only way to wake up all slaves.

## A.9 DRD vs. RSSI

In this section, we will compute the power consumption difference between using the DRD and the only RSSI as the sampling criteria, see Sec. 4.1.

First, the power consumption of using DRD in no traffic case is denoted by Equ. A.6.

Second, if only RSSI is used as the sampling criteria, slaves can overhear noise and interference causing the unnecessary wake-up, wasting power. The average power consumption of each unnecessary wake-up in the WUF scheme is

$$\begin{aligned} P_{ovh_{rssi}} &= \hat{P}_{rf\_rx}(T_{s\_rf} + 2T_{swuf}) \\ &\quad + \hat{P}_{fpga}(T_{s\_fpga} + T_{s\_rf} + 2T_{swuf}). \end{aligned} \tag{A.168}$$

It is assumed that when a slave is woken up, if it cannot receive anything within the transmitting time of two SWUFs, it goes back to sleep.

The sampling time of using simple RSSI is shorter than that of the DRD, which is denoted by $T_{sample\_rssi}$. If the percentage of the occupied channel by the noise and interference is $C_{percent}$, the power consumption of using the simple RSSI in the no traffic case is

$$\begin{aligned} P_{rssi_{no\_traffic}} &= \hat{P}_{rf\_rx}\frac{T_{s\_rf} + T_{sample\_rssi}}{T_{cycle}} + P_{sleep} \\ &\quad + \overline{P_{ovh_{rssi}}}\frac{C_{percent}}{T_{cycle}}. \end{aligned} \tag{A.169}$$

# Appendix B

# Basic Parameters

Table B.1: Basic Parameters of the Sindrion System

| Parameter | Description | Value |
|---|---|---|
| $U_{system}$ | Supply voltage of the Sindrion system | $3\,V$ |
| $B_{capacity}$ | Capacity of the battery | $1000\,mAh$ |
| $f_{\mu C}$ | Working frequency of the microcontroller | $9\,MHz$ |
| $f_{fpga}$ | Working frequency of the FPGA | $f_{\mu C}$ |
| $\theta$ | Frequency tolerance of the quartz used in the WiseMAC. | $30\,ppm$ |
| $P_{rf\_sleep}$ | Power consumption of the RF in sleep mode | $27\,\mu W$ |
| $P_{fpga\_sleep}$ | Power consumption of the FPGA in sleep mode | $1.5\,\mu W$ |
| $P_{\mu c\_sleep}$ | Power consumption of the microcontroller in sleep mode | $90\,\mu W$ |
| $P_{others}$ | Power consumption of other components: memory, RTC, etc. | $27\,\mu W$ |
| $P_{rf\_rx}$ | Power consumption of the RF in RX mode | $27\,mW$ |
| $P_{rf\_tx}$ | Power consumption of the RF in TX mode | $39.9\,mW$ |
| $P_{fpga}$ | Power consumption of the FPGA in working mode | $20.4\,mW$ |
| $P_{\mu c}$ | Power consumption of the microcontroller in working mode | $32.7\,mW$ |
| **Continued** . . . | | |

Table B.1: (continued)

