Delft University of Technology
Master's Thesis in Electrical Engineering

Designing Efficient Wireless Power Transfer Networks

Michał Goliński

Designing Efficient Wireless Power Transfer Networks

Master's Thesis in Electrical Engineering

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

Michał Goliński
M.Golinski@student.tudelft.nl

29th May 2015

Author

Michał Goliński (M.Golinski@student.tudelft.nl)

Title

Designing Efficient Wireless Power Transfer Networks

MSc presentation

29th May 2015

Graduation Committee

prof. dr. Koen Langendoen	Delft University of Technology
dr. Przemysław Pawełczak	Delft University of Technology
dr. Martijn Warnier	Delft University of Technology

Abstract

The techniques of wireless power transfer has been gaining increasing popularity in the recent years due to the widespread use of mobile electronic devices such as laptops, mobile phones, wearable electronics and wireless sensors. Traditionally, these devices are powered using a battery that needs to be recharged. Wireless power transfer has the potential of enabling new applications of those and many different devices, where it is not possible to use the battery (e.g. due to its size or weight), or where recharging the battery using a wired charger is not practical (e.g. wireless sensor networks).

In this thesis we investigate Wireless Power Transfer Networks (WPTNs). These are the networks of wireless power transmitters and receivers that are connected together in a network with the goal of increasing network efficiency through maximizing charging of receivers while minimizing the consumption of energy in the network. In this thesis, the architecture of devices in such WPTNs has been proposed. Subsequently, a physical prototype of such networks has been designed and build. Finally, two fundamentally different approaches to achieving high efficiency has been proposed and analyzed in this thesis.

After designing and creating the proof-of-concept WPTN system, its performance has been analyzed, modeled mathematically and verified experimentally. A number of performance measures of WPTNs has been proposed and two approaches to solving the optimization goal of WPTNs have been compared using these measures. Finally, certain conclusions have been drawn on the trade-offs between the first — simpler approach, and the second — more complex approach.

Acknowledgements

This thesis was conducted as a part of Wireless Power Transfer Networks project at the Embedded Software group of TU Delft composed of the following team members: Michał Goliński, Qingzhi Liu, Przemysław Pawełczak and Martijn Warnier. The project was supported by the SHINE project of Delft Institute for Research on ICT and by the Dutch Technology Foundations STW under contract 12491. Parts of this work are also available as arXiv report 1504.00639 as well as a TU Delft Technical Report number ES-2015-01 Green Wireless Power Transfer Networks and submitted for possible publication.

Preface

Telecommunication is one of the very applied branches of technology. During my studies in that field I was always interested in finding a bridge between innovation and application. This pursuit led me to my post-graduate studies at TU Delft and finally to choosing a research topic within the Embedded Software group. The purpose of my research topic was to attempt the seemingly impossible — to design and implement an efficient Wireless Power Transfer Network (WPTN). It was a task that, to our knowledge, has never been done before. On the one hand, it sounded simple, but on the other, just a short introduction to the topic unraveled many hidden complexities that needed to be tackled. Many of those challenges were discovered, some of them solved and some of them just barely touched. My main motivation throughout the project was this journey of discovering and defining new concepts, bringing them to life and demonstrating that they really work. It was a tough journey with many ups and downs and the result is this thesis. I hope my work contributes to the development of the field of WPTNs.

This undertaking would not have been possible with strong support of so many people. First of all, I would like to express my deepest thanks to my thesis supervisor and a great mentor Dr. Przemysław Pawełczak for his continuous support, faith and guidance. During many fruitful discussions I had the chance to learn how to conduct research, think critically and clearly communicate my ideas. I would also like to thank Dr. Martijn Warnier and Mr. Qingzhi Liu for many discussions during the time I was working on my thesis. Their immense experience was of great help for shaping my thoughts and my research results. Furthermore, I would like to express my thanks to prof. dr. Koen Langendoen for his careful review of my work and useful feedback. Finally, I would like to thank my family and friends for their never-ending support.

Michał Goliński

Delft, The Netherlands
29th May 2015

Contents

Preface **vii**

1 Introduction **3**
1.1 Project motivation and goals 4
1.2 Contributions 5
1.3 Thesis structure 5

2 Wireless Power Transfer networks: State of the Art **7**
2.1 Wireless Power Transfer (WPT) Techniques 7
2.1.1 Electromagnetic (EM) radiation 7
2.1.2 Inductive coupling 8
2.1.3 Magnetic resonant coupling 9
2.1.4 Acoustic waves 10
2.2 Motivation for WPT Networks 11
2.3 Existing WPT Networks Classification 12
2.3.1 WPTN topology 12
2.3.2 Mobility 12
2.3.3 WPT Physical Layer technique 13

3 Network model for Wireless Power Transfer **15**
3.1 Proposed Model 15
3.1.1 Types of nodes 15
3.1.2 Assumptions about the environment 16
3.2 Hardware platform 17
3.2.1 WPT hardware 17
3.2.2 Communication layer 19
3.2.3 Controller 19
3.2.4 WPT transmitter and receiver 20
3.3 Rectifier efficiency 21
3.4 Proposed protocols 22
3.4.1 Beaconing-based protocol 23
3.4.2 Probing-based protocol 29
3.5 Mathematical analysis 38

3.5.1 Description of the scenario 39
3.5.2 Time analysis of protocol phases 39
3.5.3 Time to start charging 41

4 Implementation 47
4.1 System design 47
4.2 Implementation 48
4.2.1 Software architecture 48
4.2.2 Implementation of control protocol 49
4.2.3 Implementation of measurement protocol 51
4.3 Accuracy of time synchronization 51

5 Simulation and measurements 55
5.1 Measurement scenario 55
5.2 WPTN Performance Metrics 56
5.2.1 Choice of performance measures 56
5.2.2 Accuracy 57
5.3 Results 62
5.3.1 ERx Time to Charge 62
5.3.2 Reference measurement 63
5.3.3 Line-of-sight scenario 65
5.3.4 Non-line of sight scenario 69
5.3.5 Results summary 72

6 Conclusions and Future Work 75
6.1 Conclusions 75
6.2 Future Work 77

A Source code of WPTN implementations 85

Glossary

CDF Cumulative Distribution Function.

DC Direct current.

DSSS Direct Sequence Spread Spectrum.

EIRP Effective Isotropic Radiated Power.

FCFS First come first served.

FSM Finite State Machine.

IEEE Institute of Electrical and Electronics Engineers.

MSC Message Sequence Chart.

QoS Quality of Service.

RF Radio frequency.

RFID Radio-frequency identification.

RSSI Received signal strength indication.

SHARP Stationary High Altitude Relay Platform.

UAV Unmanned Aerial Vehicle.

WPT Wireless Power Transfer.

WPTN Wireless Power Transfer Network.

Chapter 1

Introduction

Wireless power transfer is the idea of transmitting power without the use of any wires or cables, as it is traditionally done. The idea of wireless power transfer inspired researchers ever since James Clerk Maxwell derived a set of equations [1], that are the foundations of electrodynamics, which is the basis of today's telecommunication and wireless power transfer. As the use of electricity became more widespread, the idea of wireless power transfer gained more interest. At the beginning of the twentieth century, approximately thirty years after Thomas Edison patented a system for electricity distribution [2], the electrical engineer Nikola Tesla patented a technique for a large-scale wireless power distribution [3]. Even though there is no solid evidence that Tesla could transmit power on a large-scale distance [4], the idea inspired many generations of researchers worldwide.

During World War II there was not much direct interest in the wireless power transfer. However, major developments were made in the field of microwave transmission [5]. The new era of wireless power transmission started after developments made by William C. Brown, who was a pioneer of microwave power transmission. In 1964 he invented a rectenna, a rectifying antenna, that can be used to covert microwave energy into direct current electricity. He patented his invention in 1969 [6]. A typical rectenna consists of an antenna and a rectifier. Usually, low voltage drop Shottky diodes are used as the rectifying element to make a conversion from microwave to direct current energy. Brown demonstrated his idea using a small helicopter that was equipped with a rectenna and powered wirelessly using microwaves [7].

Further developments regarding wireless power transfer were connected with Solar-Power Satellite [5] — a satellite that would collect solar power in space and send it to earth using microwaves. The concept has been in development since the early 1970s. Another line of research pertained to powering unmanned aerial vehicles (UAV) wirelessly. An example of such research is the Stationary High Altitude Relay Platform (SHARP) developed by the Communications Research Center in Canada [8]. The aircraft could fly at an altitude of 21 km powered by microwaves beamed at the aircraft from the earth.

In addition to the above-mentioned far-field techniques, there was also considerable effort done to advance near-field wireless power transmission techniques, which are based on either inductive or capacitive coupling (magnetic or electric fields, respectively). The first passive radio frequency identification (RFID) devices were invented in the 1980s [9] [10] and applied widely in industry in the 1990s. The development of a variety of mobile devices such as smartphones, tablets and laptops led to the creation of the Wireless Power Consortium in 2008. The aim of this consortium is to establish international standards of wireless charging, which has been crystallized in the form of the Qi standard [11].

The future of WPT seems bright — in the recent years there has been a growing interest in the applications of WPT techniques in the area of sensor networks [12] and biology research (e.g. monitoring insects [13] and animals [14]). There has been also an increasing interest in the field from industry. The result is a number of companies in the WPT field, such as WiTricity [15], uBeam [16], Ossia [17], Artemis [18], Energous [19], and Proxi [20].

1.1 Project motivation and goals

Most of the above-mentioned research takes into consideration a one-to-one link between energy transmitter and energy receiver. Currently, there is limited research being conducted in the field of networks of power receivers and transmitters. Bridging this gap seems to be a natural next step to making wireless power transfer technology ubiquitous. However, there is one major challenge that needs to be addressed while considering far-field wireless power transfer techniques, namely propagation loss, which accounts for the highest power loss of such systems, resulting in low overall system efficiency. This thesis is an attempt to answer the following questions:

- How to build energy-efficient Wireless Power Transfer Networks (WPTNs) and what kind of protocols could control them?

- Can we create a proof-of-concept system using off-the-shelve hardware and custom made software?

- How can we measure efficiency of such networks?

- What kind of techniques are available to address the problem of increasing overall efficiency?

1.2 Contributions

The main contributions of this thesis directly follow the project goals defined initially, and these are:

- Propose a new WPTN control protocol that improves three performance metrics: harvested energy, efficiency and accuracy;
- Creating a novel hardware based implementation of WPTN, based on off-the-shelve hardware;
- Conducting extensive measurements of multiple versions of WPTN protocols to compare them against the proposed efficiency measures.

1.3 Thesis structure

This thesis is organized as follows. Chapter 2 provides the necessary background for further understanding of the thesis. It provides background on different wireless power transfer techniques, motivates the need for wireless power transfer networks research and presents the classification of such networks existing in literature. Chapter 3 further describes the network model that will be addressed in this thesis. It presents our WPTN protocol in two flavors (probing and beaconing-based) and provides mathematical analysis of the characteristic features of those protocols. Chapter 4 provides details of the hardware implementation that was made to evaluate the two different approaches to WPTN protocols. It describes the hardware platform, WPTN protocol (which is also called *control protocol*) and all the additional features implemented to enable measurements of efficiency measures of those implementations (all those features are called *measurement protocol*). Chapter 5 describes simulations and experiments that have been done in order to verify and compare the protocols proposed in the thesis. Chapter 6 concludes the work.

Chapter 2

Wireless Power Transfer networks: State of the Art

This chapter provides background of Wireless Power Transfer and motivates the project. In Section 2.1 existing Wireless Power Transfer (WPT) techniques are described. Section 2.2 motivates the need for WPT Networks (WPTNs) and Section 2.3 presents a classification of existing WPTNs.

2.1 Wireless Power Transfer (WPT) Techniques

There are a lot of different techniques of Wireless Power Transfer. A very popular technique is the rectification of electromagnetic (EM) radiation. It can be used with omnidirectional or directional antennas and using different transmission techniques (microwaves, laser etc.). The technique that gained popularity in consumer electronics is inductive coupling, present for instance in charging of electric toothbrushes, cell phones and laptops. The technique that is gaining interest in the recent years is magnetic resonant coupling, mainly due to recent advances by Kurs *et al.* [21] in 2007. Apart from that, there are WPT techniques that use acoustic waves to transmit energy. In the following subsections every technique will be briefly explained.

2.1.1 Electromagnetic (EM) radiation

EM radiation makes energy transfer possible by the transmission of the electromagnetic waves from the transmitting antenna connected to the power source to the receiving antenna. In the context of Wireless Power Transfer the EM waves most frequently used are microwaves (between 300 MHz and 300 GHz) and visible light (frequencies between approximately 430 THz and 790 THz). EM radiation is widely used for information transfer, but transferring power suffers from efficiency problems due to a very high attenuation of EM waves in space. In mathematical models in literature loss is usually proportional to the power (of two or more) of the distance between a receiver and a transmitter antenna.

Systems based on light are mainly used directionally (in opposition to microwaves that are also used in omnidirectional antennas). This technique has been successfully used to power UAVs [8]. However, it is not the best candidate for WPTN applications due to the fact that it requires a line-of-sight scenario.

Advantages

- Small receiver size;
- Efficient power transfer over long distances when directionality is used (microwave/laser).

Disadvantages

- High attenuation of EM waves in atmosphere;
- In terms of directional devices - requires complex tracking mechanisms and line-of-sight (LOS).

2.1.2 Inductive coupling

Inductive coupling is a technique where energy is being transferred between two coils placed in close proximity. Coils are configured in such a way that alternating current flowing in one of the coils induces a current flowing in the second one. The two coils can be seen as an electrical transformer. This technique was the basis for the world's first international wireless charging standard — Qi — created by Wireless Power Consortium [11]. The first version of the Qi standard (for low-power inductive charging — up to 5 W) was published in August 2009. The standard specifies a technique for power transfer over small distance (up to 4 cm). The standard was extended to medium-power (up to 120 W) in 2011. However, for this technique to work the transmitter and receiver coils need to be located very closely to each other and need to be specifically aligned. That is the reason this technique is not the most suitable for Wireless Power Transfer Networks applications (which assume distances longer than a few centimeters).

Inductive coupling has been successfully deployed in consumer electronics (e.g. toothbrushes, mobile phones) and electric vehicles. Due to the fact that the power transfer using this technique is efficient only on extremely short distance, this method did not find many other applications.

Advantages

- Simplicity;

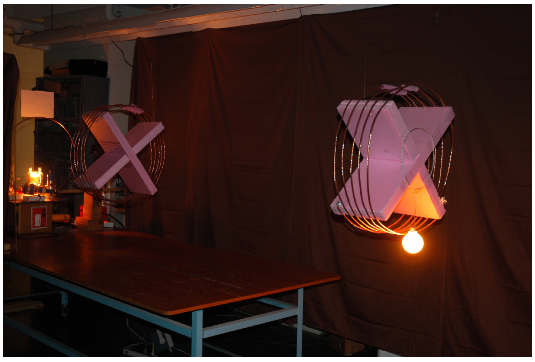


Figure 2.1: 60 W light bulb being lit on a distance of 2 m using magnetic resonant coupling. Image source: online supplementary material for [21].

- Very short range (less than a few centimeters) with a very high efficiency.

Disadvantages

- Requires accurate alignment of the transmit and receive coils;
- Efficient charging only on short distance (less than the coil diameter).

2.1.3 Magnetic resonant coupling

Magnetic resonant coupling is a technique developed by Kurs *et al.* [21] in 2007. It is a technique that uses strongly coupled, self resonant coils. Using this technique very high efficiency could be achieved over distances up to 8 times the radius of the coil. The authors of [21] were able to demonstrate an efficient (approximately 40% efficiency) power transfer to a light bulb over a distance of more than 2 meters (Figure 2.1).

The main idea behind this technique is to use two coils with a high quality factor (Q). Coils (inductors) are elements that store energy in the magnetic field. A high Q factor of the coil means that the coil is closer to the ideal coil (coil with only inductance, no capacitance nor resistance), thus it does not loose much energy due to resistance of the wire or capacitance. In order to make those two coils resonant, capacitors are added to the system, creating resonant RC circuits.

The developments presented in [21] were the basis of creating a WPT company — WiTricity [15] — that tries to commercialize this technology.

Advantages

- High efficiency over several times the coil diameter (e.g. 2 meters, as demonstrated by Kurs *et al.* [21]);

- Insensitive to weather conditions (unlike EM radiation);
- Not requiring LOS.

Disadvantages

- Requires alignment of the transmit and receive coils (so that the coupling factor between the coils is maximized);
- Efficient charging only over a distance of a few meters.

2.1.4 Acoustic waves

All of the techniques described in the previous subsections use electromagnetic fields or waves to transfer energy. It is important to notice that the principle of energy transfer applies to not only electromagnetic waves, but to any kind of waves. One of the alternative approaches to WPT is wireless energy transfer through acoustic waves, which are pressure waves.

One of the main advantages of using acoustic waves instead of electromagnetic waves is the lower propagation speed of those waves compared to electromagnetic waves. Let c_{em} be the propagation speed of electromagnetic waves and c_{ac} be the propagation speed of acoustic waves in the same medium. As a result, the frequency to be used for a given wavelength needs to be of an order $\frac{c_{em}}{c_{ac}}$ smaller [22]. Because of the fact that frequencies used are much smaller than for EM propagation, there are fewer losses in the power electronics that drive the circuit, as well as the design of this electronics is simpler [23]. On the other hand, if the used frequency is fixed, choice of acoustic power transfer technology will result in a smaller transmitter and receiver size for the same directionality of the wave [24]. Another advantage of this technology is the fact that the efficiency of a system is higher than EM-based systems for longer distances [24]. Figure 2.2 shows the results of a simulation performed in [24]. From the figure we can conclude that although inductive power transfer is more efficient for smaller distances, acoustic power transfer is much more efficient for higher distances (with a factor of approximately 100 for a distance of 6 cm).

The challenges of this technique are reflections and resonances of acoustic waves. It puts restrictions on how the transmitter and the receiver can be placed with regard to each other for an optimal power transfer. There are also problems regarding matching impedance of different transmission mediums for efficient transfer (air, piezoelectric material, tissue, metal) due to lack of low-loss materials with a specific acoustic impedance.

Advantages

- High efficiency over distances that are large in comparison to the dimensions of the receiver and transmitter devices;

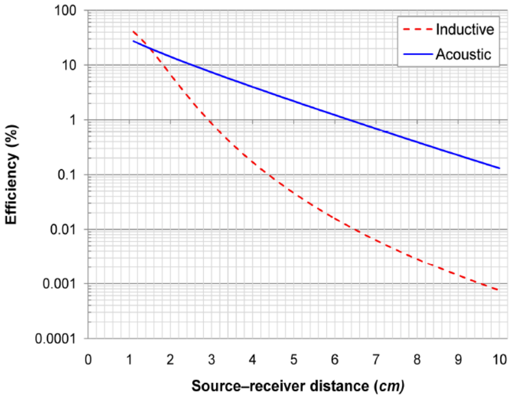


Figure 2.2: Efficiency as a function of source-receiver distance (for a 5 mm receiver). Source: [24].

- Theoretically, an efficiency higher than that of inductive coupling by an order of magnitude;
- As a consequence of using lower frequencies, the system experiences low switching loses;
- Usage of low frequencies simplifies electronic design;
- Can be used when EM-based techniques cannot (e.g. WPT through metal wall).

Disadvantages

- Generating high acoustic power with high efficiency is challenging;
- Difficulty of matching impedance in different transmission media;
- Currently documented in literature measurements of power transfer though air show that currently the transferred power is too low for any practical applications [22].

2.2 Motivation for WPT Networks

As mentioned in the previous chapter, there has been a rapid increase in the interest in WPT, triggered by some recent developments (e.g. [15,21,25,26]). A lot of research done in the field of WPT focuses on improving the quality of one-to-one or one-to-many transmission. The focus is put on the physical layer and improving link efficiency through sophisticated techniques.

The need for WPT seems obvious if we look into platforms that are based on ambient energy harvesting. Up till this moment no such systems are deployed with high success rate. This problem can be alleviated by using

WPT techniques where wireless power is required. WPT has one major drawback connected with its efficiency. Most of the techniques are not very efficient, especially over larger distances. The range of efficient wireless power transfer is usually within a couple of centimeters up to a couple of meters from the transmitter. This introduces the need to keep track of energy receivers and make energy transmitters active only when there are receivers that could receive energy transmitted by a given transmitter.