| Parameter | Description | Value |
|---|---|---|
| $T_{to\_rx}$ | Setup time of the RF RX mode | $2.2\,ms$ |
| $T_{to\_tx}$ | Setup time of the RF TX mode | $1.1\,ms$ |
| $T_t$ | Transition time of the RF between RX and TX mode | $T_{to\_rx}$ |
| $T_{s\_rssi}$ | Setup time of the RF RSSI | $0.4\,ms$ |
| $T_{s\_rf}$ | Setup time of the RF from sleep to awake | $T_{to\_rx}$ |
| $T_{s\_fpga}$ | Setup time of the FPGA from sleep to awake | $0.1\,ms$ |
| $T_{s\_\mu c}$ | Setup time of the microcontroller from sleep to awake | $2.6\,ms$ |
| $T_{sample}$ | Time for sampling the channel if DRD is used | $0.8\,ms$ |
| $T_{sample_{rssi}}$ | Time for sampling the channel if only RSSI is used | $0.4\,ms$ |
| $T_{cycle}$ | Time of the RF self-polling cycle | - |
| $T_{\mu c}$ | Time for the microcontroller to process a data packet | $3\,ms$ |
| $R_{data}$ | Transmission rate for DATA frame | $100\,bit/s$ |
| $R_{wus}$ | Transmission rate for wake-up-signal | $70\,bit/s$ |
| $C_{data}$ | Coding ratio for modulation and ECC code of DATA | 1.5 |
| $C_{wus}$ | Coding ratio for modulation and ECC code of wake-up-frame | 2.7 |
| $L_{ack}$ | Length of ACK frame | $96\,bit$ |
| $L_{swuf}$ | Length of a short wake-up-frame | $326\,bit$ |
| $L_{ip\_max}$ | Maximum length of IP packet | $576 \times 8\,bit$ |
| $L_{data\_max}$ | Maximum length of DATA frame | $7200\,bit$ |
| $L_{bcn}$ | Length of an IEEE 802.15.4 MAC beacon frame | $276\,bit$ |
| $L_{bcn\_extra}$ | Length of an IEEE 802.15.4 frame indication in beacon | $96\,bit$ |
| $L_{cmd}$ | Length of an IEEE 802.15.4 MAC data request command frame | $228\,bit$ |
| $L_{to\_addr}$ | Length of a frame until the destination address | $192\,bit$ |
| $T_{ack}$ | Time to transmit an ACK frame | $0.96\,ms$ |
| $T_{swuf}$ | Time to transmit an SWUF frame | $4.66\,ms$ |
| **Continued** . . . | | |

Table B.1: (continued)

| **Parameter** | **Description** | **Value** |
|---|---|---|
| $T_{data}$ | Time to transmit a DATA frame | - |
| $T_{data\_max}$ | Time to transmit a DATA frame with maximal length | $72\,ms$ |
| $T_{bcn}$ | Time to transmit an IEEE 802.15.4 beacon frame | $2.76\,ms$ |
| $T_{bcn\_extra}$ | Time to transmit an IEEE 802.15.4 frame indication in beacon | $0.96\,ms$ |
| $T_{cmd}$ | Time to transmit an IEEE 802.15.4 data request command frame | $2.28\,ms$ |
| $T_{to\_addr}$ | Time to transmit a frame until the destination address | $1.92\,ms$ |
| $T_{data-mac}$ | Time to transmit the data frame excluding the MAC header | - |
| $\lambda$ | Possion packet arrival rate | - |
| $N$ | Number of slaves | - |
| **The End** | | |

# Appendix C

# Frame Structure

## C.1 Frame Structure

The structure of the Data, the ACK, and the SWUF frames is shown in Fig. C.1, Fig. C.2, and Fig. C.3, respectively.