In the general scenario we can imagine energy transmitters that are static and energy receivers that move (or appear/disappear from the network). Even though such systems resemble a cellular network in terms of topology, such systems have never been considered in the area of WPT. The goal of this thesis is to propose and evaluate protocols that could give intelligence to such a Wireless Power Transfer Network (WPTN).

2.3 Existing WPT Networks Classification

Existing WPT Networks can be classified in a number of ways. We can consider WPTN physical layer technique used or WPTN topologies.

2.3.1 WPTN topology

A WPTN topology is based on the number of n energy receivers (called ERx devices in this thesis) and m energy transmitters (called ETx devices). As a consequence, we can analyze four types of WPTNs:

- $m = 1$, $n > 1$ e.g. [27, 28];
- $m > 1$, $n > 1$ e.g. [29, 30];
- $m = 1$, $n = 1$ e.g. [31, 32];
- $m > 1$, $n = 1$ (not considered in a literature according to the best of our knowledge).

Moreover, from the topology point of view, WPTN networks can be seen as planned, e.g. [27–29, 31] or unplanned, e.g. [33].

2.3.2 Mobility

WPTNs can be also categorized regarding the mobility of ERx and ETx devices. In the literature we can find the following classification:

- static ETx/static ERx [30, 31];
- mobile ETx/static ERx [28, 34];
- static ETx/mobile ERx [12, 35, 36];

- mobile ETx/mobile ERx (not considered in a literature according to the best of our knowledge).

2.3.3 WPT Physical Layer technique

In Section 2.1 we have presented different WPT techniques. WPTNs can be classified according to which technique of wireless power transfer they use: electromagnetic radiation, inductive coupling, magnetic resonant coupling or other techniques.

Energy/Communication transmission Separation

Separation of energy provision and communication/control in WPTN can be categorized into

- joint energy and information transmission (through power splitting) [29, 37, 38];
- time division approach [27, 31];
- frequency division [34, 39] (often in relation to inductive-based WPT).

In this thesis we introduce one more way of categorizing WPTNs, that is based on type of feedback loop in the system. Regardless of the type of energy and communication transmission separation, we could consider networks, that operate based on no feedback, feedback from communication (beaconing-based) and feedback from received power measurements (probing-based).

Chapter 3

Network model for Wireless Power Transfer

This chapter describes the network model, for which our WPTN protocol was designed. In Section 3.1 details of the proposed model and its assumptions are described. Since the model used is a direct consequence of the hardware platform chosen for the implementation, this platform is explained in Section 3.2. Section 3.3 describes issues related to the non-linearity of our WPT hardware. Afterwards, in Section 3.4 two protocols for solving WPTN problems are being proposed and finally in Section 3.5 a mathematical analysis of both protocols is provided.

3.1 Proposed Model

Based on the description of different techniques, presented in Chapter 2, we decided to choose electromagnetic (microwave) radiation technology in our protocol. The hardware used in the implementation is described in this chapter. The technical details of the implementation of our setup will be explained in detail in Chapter 4.

Despite that choice, we designed the protocol not to be dependent on a particular WPT technology. The WPT layer is being abstracted in the design and the focus is put in optimizing charging on the network layer.

3.1.1 Types of nodes

In the protocol we consider two types of nodes - energy transmitters and energy receivers, called ETx and ERx, respectively.

ETx nodes are assumed to be static and equipped with a charging (in our experimental setup — directional — 60°) antenna that is able to cover a certain area. ETx devices can be in two modes: charging or no-charging. When ETx is in charging mode, it transmits energy through its antenna; if it is in no-charging mode, it does not transmit energy. The location of ETx nodes is pre-determined in the phase of planning of a WPTN. The goal of the process is similar to the one used in planning of mobile communication networks, where locations for base stations are chosen in a way that the mobile network signal covers the desired area. In the same way, ETx placement should be such, that it enables a certain level of Quality of Service (QoS) to be met. Apart from a wireless charging module, ETx nodes are equipped with a controller module and a communication module. The controller module controls the behavior of the wireless charging module (state — charging or no-charging). It makes its decision whether to charge or not based on communication with other ETx and ERx devices nearby.

ERx nodes are assumed to be mobile. They can randomly appear and disappear from the network, move around or stay static. They are equipped with an antenna (in our experimental setup — omnidirectional) and a rectifier that is able to convert the RF energy received through the antenna to constant DC energy that can be supplied to any kind of functional device. Apart from the energy harvesting layer, similarly to ETx devices, ERx devices also have controller module and a communication module. The controller module measures the harvested power and controls the WPTN protocol specific behavior of the ERx device, such as advertising its presence. Details of ETx and ERx behavior, and interaction will be explained in Section 3.4.

3.1.2 Assumptions about the environment

We assume our WPTN environment to be a typical office space. Our network would be deployed, so that it provides sufficient coverage and makes wireless power transfer possible in the areas where it is required. Power receivers would be typically devices that are held by people in the office space (e.g. mobile phones, wearable sensors). That is the reason we assume mobility of power receiver nodes. The placement and coverage WPTN problems are out of the scope of this thesis.

The goal of the protocol is depicted in Figure 3.1. The goal could be stated as: *minimize energy consumption in the network while maximizing charging.*

In the most naive deployment, a WPTN would be deployed and connected to a power source. ETx devices would be in charging state no matter if there are any nearby devices that could receive power that is being beamed. This approach ensures that whenever an ERx device appears in the network, it will be charged to the maximum possibility of WPTN. However, when no ERx device is nearby, the energy transmitted by an ETx is wasted. It can be stated that energy consumed by the network is being wasted, because it is not being efficiently utilized. The requirement for minimalization of unnecessary energy consumption appears. Another naive approach, this

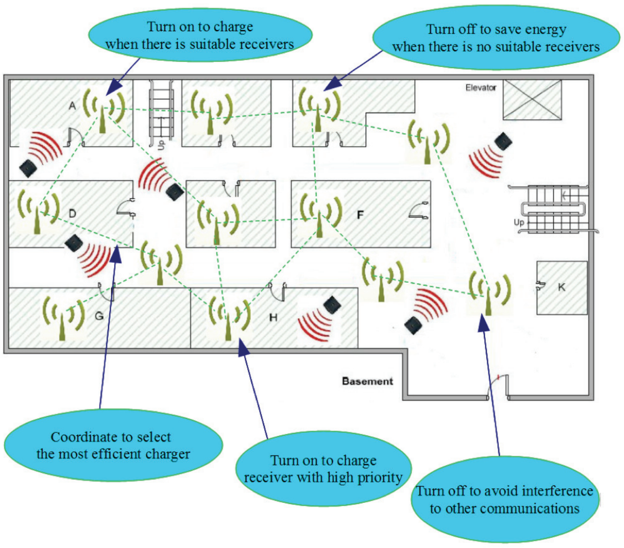

Turn on to charge when there is suitable receivers
Turn off to save energy when there is no suitable receivers
Elevator
A
B
D
F
G
H
K
Up
Basement
Coordinate to select the most efficient charger
Turn on to charge receiver with high priority
Turn off to avoid interference to other communications

Figure 3.1: WPTN scenario. Figure courtesy of Qingzhi Liu.

time fulfilling the requirement of minimized unnecessary energy consumption could assume no energy at all is consumed in the network — all ETx devices are in no-charging state, and thus do not consume energy from the power sources. In such a case, however, when an ERx appears in the network, it will not be charged, because all ETx devices are off. Any practical solution needs to address those two opposing goals — minimizing the energy consumption in the network while maximizing the charging of the ERx devices. Two protocols that address this problems are explained in detail in Section 3.4.

3.2 Hardware platform

Our hardware platform consists of three elements — WPT hardware, communication layer and a controller — that will be explained in the subsections below.

3.2.1 WPT hardware

We have chosen EM radiation as the technology to evaluate our WPTN design. The reason for that is that it is the simplest technology to model and the most accessible one. It relies on basic EM wave propagation principles.

Our WPTN is composed of four Powercast TX91501-3W transmitters with integrated 8dBi, 60°-directional antennas [40, /products/powercaster-transmitters]. The transmitter broadcasts radio waves in the 915 MHz band with a power of 3W Effective Isotropic Radiated Power (EIRP). The power transmitted at the fixed frequency is modulated using Direct Sequence Spread

	P1110	**P2110**
Max. distance	short ($\sim 3\,\mathrm{m}$)	long ($\sim 10\,\mathrm{m}$)
RF range	-5 dBm to 20 dBm	-11.5 dBm to 15 dBm
Output voltage	1.8 V to 4.2 V	1.8 V to 5.25 V
Output voltage type	continuous	intermittent
Max. conversion efficiency	$\sim 70\,\%$	$\sim 55\,\%$
Interrupt available	No	Yes

Table 3.1: Comparison of Powercast P1110 and P2110 features

Spectrum (DSSS). These power transmitters are powered by a 5V DC power suppl.

Powercast transmitters transmit power to the receivers. The Powercast receiver is a rectifier, that converts AC energy received through the connected antenna to DC energy. There are two types of Powercast receivers that could have been chosen: P1110 and P2110. Even though both receivers can efficiently harvest energy, the type of applications they were designed for differ. The main difference between those devices is the type of provided DC power: in case of P1110 it is continuous while in case of P2110 it is pulsed. Thus, the P1110 can be directly used for battery charging, while the P2110 is better suited for battery-free devices such as sensors and low-power electronics. The P2110 needs additional circuitry in order to charge batteries. Another major difference is the difference in relation of input (AC) power to output (DC) power provided, see Figure 3.2. This relation is called the efficiency of RF–to–DC conversion. As we can see from Figure 3.2, a P1110 device has higher efficiency than a P2110 device. However, it operates in the range of higher input power. Thus it is a device for low distance, high power applications. The P2110 power harvester in turn, has lower efficiency, but it operates in a lower range of input powers, which means longer distances. Last but not least, the P2110 device has an interrupt available on one of its outputs. The interrupt becomes available when power is received at the input antenna. This allows to wake up additional circuitry such as sensors or microcontrollers. The P1110 does not have interrupt available, which affects the way system using this chip will be designed. The summary of differences between P1110 and P2110 can be found in Table 3.1.

As mentioned in Subsection 3.1.2, our scenario considers WPTN in office spaces. In many office spaces, the maximum distance within one room is in the range of a few meters. The approximate maximum range of a P1110 is 3 m, so by placing two such transmitters in one office it should be possible to cover the office space.

Our powerharvester is provided with the P1110-EVB receiver evaluation board with co-supplied 1 dBi unidirectional antenna [40, /products/development-kits], see Fig. 3.3.

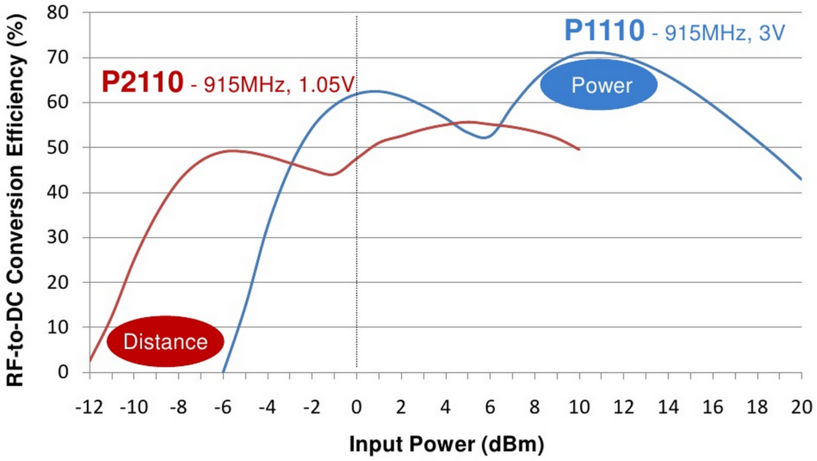


Figure 3.2: Powercast P1110 and P2100 efficiency comparison. Source: [40, /products/powerharvester-receivers].

3.2.2 Communication layer

For the communication layer the IEEE 802.11.4 standard was used. IEEE 802.11.4 is a standard that specifies the physical and media access layers for low-rate wireless networks. It is the basis for the ZigBee suite of protocols, which are typically used for personal area networks, that can be characterized by low data rates and requirement of long battery life. This last requirement was key to choosing this technology over the IEEE 802.11 standard, that is the basis for wireless local area networks (WLAN).

The device we use for communication is the XBee 802.15.4 module, see [41] and Figure 3.4(a). The reason for choosing this particular module is the fact it is designed for low-power and low-cost applications, as well as the fact that it integrates seamlessly with the Arduino Uno controller layer (see Subsection 3.2.3) through the Arduino Wireless SD Shield.

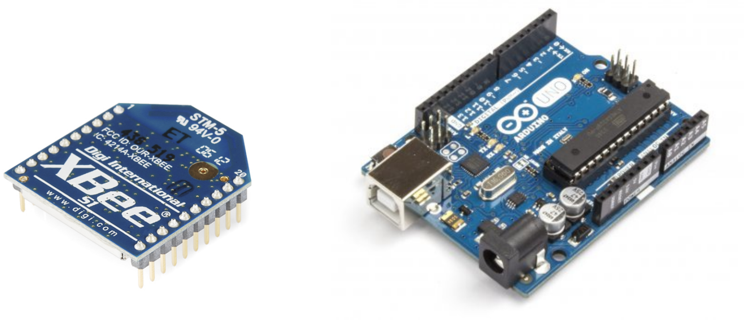
3.2.3 Controller

As a controller layer for our WPTN, the Arduino Uno board was chosen, see Figure 3.4(b). Arduino is an open-source hardware platform based on a simple input/output board with a microcontroller and an integrated software development environment. The Arduino Uno board is based on ATmega328 microcontroller. It has 14 digital input/output pins (used to control the communication module and powerharvester) and 6 analog inputs (used for measurements).

(a) WPT transmitter

(b) WPT receiver

Figure 3.3: Hardware used in the WPTN implementation: (a) WPT transmitter — Powercast TX91501-3W, and (b) WPT receiver — Powercast P1110-EVB. Source: [40].



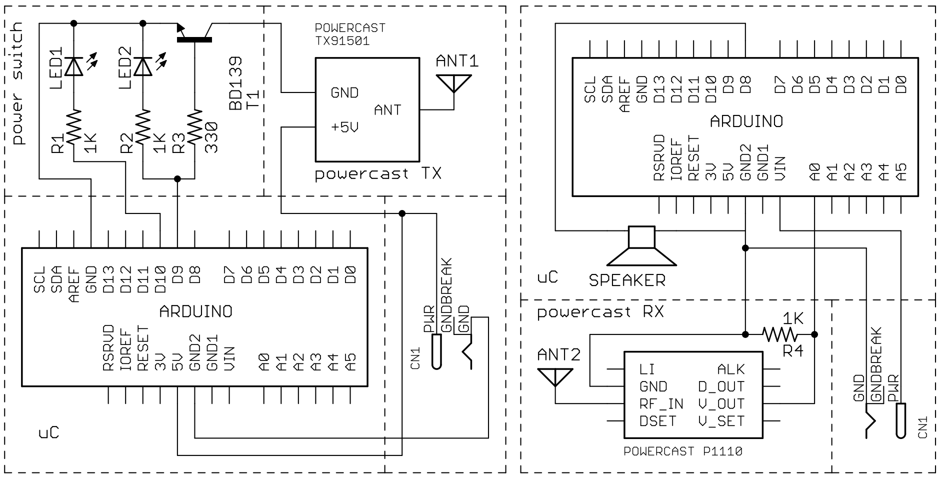
(a) XBee 802.15.4

(b) Arduino Uno

Figure 3.4: Hardware used in the WPTN implementation: (a) communication layer (XBee Series 1 - 802.15.4), and (b) Controller (Arduino Uno). Source: [42].

3.2.4 WPT transmitter and receiver

Each ETx, see Fig. 3.5(a), is connected with the mains power through a transistor switch controlled by the Arduino Uno board [42]. Analogically, the ERx emulator is controlled by the same Arduino board, see Fig. 3.5(b). Note that we use the word 'emulator', as the ERx is still connected to the power supply. This choice is dictated by the simplicity of the design and unavailability of the ID-based wake-up radios at the time of the experiment. Another reason for connecting the ERx to the power supply is the fact that our implementation contains not only a WPTN protocol module, but also measuring and data collection modules. Our implementation should be considered mainly as an evaluation testbed for various wireless power transfer protocols. All Arduino Uno boards are equipped with Wireless SD



(a) ETx

(b) ERx emulator

Figure 3.5: Components of the implemented WPTN: (a) ETx, and (b) ERx emulator and charge measuring unit. Notes: LED1 and LED2 are used for ETx state indication purpose; the value of R4 is user-changeable.

Shields with XBee 802.15.4 [41] modules attached. XBee modules are used to provide the communication layer for our devices. Each XBee device is using firmware version 10ec and is configured to work as an end-node in a frame-based API mode and given an unique 16-bit address. The rest of the XBee configuration parameters has a default value. Each device logs its measurements and events to an SD card that is placed in the slot of the Wireless SD Shield.

3.3 Rectifier efficiency

The ERx rectifier can be characterized by a certain efficiency of RF-to-DC conversion. The efficiency of this conversion depends on the input power to the rectifier (power received through the ERx antenna). As described in Subsection 3.2.1, a power harvesting receiver does not have a linear characteristic of input (AC) to output (DC) power. This is the case not only for the receiver of our choice, but is valid in general if it comes to converting one type of energy into another (e.g. RF-to-DC or mechanical to electrical).

To gain more insight into the characteristic of efficiency of a rectifier of our choice (see Figure 3.2) we have preformed an experiment to understand what factors influence it. The main factor that influences efficiency is the load that is connected to the rectifier, see Figure 3.6(a). According to the technical documentation of the P1110 receiver [40, /products/powerharvester-receivers], the device impedance is internally matched to the 50 Ω antenna.

However, the documentation does not specify how the internal impedance changes with the load that is attached to the device. In order to investigate this, we have changed the regular WPT scenario (see Fig. 3.6(a)) with controlled measurements using a signal generator (see Fig. 3.6(b)). Different resistors were attached as a load to the P1110 device and the voltage across those resistors was measured. Subsequently, the output power was calculated using formula $P = \frac{U^2}{R}$, with P — output power, U — voltage on the attached load, and R — resistance of the load.

In our setup, Agilent E4438C vector signal generator was generating the signal of the same frequency as the power transmitter used in our setup (Powercast TX91501-3W) - 915 MHz. The voltage level was measured using a Dynatek 9020a multimeter attached to the load.

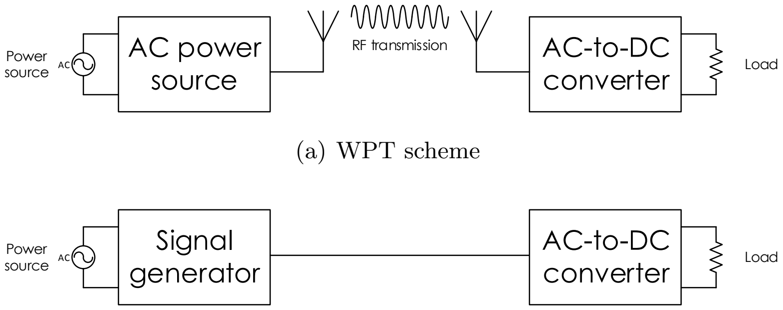
The measurement results are presented in Figure 3.7. Figure 3.7(a) depicts the measurements of the input AC power received through the antenna to DC power on the load connected to the rectifier, for 4 different loads: 470 Ω, 1 kΩ, 10 kΩ and 22 kΩ. As we can see from this figure, for each load there is a certain level of input power (marked in the figure), for which the output becomes saturated - there is no further increase of output power with the increase of input power.

We define the efficiency of RF-to-DC conversion as $P_{RFtoDC} = \frac{P_{out}}{P_{in}}$, where P_{out} — output power of the rectifier (power delivered to the attached load), P_{in} — input power delivered to the antenna connector of the rectifier. Figure 3.7(b) shows the calculated efficiency in relation to AC input power for the same four loads as in Figure 3.7(a). These measured and calculated efficiencies are compared with the efficiency data taken from the Powercast datasheet. There is a major discrepancy between our measured values and data provided by the producer of the P1110. The reason could be the fact that our generated signal was different from the signal of the TX91501-3W transmitter — we generated a 915 MHz sinusoid, while the TX91501-3W generates a DSSS modulated signal. Another reason could be the fact that for each input power level the highest efficiency is for a different load. If the input power level is low, the efficiency will be high for a high resistance of the connected load. For high values of input power efficiency becomes higher for a lower resistance of the load. A thorough investigation of the reasons of this difference are outside the scope of this work.