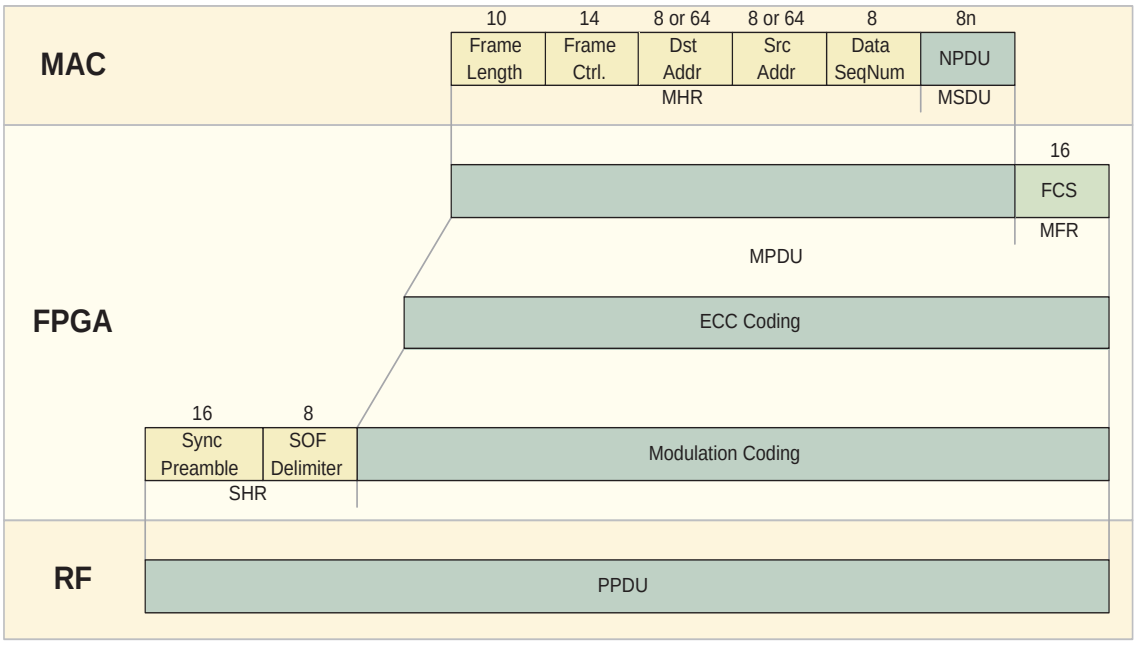


Figure C.1: DATA Frame Structure

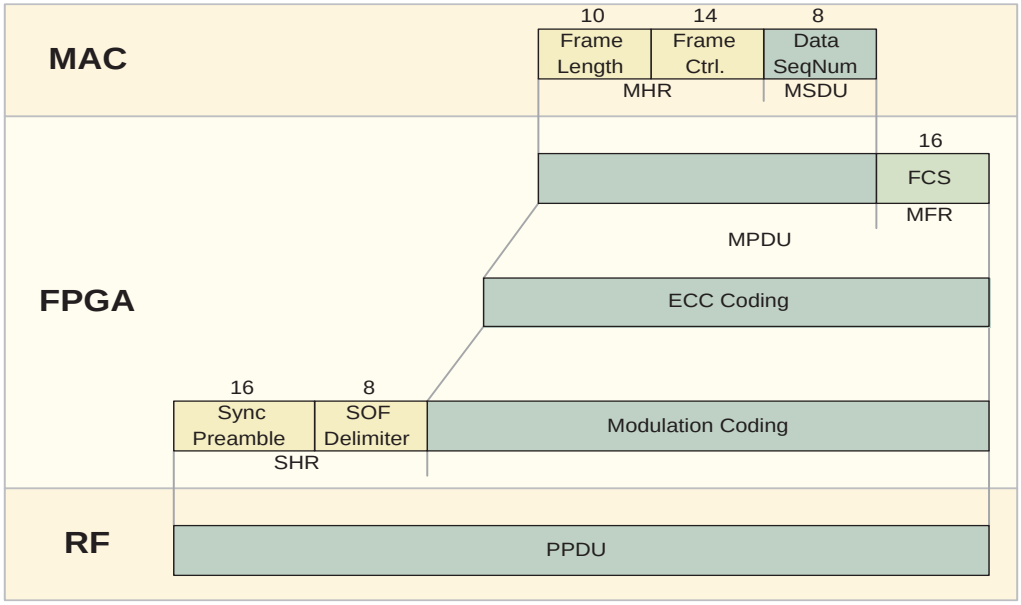


Figure C.2: ACK Frame Structure

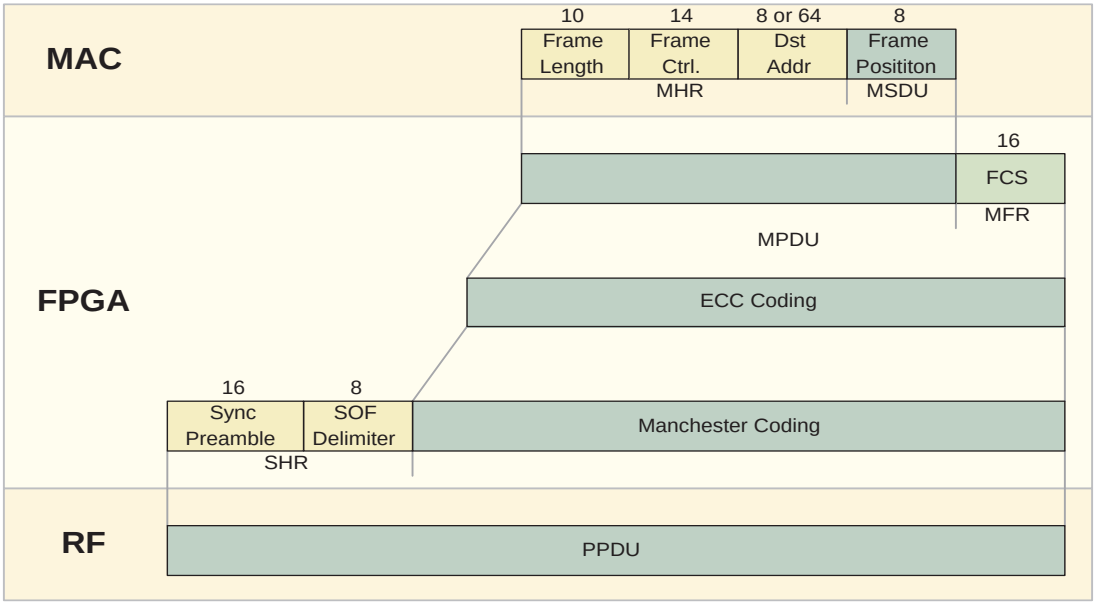


Figure C.3: Short Wake-Up-Frame Structure

As shown in the above frame structures, the data payload from the Network layer, NPDU (Network Protocol Data Unit), or the data payload generated by the MAC layer is fitted into the MSDU (MAC Service Data Unit). Then an MHR (MAC Header) is added at the front of the MSDU, including frame length, frame control, address information and sequence number. After that, the frame is sent to the FPGA and an MFR (MAC Footer), i.e. FCS (Frame Check Sequence), is added forming an MPDU (MAC Protocol Data Unit). Next, the MPDU are ECC coded and modulation coded. Finally, an SHR (Sync Header) is added at the front and the whole frame is sent to the RF transceiver as the PPDU (PHY Protocol Data Unit).

**Note:** The frame should be transmitted and received with the following order:

1. The leftmost field in the frame should be transmitted or received first;
2. All multiple octet fields should be transmitted or received least significant octet first;
3. Each octet should be transmitted and received least significant bit first;

## C.2 Fields Definition

### C.2.1 MAC

- *Frame Length*: 10-bit field indicating the length of the whole MAC frame in byte.
- *Frame Control:*

| | | |
|---|---|---|
| **2-0** | Frame Type | **000**: DATA **001**: ACK **010**: WUF |
| **3** | ACK requested | **0**: ACK not requested **1**: ACK requested |
| **5-4** | protocol type | **00**:IP **01**:ARP **10**:ICMP |
| **6** | Destination Addr. Mode | **0**: Short MAC Address **1**: Long MAC Address |
| **7** | Source Addr. Mode | **0**: Short MAC Address **1**: Long MAC Address |
| **13-8** | Reserved | |