The main conclusion of this investigation is the fact that there is a strong non-linearity of RF-to-DC conversion, that further complicates designing the optimal WPT network.

3.4 Proposed protocols

As mentioned in Subsection 3.1.2, the goal of WPTN protocols is to optimize the behavior of the network, namely *minimize energy consumption in the*



(a) WPT scheme

(b) RF-to-DC efficiency measurements

Figure 3.6: (a) WPT scheme. The transmitter transmits the energy, and the receiver converts it from radio frequency AC to DC and feeds it into the load; (b) Measurements of RF-to-DC efficiency.

network while maximizing charging. This is a challenging task due to the fact there are multiple goals such an optimization needs to meet: limiting the time transmitters are in charging state while maximizing the time receivers are receiving energy. If we consider a network of transmitters trying to charge a group of receivers present in the network, there are a few questions a protocol needs to answer:

1. What does it mean that ERx is being charged efficiently?
2. Which subset of ETx devices needs to be in charging state so that given ERx is charged efficiently?
3. How to decide which subset of ETx devices needs to be in charging state in case there are multiple ERx devices in the network?

We consider two approaches to solving the above problems: beaconing-based and probing-based protocols. They differ fundamentally in the way they address above problems, especially problem 2 and 3.

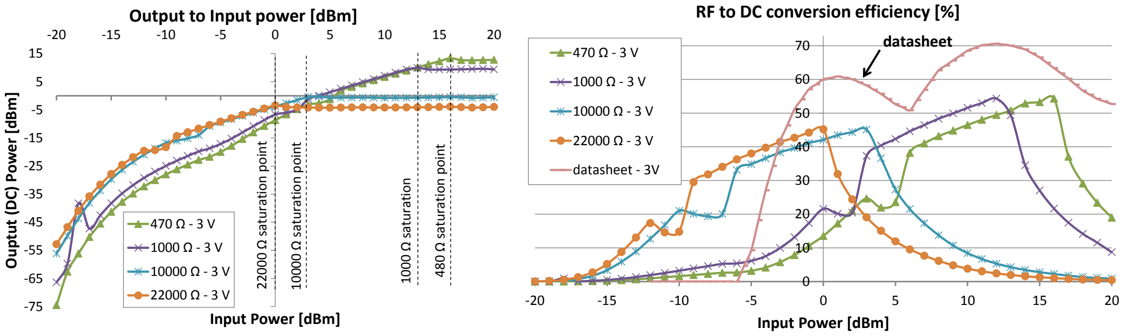
3.4.1 Beaconing-based protocol

Services provided

The beaconing-based protocol provides a functionality of choosing the subset of ETx devices that will be in charging mode when an ERx requests energy charging. It provides procedures for initiating and terminating the energy transfer.

Assumptions regarding environment

The main assumptions regarding the WPTN environment were listed in Subsection 3.1.2. Those are:



(a) P1110 output to input power characteristics

(b) P1110 RF-to-DC conversion efficiency

Figure 3.7: Results of P1110 device measurements: (a) output to input power characteristics for different loads, and (b) RF-to-DC conversion efficiency; refer to Figure 3.6(b) for hardware details.

- The WPTN is deployed in an office environment;
- ERx devices are mobile;
- ETx devices are static.

There are additional assumptions that hold for a beaconing-based protocol, and which are different from a probing-based protocol described in the Subsection 3.4.2, namely:

- ETx devices are independent, not connected in the network;
- ERx devices do not have a capability of measuring the power they harvest;
- ERx devices are capable of reading the RSSI of the control layer packets;
- ERx devices are in constant need of charging.

All devices use the IEEE 802.15.4 protocol for physical layer and media access control at the 2.4 GHz frequency, working at 9600 bit/s.

Vocabulary and encoding of messages

Each packet is enclosed in an 802.15.4 frame. On the reception of the packet it is possible to read the RSSI of this particular packet. The only message in the beaconing-based protocol is a Charge Request Message ($\mathtt{REQ}_{\mathtt{CRG}}$).

Procedure rules for message exchanges

In this section a detailed explanation of the protocol will follow. First, a high-level description of the protocol execution is provided. Subsequently, a more formal description, based on finite state machines (FSMs) is introduced. Next, both the description and the FSM-based definition are combined in the form of pseudocode of the protocol, see Algorithm 1. Finally, there are two examples of Message Sequence Charts (MSCs) explained — an optimistic and a pessimistic one.

High-level description The beaconing-based protocol is a simple protocol for enabling efficient wireless power transfer. This protocol uses the signal strength of received packets of the communication layer to reason about the possibility of efficient power charging. From the receiver point of view, there are two phases of the protocol: the Charging Request Phase and the Charging Phase. The Charging Request Phase is a phase in which an ERx requires charging and is not being charged. In this phase the ERx broadcasts a Charge Request message ($\mathtt{REQ_{CRG}}$) every fixed interval of time - $t^{\mathrm{ERx}}_{\mathrm{Ping}}$ (see Table 3.2 on page 44 for the full list of both beaconing and probing based protocol parameters). If an ETx device receives a $\mathtt{REQ_{CRG}}$ packet with a signal strength higher than a threshold $t^{\mathrm{ETx}}_{\mathrm{CommTh}}$, it starts charging the ERx. At this point the protocol is in the so-called Charging Phase. The assumption here is that if the RSSI of the $\mathtt{REQ_{CRG}}$ received at the ETx is above $t^{\mathrm{ETx}}_{\mathrm{CommTh}}$, then charging can be successfully performed. This is not always the case and depends largely on the propagation environment and the choice of $t^{\mathrm{ETx}}_{\mathrm{CommTh}}$. The ETx will continue to charge the ERx as long as it receives $\mathtt{REQ_{CRG}}$ messages with the RSSI above $t^{\mathrm{ETx}}_{\mathrm{CommTh}}$ at least every fixed interval of $t^{\mathrm{ETx}}_{\mathrm{CrgReq}}$. If there is no successful $\mathtt{REQ_{CRG}}$ reception within $t^{\mathrm{ETx}}_{\mathrm{CrgReq}}$ time, the ETx will stop charging the given ERx. The important characteristic of the protocol is the one-to-one relation between ERx and ETx. For each ETx to continue charging an ERx, the following condition needs to be fulfilled: *the time between subsequent receptions of the $\mathtt{REQ_{CRG}}$ messages from the ERx is smaller than t^{ETx}_{CrgReq} AND the $\mathtt{REQ_{CRG}}$ messages are received with an RSSI above t^{ETx}_{CommTh} .*

Finite State Machine based description The exact procedures for message exchanges are described with the use of finite state machines (FSMs). Based on those FSMs, pseudo-code is generated, that is subsequently implemented using the C++ programming language (see Chapter 4).

An FSM model is an abstract mathematical model that describes a machine that can be in a finite number of states. The FSM can be only in one state at the time and can change states when triggered by a certain event or condition (state transition). To describe the protocol we use a specific type of finite state machine, called Mealy machine. Mealy machines can be characterized by a 6- tuple $(S, S_0, \Sigma, \Delta, T, G)$, where:

- S — a set of states in which the FSM can be;
- S_0 — an initial state, which is an element from the set S;
- Σ — a set of input symbols (called the input alphabet);
- Δ — a set of output symbols (called the output alphabet);
- T — a transition function $T : Sx\Sigma \to S$ mapping pairs of state and an input symbol to the next state of a FSM;
- G — an output function $G : Sx\Sigma \to \Delta$ mapping pairs of state and an input symbol to the output symbol.

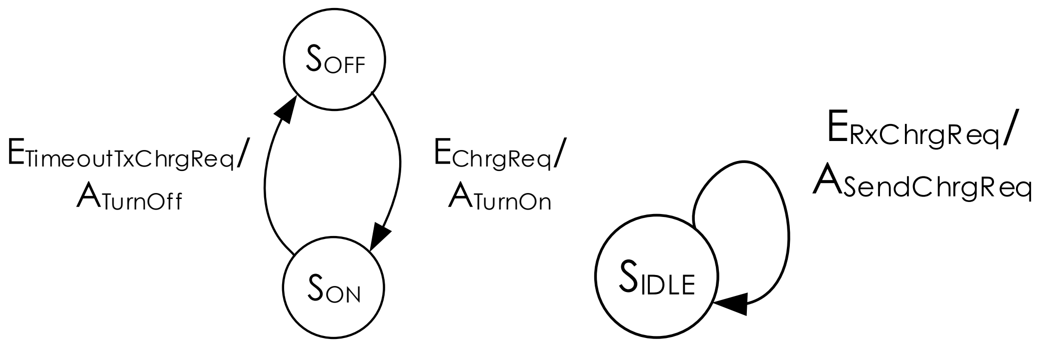
For the beaconing-based WPTN protocol two FSMs are defined — ERx and ETx FSMs.

The ETx FSM is defined as follows:

- $S \in (\mathsf{S_{OFF}}, \mathsf{S_{ON}})$, where:
  - $\mathsf{S_{OFF}}$: The ETx does not transmit power (the ETx is not in charging state);
  - $\mathsf{S_{ON}}$: The ETx does transmit power (the ETx is in charging state);
- $S_0 = \mathsf{S_{OFF}}$;
- $\Sigma \in (\mathsf{E_{TxChrgReq}}, \mathsf{E_{TimeoutTxChrgReq}})$, where:
  - $\mathsf{E_{TxChrgReq}}$: A charge request packet ($\mathtt{REQ}_{\mathsf{CRG}}$) was received from an ERx (the identity of the ERx does not matter for the protocol, any ERx sending a $\mathtt{REQ}_{\mathsf{CRG}}$ will activate the ETx);
  - $\mathsf{E_{TimeoutTxChrgReq}}$: Timeout event in case a charge request packet ($\mathtt{REQ}_{\mathsf{CRG}}$) was not received from an ERx for longer than $t^{\text{ETx}}_{\text{CrgReq}}$ time;
- $\Delta \in (\mathsf{A_{TurnOff}}, \mathsf{A_{TurnOn}})$, where:
  - $\mathsf{A_{TurnOff}}$: Turning off power transmission;
  - $\mathsf{A_{TurnOn}}$: Turning on power transmission;
- Transition function T and output function G are defined in Table 3.3.