- *Destination Address:* The MAC address of the destination device. Whether it uses 8-bit short MAC address or 64-bit long MAC address is indicated by bit 6 of the frame control field.

- *Source Address:* The MAC address of the source device. Whether it uses 8-bit short MAC address or 64-bit long MAC address is indicated by bit 7 of the frame control field.
- *Sequence Number:* The Data Sequence Number (DSN), an 8-bit field, is used to match an ACK frame to the DATA frame. The value of the sequence number is initialized to a random value, and is incremented by one each time a MAC DATA frame is sent to the FPGA.
- *Frame Position:* This 8-bit field is only used in the SWUF frame, indicating the position of the SWUF in a WUF. The destination calculates the length of the rest of the WUF according to this field. When the MAC layer wants to send a WUF, it sends an SWUF frame to the FPGA, in which the Frame Position field indicates the total number of SWUFs in the WUF.

### C.2.2 FPGA

- *FCS*: FCS is a 16-bit ITU-T Cyclic Redundancy Code (CRC), which is calculated over the MAC header and the MAC payload.
- *SOF Delimiter*: Start Of Frame (SOF) is a delimiter after the sync preamble indicating the start point of the frame.
- *Sync Preamble*: Sync preamble is used for the receiver to get synchronized with the incoming signal. Its content should be an alternating sequence of ones and zeros.

# Bibliography


[1] Mark Weiser. The Computer for the Twenty-First Century. *Scientific American*, pages 94–104, September 1991.

[2] Mark Weiser. Some computer science issues in ubiquitous computing. *Communications of the ACM*, 36(7):75–84, July 1993.

[3] Mark Weiser. Hot Topics: Ubiquitous Computing. *IEEE Computer*, 26(10):71–72, October 1993.

[4] Jan M. Rabaey, M. Josie Ammer, Julio L. da Silva, Danny Patel, and Shad Roundry. PicoRadio Supports Ad Hoc Ultra-Low Power Wireless Networking. *IEEE Computer*, 33(7):42–48, July 2000.

[5] Christian C. Enz, Amre El-Hoiydi, Jean-Dominique Decotignie, and Vincent Peiris. WiseNET: An Ultralow-Power Wireless Sensor Network Solution . *IEEE Computer*, 37(8):62–70, August 2004.

[6] Yvonne Gsottberger, Xiaolei Shi, Guido Stromberg, Thomas F. Sturm, and Werner Weber. Embedding Low-Cost Wireless Sensors into Universal Plug and Play Environments. In *EWSN*, January 2004.

[7] Microsoft Corporation. *Universal Plug and Play Device Architecture 1.0*. Microsoft.

[8] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava. Energy-Aware Wireless Microsensor Networks. *IEEE Signal Processing Magazine*, 19:40–50, March 2002.

[9] C. E. Jones, K. M. Sivalingam, P. Agrawal, and J. C. Chen. A Survey Of Energy Efficient Network Protocols for Wireless Networks. *Wireless Networks*, 7(4):343–358, July 2001.

[10] J. P. Monks, V. Bharghavan, and W. E. Hwu. A Power Controlled Multiple Access Protocol for Wireless Packet Networks. In *INFOCOM*, pages 219–288, April 2001.

[11] S. Narayanaswamy, V. Kawadia, R. S. Sreenivas, and P. R. Kumar. The COMPOW protocol for Power Control in Ad hoc Networks: Theory, Architecture, algorithm, implementation and Experimentation. In *MobiHoc*, June 2003.

[12] W. R. Heinzelman and A. Sinha and A. Wang and A. P. Chandrakasan. Energy-Scalable Algorithms and Protocols for Wireless Microsensor Networks. In *ICASSP*, June 2000.

[13] Wendi Rabiner Heinzelman, Joanna Kulik, and Hari Balakrishnan. Adaptive Protocols for Information Dissemination in Wireless Sensor Networks. In *MOBICOM*, pages 174–185, August 1999.

[14] C. Schurgers, G. Kulkarni, and M. B. Srivastava. Distributed Assignment of Encoded MAC Addresses in Sensor Networks. In *MobiHoc*, October 2001.

[15] CMikael Degermark, Mathias Engan, Bjrn Norgreen, and Stephen Pink. Low-loss TCP/IP Header Compression for Wireless Networks. In *MobiCom*, November 1996.

[16] R. Caceres and L. Iftode. Improving the performance of reliable transport protocols in mobile computing environments. *IEEE Journal on Selected Areas in Communications*, 1995.

[17] Kartik Chandran, Sudarshan Raghunathan, S. Venkatesan, and Ravi Prakash. A Feedback-based Scheme for Improving TCP Performance in Ad Hoc Wireless Networks. In *ICDCS*, page 472. IEEE Computer Society, 1998.

[18] J. Liu and S. Singh. ATCP: TCP for Mobile Ad Hoc Networks. *IEEE Journal on Selected Areas in Communications*, 19(7):1300–1315, July 2001.

[19] C. Parsa and J.J. Garcia-Luna-Aceves. TULIP: A Link-Level Protocol for Improving TCP over Wireless Links. In *WCNC*, 1999.

[20] R. A. F. Mini, B. Nath, and A. A. F. Loureiro. A Probabilistic Approach to Predict the Energy Consumption in Wireless Sensor Networks. In *4th Workshop de Comunicao sem Fio e Computao Mvel*, October 2002.

[21] A. Sinha and A. P. Chandrakasan. Operating System and Algorithmic Techniques for Energy Scalable Wireless Sensor Networks. In *MDM*, pages 199–209. Springer-Verlag, 2001.

[22] R. Kravets and P. Krishnan. Application-driven power management for mobile communication. *Wireless Networks*, 6:263–277, 2000.

[23] Eui-Young Chung, L. Benini, and G. De Micheli. Dynamic Power Management for Non-Stationary Service Requests. In *Design, Automation and Test in Europe Conference*, pages 77–81, March 1999.

[24] C. S. Raghavendra and S. Singh. PAMAS: Power efficient MAC protocol for multihop radio networks. In *ACM Computer Communication Review*, 1998.

[25] W. Ye, J. Heidemann, and D. Estrin. An Energy-Efficient MAC Protocol for Wireless Sensor Networks. In *INFOCOM*, pages 1567–1576. IEEE, June 2002.

[26] Tijs van Dam and Koen Langendoen. An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks. In *ACM SenSys*, November 2003.