The ERx FSM is defined as follows:

- $S \in (\mathsf{S_{IDLE}})$, where:


$\mathsf{S}_{\mathsf{OFF}}$
$\mathsf{E}_{\mathsf{TimeoutTxChrgReq}}$/
$\mathsf{A}_{\mathsf{TurnOff}}$
$\mathsf{E}_{\mathsf{ChrgReq}}$/
$\mathsf{A}_{\mathsf{TurnOn}}$
S_{ON}
$\mathsf{E}_{\mathsf{RxChrgReq}}$/
$\mathsf{A}_{\mathsf{SendChrgReq}}$
$\mathsf{S}_{\mathsf{IDLE}}$

(a) ETx FSM

(b) ERx FSM

Figure 3.8: Finite State Machines of beaconing-based protocols: (a) ETx FSM, and (b) ERx FSM.

  - $\mathsf{S}_{\mathsf{IDLE}}$: The ERx broadcasts $\mathtt{REQ}_{\mathtt{CRG}}$ packets periodically (every $t^{\text{ERx}}_{\text{Ping}}$) looking for the available ETx devices that could deliver energy to it. In this state the ERx device may receiving energy (when an appropriate ETx is in S_{ON}).

- $S_0 = \mathsf{S}_{\mathsf{IDLE}}$;
- $\Sigma \in (\mathsf{E}_{\mathsf{RxChrgReq}})$, where:
  - $\mathsf{E}_{\mathsf{RxChrgReq}}$: Event for sending charge request packet ($\mathtt{REQ}_{\mathtt{CRG}}$). This event is being executed every $t^{\text{ERx}}_{\text{Ping}}$;
- $\Delta \in (\mathsf{A}_{\mathsf{SendChrgReq}})$, where:
  - $\mathsf{A}_{\mathsf{SendChrgReq}}$: Action of charge request packet being send;
- Transition function T and output function G are defined in Table 3.4.

Both ERx and ETx FSMs of the beaconing-based protocol are presented in Figure 3.8. The details of the protocol implementation are provided in Algorithm 1. The set of parameters describing the implementation is given in Table 3.2. As a worst case scenario, in the implementation we assume that ERx is constantly in need of charging.

Algorithm based description Pseudocode of the beaconing-based protocol can be found in Algorithm 1.

Message Sequence Charts The Message Sequence Chart (MSC) scenarios are presented in Figure 3.9. They present an example execution of the protocol and relate to the different phases of the protocol and message exchanges. The scenario considered includes one ERx and three ETx devices.

Algorithm 1 Beaconing—ETx and ERx events

```
upon $\mathsf{E}_{\mathsf{RxChrgReq}}$ ()                              ▷ ERx event—Note (a)
    BROADCAST($\mathtt{REQ}_{\mathtt{CRG}}$)
upon $\mathsf{E}_{\mathsf{TxChrgReq}}$ ($\mathtt{REQ}_{\mathtt{CRG}}$)                    ▷ ETx event
    TURNONPOWERTRANSMISSION()
    STATE $\leftarrow \mathsf{S}_{\mathsf{ON}}$
upon $\mathsf{E}_{\mathsf{TimeoutTxChrgReq}}$ ()                       ▷ ETx event—Note (b)
    TURNOFFPOWERTRANSMISSION()
    STATE $\leftarrow \mathsf{S}_{\mathsf{OFF}}$
```

[a] Executed every $t^{\mathrm{ERx}}_{\mathrm{Ping}}$ s

[b] Executed in S_{ON} if $\mathtt{REP}_{\mathtt{PWR}}$ was not received for more than $t^{\mathrm{ETx}}_{\mathrm{PwrProbe}}$ s

The ERx is within communication range of all three ETx devices (all ETx devices can hear this ERx). Two scenarios are considered - optimistic, in which all ETxs can charge the given ERx, and pessimistic, where no ETx can charge the given ERx.

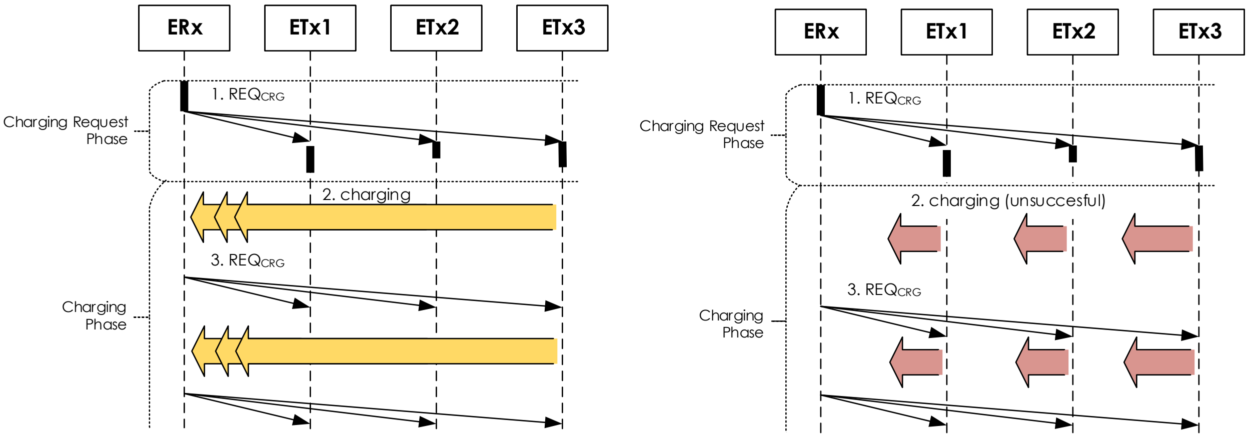
We first consider an optimistic MSC, see Figure 3.9(a). The ERx starts execution of the protocol in the Charging Request Phase. In this phase $\mathtt{REQ}_{\mathtt{CRG}}$ is being sent periodically. Each of ETx devices receives this request with sufficient signal strength ($t^{\mathrm{ETx}}_{\mathrm{CommTh}}$). It means each of those ETx devices will start charging the given ERx. The protocol enters the Charging Phase, during which ERx is being charged by all three ETxs. During that phase, ERx continues to periodically announce its presence through broadcasting $\mathtt{REQ}_{\mathtt{CRG}}$ messages.

The other situation considered is a pessimistic scenario, where no ETx can charge the ERx, see Figure 3.9(b). The execution of the protocol is exactly the same as in the optimistic scenario with the difference that there is no successful power transfer - even though all three ETx devices are charging ERx, the ERx cannot receive anything, because it is outside of the charging range of each of those ETx (but still within communication range).

The communication range should be matched with the charging range. Since in the beaconing-based protocol we do not have any feedback on the received power from the ERx, the only method to match those ranges is by modifying $t^{\mathrm{ETx}}_{\mathrm{CommTh}}$ parameter appropriately. The choice of this parameter will depend on many factors, such as propagation environment, transmitter and receiver antennas and the amount of harvested energy, that is considered useful. Any value will apply only to a specific case of certain network topology and environment.

Another point to consider is the fact that in the beaconing-based protocol there is no method to stop the energy transfer if it is not successful. As long as the ERx is within communication range of ETx devices, those ETx devices will try to charge the ERx, and thus waste their energy.

Those problems are addressed in the following Subsection 3.4.2, where the probing-based protocol is introduced.



(a) Optimistic scenario

(b) Pessimistic scenario

Figure 3.9: Example Message Sequence Charts (MSCs) of the beaconing-based protocol: (a) optimistic scenario, and (b) pessimistic scenario.

3.4.2 Probing-based protocol

The probing-based protocol works in phases. Three characteristic phases of the protocol are defined: the Charging Request Phase, the Probing Phase and the Charging Phase. The Charging Request Phase is the phase in which the ERx device is in the mode of looking for charging and ETx devices are waiting for the ERx to send a charging request message. On that happening, the protocol moves to the Probing Phase, where the ETx conducts power probing in order to find out if the ERx is not harvesting any energy from other sources at this moment. If the result of probing is positive (the ERx needs the given ETx to charge it), the ETx starts transmitting energy. If the energy is received successfully by the ERx (the conditions for the successful transmission are defined below), the protocol arrives in the Charging Phase, where the ERx is charged by the ETx device. If the power transmission is not successful or during the Probing Phase it turns out that the ERx is already harvesting the energy from other sources, there is no charging and the protocol arrives again at the Charging Request Phase. In the sections below details of the protocol are provided.

Services provided

The services provided by the probing-based protocol are the same as provided by the beaconing protocol (see Section 3.4.1). The protocol aims at meeting those goals in a more efficient manner thanks to the usage of a probing technique, which provides feedback on the actual power received at the ERx.

Assumptions regarding environment

Similarly to services provided by the probing protocol, also assumptions we make regarding the environment and network devices themselves are similar (see Section 3.4.1). There is one major difference in the probing-based protocol compared to the beaconing-based protocol regarding the capabilities of the ERx device.

The assumptions specific to the probing protocol are:

- ETx devices **are connected in the network** and are able to communicate with each other;
- ERx devices **have a capability** of measuring the power they harvest;
- Increasing the number of ETx devices charging one ERx device does not bring a benefit;
- ERx devices are capable of reading the RSSI of the control layer packet;
- ERx devices are in constant need of charging.

The first three assumptions are connected with each other. The rationale behind the third assumption is a consequence of the non-linearity of the energy conversion (see Section 3.3). It can happen that increasing the received power will lower down the efficiency, and eventually even the received power (even though the power transmitted is higher). In our designs we do not look for the most optimal scheme of charging (which is proven to be an NP-complete problem, see [43]). Instead, we decided to use heuristics in order to design and implement our WPTN.