[27] Infineon Wireless Solutions. *ASK/FSK 434MHz Wireless Transceiver TDA 5255 E1*. Infineon Technologies AG, http://www.infineon.com/, version 1.1 edition, December 2002.

[28] ACTEL, http://www.actel.com/. *PROASIC Flash Family FPGAs*, August 2003.

[29] Infineon Technologies AG, http://www.infineon.com/. *C161PI User Manual*, August 1999.

[30] Microchip Technology Inc., http://www.microchip.com/. *PIC16F87/88 Data Sheet*, July 2003.

[31] BSI, http://www.bsi.com.tw/. *BS62LV1024 Data Sheet*, February 2003.

[32] AMD, http://www.amd.com/de−de/FlashMemory/ProductInformation/. *AM29LV160D Data Sheet*, November 2000.

[33] IEEE Computer Society. *IEEE Standard for Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low Rate Wireless Personal Area Networks (LR-WPANs)*. The Institute of Electrical and Electronics Engineers, Inc., 3 Park Avenue, New York, NY 10016-5997, USA, October 2003.

[34] Amre El-Hoiydi. Aloha with Preamble Sampling for Sporadic Traffic in Ad Hoc Wireless Sensor Networks. In *ICC*, April 2002.

[35] Amre El-Hoiydi and J.-D.Decotignie. WiseMAC: An Ultra Low Power MAC Protocol for the Downlink of Infrastructure Wireless Sensor Networks. In *ISCC*, June 2004.

[36] Amre El-Hoiydi and J.-D.Decotignie. WiseMAC: An Ultra Low Power MAC Protocol for Multi-Hop Wireless Sensor Networks. In *ALGOSENSORS*, July 2004.

[37] Amre El-Hoiydi, J.-D.Decotignie, C. Enz, and E.Le Roux. Poster Abstract: WiseMAC, an Ultra Low Power MAC Protocol for the WiseNET Wireless Sensor Network. In *SenSys*, November 2003.

[38] Amre El-Hoiydi, Jean-Dominique Decotignie, and Jean Hernandez. Low Power MAC Protocols for Infrastructure Wireless Sensor Networks. In *European Wireless*, February 2004.

[39] *Guidelines for 64-BIT Global Identifier (EUI-64) Registration Authority*. http://standards.ieee.org/regauth/oui/tutorials/EUI64.html.

[40] The VINT Project, http://www.isi.edu/nsnam/ns/ns−documentation.html. *The ns Manual*, Dec 2003.

[41] UCLA PCL, http://pcl.cs.ucla.edu/projects/glomosim/GloMoSimManual.html. *GloMoSim Manual*, version 1.2 edition, April 2003.

[42] Theodore S. Rappaport and Scott Y. Seidel. Statistical Channel Impulse Response Model for Factory and Open Plan Building Radio Communication System Design. *IEEE Transactions on Communications*, 39(5):794–807, May 1991.

[43] Theodore S. Rappaport. *Wireless Communications,Principles and Practice*. Prentice Hall, Braunschweig/Wiesbaden, 2nd edition, 2002.

[44] Mineo Takai, Rajive Bagrodia, Addison Lee, and Mario Gerla. Impact of channel models on simulation of large scale wireless networks. In *MSWiM*, pages 7–14. ACM Press, 1999.

[45] J. Eric Nuckols. *Implementation of Geometrically Based Single-Bounce Models for Simulation of Angle-of-Arrival of Multipath Delay Components in the Wireless Channel Simulation Tools, SMRCIM and SIRCIM*. PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 1999.

[46] Homayoun Hashemi. The Indoor Radio Propagation Channel. In *Proceedings of the IEEE*, volume 81, pages 943–968, July 1993.

[47] James A. Roberts and Juan M. Bargallo. DPSK Performance for Indoor Wireless Rician Fading Channels. *IEEE Transactions on Communications*, 42(2/3/4), February/March/April 1994.

[48] Chong Cai. IEEE 802.15.4 Low-Rate Wireless Personal Area Network MAC Protocol Simulation. Master's thesis, Technische Universität München, September 2004.

[49] Mischa Schwartz. *Telecommunication Networks: Protocols, Modelling and Anaylsis*. Addison-Wesley Publishing Company, Department of Electrical Engineering and Center for Telecommunications Reesearch, Columbia University, 1 edition, November 1998.

[50] Resa Corporation. *Statistical Distributions*. http://www.xycoon.com/contdistroverview.htm.