Vocabulary and encoding of messages

Each packet from/to an ERx/ETx is enclosed in an IEEE 802.15.4 frame, with the frame header encapsulating the source and the destination address. In the protocols implementation of WTPN, on the reception of the packet, we allow to read the received signal strength indicator of this particular packet. The following packet types used in our WPTN implementation are introduced:

- $\texttt{REQ}_{\texttt{CRG}}$: packet with charging request, broadcasted every $t_{\text{Ping}}^{\text{ERx}}$ s in the Charging Request Phase by the ERx;
- $\texttt{REQ}_{\texttt{PWR}}$: power report packet request sent by an ETx to the ERx used in the Power Probing Phase;

- $\texttt{REP}_{\texttt{PWR}}$: packet containing two values: (i) voltage level on the load of the ERx—V and (ii) threshold level of the ERx— $t^{\mathrm{ERx}}_{\mathrm{VoltageTh}}$ [1]. $\texttt{REP}_{\texttt{PWR}}$ can be a response to $\texttt{REQ}_{\texttt{PWR}}$ (if in Power Probing Phase) or sent by the ERx unsolicited (if in Charging Phase).

Procedure rules for message exchanges

Similarly to the description of the beaconing-based protocol, this Section explains the protocol execution in detail. First, the high-level description of the protocol execution is provided. Subsequently, more formal description, based on finite state machines (FSMs) is introduced. Next, both the description and FSM-based definitions are combined in the form of pseudocode of the protocol, see Algorithm 2 and 3, and finally, there are two scenarios (optimistic and pessimistic) discussed, using Message Sequence Charts (MSCs).

High-level description The protocol executes in three phases: Charging Request Phase, Power Probing Phase and Charging Phase. As in the case of the beaconing-based protocol, it is assumed that ERx constantly requires charging.

In the Charging Request Phase the ERx, every $t^{\mathrm{ERx}}_{\mathrm{Ping}}$ s, broadcasts a $\texttt{REQ}_{\texttt{CRG}}$. This phase is similar to the beaconing-based protocol and $t^{\mathrm{ETx}}_{\mathrm{CommTh}}$ is used in this phase in the same way as in the beaconing-based protocol. At any time one or more ETx devices can receive the $\texttt{REQ}_{\texttt{CRG}}$ and initiate the Power Probing Phase. During the Power Probing Phase an ETx tries to find out if the ERx is already being charged by another ETx. After an ETx received a $\texttt{REQ}_{\texttt{CRG}}$ it will wait random time, distributed uniformly with a maximum $t^{\mathrm{ETx}}_{\mathrm{RandWait}}$ s (implemented as a simple collision avoidance scheme) and then send a $\texttt{REQ}_{\texttt{PWR}}$ to ERx from which the $\texttt{REQ}_{\texttt{CRG}}$ was received.

After the ERx receives a first $\texttt{REQ}_{\texttt{PWR}}$ it will ignore all subsequent $\texttt{REQ}_{\texttt{PWR}}$ packets from other ETxs in the current Power Probing Phase. In return the ERx sends a $\texttt{REP}_{\texttt{PWR}}$ that contains the current level of harvested energy. After the ERx sends a $\texttt{REP}_{\texttt{PWR}}$ in the Power Probing phase, it will wait a predefined time of $t^{\mathrm{ERx}}_{\mathrm{WaitForPwr}}$ s for the power transfer from the ETx after which (if no power was transferred) it concludes that power transfer from the ETx was unsuccessful. If the $\texttt{REP}_{\texttt{PWR}}$ received by the ETx contains a power level lower than a power threshold, $t^{\mathrm{ERx}}_{\mathrm{VoltageTh}}$, it means that the ERx is not currently harvesting energy and requires charging.

Subsequently the ETx tries to charge the ERx and the Charging Phase starts. If power will not be received, the ERx will go back to the Charging

[1] The reason for sending V and $t^{\mathrm{ERx}}_{\mathrm{VoltageTh}}$ from an ERx to the ETxs is due to ease of experiment result collection—V and debugging—$t^{\mathrm{ERx}}_{\mathrm{VoltageTh}}$. An ETx uses the $t^{\mathrm{ERx}}_{\mathrm{VoltageTh}}$ extracted from the packet instead of a pre-programmed one, therefore only the ERx needs to be re-programmed in order to change this parameter of the experiment.

Request phase. The ERx saves the address of the ETx that was unsuccessful in the Charging Phase in its internal queue, denoted as Q_{TX} .

All the addresses are kept in Q_{TX} for $t^{\mathrm{ERx}}_{\mathrm{RmvLast}}$ s. If the protocol is in the Power Probing state the ERx ignores all the ETxs with addresses stored in the Q_{TX} . This is done to prevent the ETx that was not successful to initiate the Power Probing Phase with the given ERx again before the network conditions change, e.g. the ERx moves to another position. In the Charging Phase, after the ETx starts charging the ERx, there is a possibility that the ERx harvests energy that is above $t^{\mathrm{ERx}}_{\mathrm{VoltageTh}}$ V. If this is the case the ERx will start sending unsolicited $\mathtt{REP}_{\mathtt{PWR}}$ to the current ETx (as the ERx keeps track of the ETx devices that tried to charge it). If the $\mathtt{REP}_{\mathtt{PWR}}$ packets are received by the ETx at least every $t^{\mathrm{ETx}}_{\mathrm{PwrProbe}}$ s, the ETx will continue charging a given ERx. If the ERx does not receive enough power, it will not send an $\mathtt{REP}_{\mathtt{PWR}}$ packet to the ETx within the specified time, which will result in the stop of power transmission from the ETx to the ERx.

Finite State Machine based description Similarly as for the beaconing-based WPTN protocol, also for the probing-based protocol two FSMs are defined — the ERx and the ETx FSMs.

The ETx FSM is defined as follows:

- $S \in$ ($\mathsf{S}_{\mathsf{OFF}}$, $\mathsf{S}_{\mathsf{PROBE}}$, S_{ON}), where:
  - $\mathsf{S}_{\mathsf{OFF}}$: The ETx does not transmit power;
  - $\mathsf{S}_{\mathsf{PROBE}}$: The ETx probes the ERx for its received power level before deciding to transmit power or not (move to $\mathsf{S}_{\mathsf{OFF}}$ or S_{ON});
  - S_{ON} : The ETx does transmit power;
- $S_0 = \mathsf{S}_{\mathsf{OFF}}$;
- $\Sigma \in$ ($\mathsf{E}_{\mathsf{TxChrgReq}}$, $\mathsf{E}_{\mathsf{PwrProbeAbove}}$, $\mathsf{E}_{\mathsf{TimeoutPwrProbeRsp}}$, $\mathsf{E}_{\mathsf{PwrProbeBelow}}$, $\mathsf{E}_{\mathsf{TimeoutOnPwrProbe}}$), where:
  - $\mathsf{E}_{\mathsf{TxChrgReq}}$: A charge request packet ($\mathtt{REQ}_{\mathtt{CRG}}$) was received from ERx;
  - $\mathsf{E}_{\mathsf{PwrProbeAbove}}$: A power probe report packet ($\mathtt{REP}_{\mathtt{PWR}}$) with a voltage level above $t^{\mathrm{ERx}}_{\mathrm{VoltageTh}}$ was received by ETx;
  - $\mathsf{E}_{\mathsf{TimeoutPwrProbeRsp}}$: Timeout event in case power a probe report ($\mathtt{REP}_{\mathtt{PWR}}$) was not received from ERx within $t^{\mathrm{ETx}}_{\mathrm{PwrProbeRsp}}$ after sending a power probe request packet ($\mathtt{REQ}_{\mathtt{PWR}}$);
  - $\mathsf{E}_{\mathsf{PwrProbeBelow}}$: A power probe report packet ($\mathtt{REP}_{\mathtt{PWR}}$) with a voltage level below $t^{\mathrm{ERx}}_{\mathrm{VoltageTh}}$ was received by ETx. It changes the state of ETx from $\mathsf{S}_{\mathsf{OFF}}$ to S_{ON} ;

  - $\mathsf{E}_{\mathsf{TimeoutOnPwrProbe}}$: Timeout event triggered when power a probe report packet ($\mathtt{REP}_{\mathtt{PWR}}$) was not received from ERx for more than $t^{\mathrm{ETx}}_{\mathrm{PwrProbe}}$. It changes the state of ETx from S_{ON} to $\mathsf{S}_{\mathsf{OFF}}$;

- $\Delta \in$ ($\mathsf{A}_{\mathsf{SendPwrProbeReq}}$, $\mathsf{A}_{\mathsf{TurnOff}}$, $\mathsf{A}_{\mathsf{TurnOn}}$), where:
  - $\mathsf{A}_{\mathsf{SendPwrProbeReq}}$: Action of sending a power probe request ($\mathtt{REQ}_{\mathtt{PWR}}$) to the ERx from which the charge request packet ($\mathtt{REQ}_{\mathtt{CRG}}$) was received;
  - $\mathsf{A}_{\mathsf{TurnOff}}$: Turning off power transmission;
  - $\mathsf{A}_{\mathsf{TurnOn}}$: Turning on power transmission;

- Transition function T and output function G are defined in Table 3.5.

The ERx FSM is defined as follows:

- $S \in$ ($\mathsf{S}_{\mathsf{IDLE}}$, $\mathsf{S}_{\mathsf{WAIT}}$, $\mathsf{S}_{\mathsf{CHRG}}$), where:
  - $\mathsf{S}_{\mathsf{IDLE}}$: State in which the ERx is in need of charging and broadcasts charge request packets ($\mathtt{REQ}_{\mathtt{CRG}}$) to the nearby ETxs;
  - $\mathsf{S}_{\mathsf{WAIT}}$: State in which the ERx has received a power probe request packet ($\mathtt{REQ}_{\mathtt{PWR}}$) from one of the ETxs and responded with a power level below $t^{\mathrm{ERx}}_{\mathrm{VoltageTh}}$. The ERx is then awaiting for power transmission. If power transmission is successful, the ERx moves immediately to $\mathsf{S}_{\mathsf{CHRG}}$. Successful transmission means that the voltage on the load R4 connected to the ERx is higher than $t^{\mathrm{ERx}}_{\mathrm{VoltageTh}}$, see Figure 3.5(b) on page 21. If the transmission is not successful (there is a communication link between the ERx and the ETx, but no power transmission link), after $t^{\mathrm{ERx}}_{\mathrm{WaitForPwr}}$ the ERx moves again to $\mathsf{S}_{\mathsf{IDLE}}$;
  - $\mathsf{S}_{\mathsf{CHRG}}$: State in which the ERx is being charged. In this state only power probe report packets ($\mathtt{REP}_{\mathtt{PWR}}$) are being send to the ETx that is charging the ERx. There is an association made between those two nodes in the network and they communicate only with each other (without broadcast packets);

- $S_0 = \mathsf{S}_{\mathsf{IDLE}}$;

- $\Sigma \in$ ($\mathsf{E}_{\mathsf{RxChrgReq}}$, $\mathsf{E}_{\mathsf{NewPwrProbe}}$, $\mathsf{E}_{\mathsf{VoltageAbove}}$, $\mathsf{E}_{\mathsf{RxPwrPr}}$, $\mathsf{E}_{\mathsf{TimeoutWaitForPwr}}$, $\mathsf{E}_{\mathsf{RmvOldest}}$, $\mathsf{E}_{\mathsf{VoltageBelow}}$), where:
  - $\mathsf{E}_{\mathsf{RxChrgReq}}$: Event for sending charge request packet ($\mathtt{REQ}_{\mathtt{CRG}}$). This event is being executed every $t^{\mathrm{ERx}}_{\mathrm{Ping}}$ if the ERx is in $\mathsf{S}_{\mathsf{IDLE}}$;
  - $\mathsf{E}_{\mathsf{NewPwrProbe}}$: Event of power probe request packet ($\mathtt{REQ}_{\mathtt{PWR}}$) being received from the ETx, which address is not in Q_{TX};

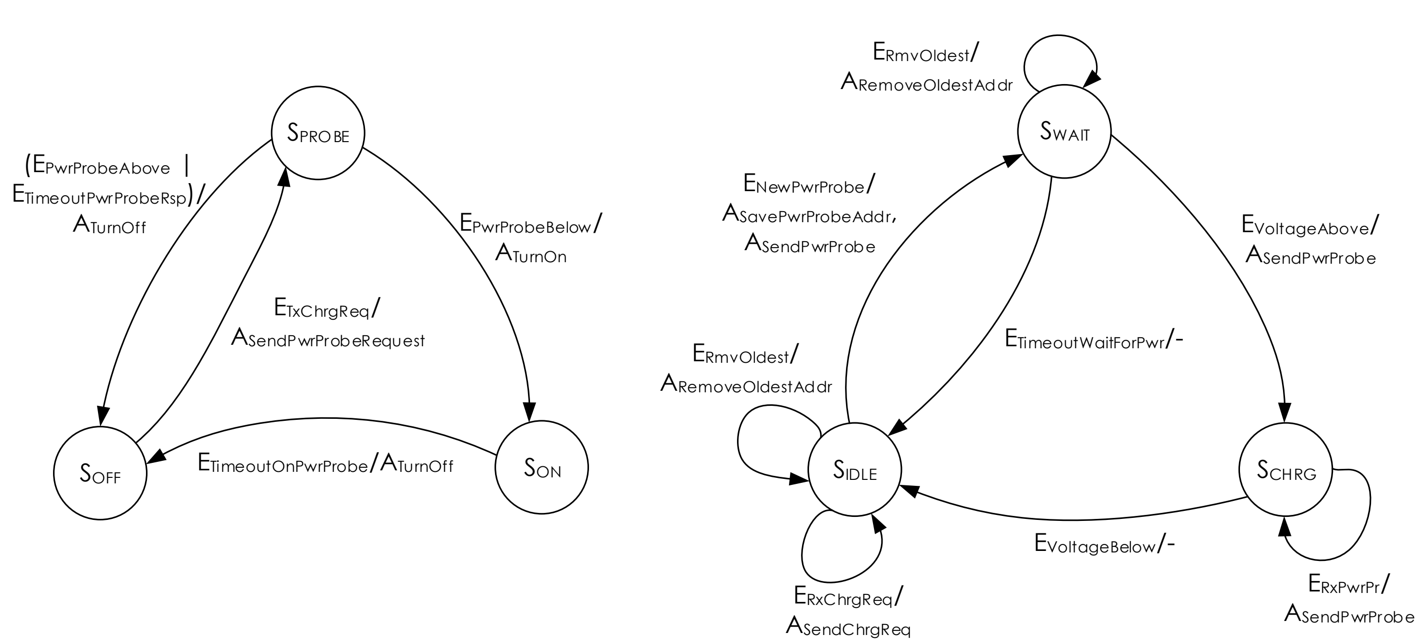
- $\mathsf{E}_{\mathsf{VoltageAbove}}$: Event of voltage level on the connected load being above $t^{\text{ERx}}_{\text{VoltageTh}}$;
- $\mathsf{E}_{\mathsf{RxPwrPr}}$: Event for sending a power probe report packet ($\mathtt{REP}_{\mathtt{PWR}}$). This event is being executed every $t^{\text{ERx}}_{\text{PwrProbe}}$ if the ERx is in $\mathsf{S}_{\mathsf{CHRG}}$;
- $\mathsf{E}_{\mathsf{TimeoutWaitForPwr}}$: Timeout event for the ERx waiting for power in state $\mathsf{S}_{\mathsf{WAIT}}$. It is triggered if a time longer than $t^{\text{ERx}}_{\text{WaitForPwr}}$ has passed from entering $\mathsf{S}_{\mathsf{WAIT}}$;
- $\mathsf{E}_{\mathsf{RmvOldest}}$: The oldest entry of Q_{TX} is being removed. This event is executed for a given entry after $t^{\text{ERx}}_{\text{RmvLast}}$ from this entry being added to the queue;
- $\mathsf{E}_{\mathsf{VoltageBelow}}$: Event of voltage level on the connected load being below $t^{\text{ERx}}_{\text{VoltageTh}}$;

- $\Delta \in (\mathsf{A}_{\mathsf{SendChrgReq}}, \mathsf{A}_{\mathsf{SendPwrProbe}}, \mathsf{A}_{\mathsf{SavePwrProbeAddr}}, \mathsf{A}_{\mathsf{RemoveOldestAddr}})$, where:
  - $\mathsf{A}_{\mathsf{SendChrgReq}}$: Action of a charge request packet ($\mathtt{REQ}_{\mathtt{CRG}}$) being send;
  - $\mathsf{A}_{\mathsf{SendPwrProbe}}$: Action of a power probe report packet ($\mathtt{REP}_{\mathtt{PWR}}$) being send;
  - $\mathsf{A}_{\mathsf{SavePwrProbeAddr}}$: Action of saving the address of the ETx that has sent power probe request packet ($\mathtt{REQ}_{\mathtt{PWR}}$) to Q_{TX} queue;
  - $\mathsf{A}_{\mathsf{RemoveOldestAddr}}$: Action of removing the oldest ETx address from the Q_{TX} queue;
- Transition function T and output function G are defined in Table 3.6.

Both ERx and ETx FSMs of the probing-based protocol are presented in Figure 3.10.

Algorithm based description The pseudocode of the probing-based protocol was divided in two parts: the ERx part, see Algorithm 2 and the ETx part, see Algorithm 3.

Message Sequence Charts The Message Sequence Chart (MSC) scenarios for the probing-based protocol are presented in Figure 3.11. As in the case of the beaconing-based protocol analysis (see Section 3.4.1) the scenario considered includes one ERx and three ETx devices. ERx is within communication range of all three ETx devices (all ETx devices can hear this ERx) and the same two scenarios are considered - optimistic, in which all ETxs can charge given ERx, and pessimistic, where no ETx can charge given ERx.



(a) ETx FSM

(b) ERx FSM

Figure 3.10: Finite State Machines of the probing-based protocol: (a) ETx FSM, and (b) ERx FSM. When the ERx is in $\mathsf{S}_{\mathsf{IDLE}}$, the ETx is in state $\mathsf{S}_{\mathsf{OFF}}$ (the Charging Request phase); when the ERx is in $\mathsf{S}_{\mathsf{WAIT}}$, the ETx is in state $\mathsf{S}_{\mathsf{PROBE}}$ (the Power Probing phase); and finally when the ERx is in $\mathsf{S}_{\mathsf{CHRG}}$, it means the ETx is in state S_{ON} (the Charging phase).

We first consider an optimistic MSC, see Figure 3.11(a). ERx starts the execution of the protocol in the Charging Request Phase. In this phase $\mathtt{REQ}_{\mathtt{CRG}}$ is being sent periodically. Each of ETx devices receives this request with sufficient signal strength ($t^{\mathrm{ETx}}_{\mathrm{CommTh}}$). On the reception of $\mathtt{REQ}_{\mathtt{CRG}}$ each of ETx devices sends $\mathtt{REQ}_{\mathtt{PWR}}$ to the ERx. ERx accepts the $\mathtt{REQ}_{\mathtt{PWR}}$ packet that it received first and associates itself with this ETx (ETx1 in Figure 3.11(a)). $\mathtt{REQ}_{\mathtt{PWR}}$ packets from other ETxs in communication range from ERx are ignored by ERx. At this moment the protocol is in the Power Probing Phase. ERx sends $\mathtt{REP}_{\mathtt{PWR}}$ packet back to the ETx with which it is associated. This happens to ensure ERx is not receiving energy from other sources (such as background radiation, other ERxs in the network or another energy source, such as battery). Based on the received $\mathtt{REP}_{\mathtt{PWR}}$, ETx decides if it will charge ERx or not. In case of our scenario ETx decides to charge ERx, because ERx was not being charged at the time of the Power Probing Phase (a voltage level on the load attached to the ERx, reported in $\mathtt{REP}_{\mathtt{PWR}}$ was below the $t^{\mathrm{ERx}}_{\mathrm{VoltageTh}}$ value). When ETx starts charging ERx, and ERx successfully receives transmitted energy, it manifests it by transmitting unsolicited $\mathtt{REP}_{\mathtt{PWR}}$.

Algorithm 2 Probing—ERx events

```
upon $E_{\text{RxChrgReq}}$ ()                                   ▷ Note (a)
    BROADCAST($\texttt{REQ}_{\texttt{CRG}}$)
upon $E_{\text{NewPwrProbe}}$ ($\texttt{REQ}_{\texttt{PWR}}$, ETx $i$)        ▷ Note (b)
    if STATE = $S_{\text{IDLE}}$ then
        $Q_{\text{TX}} \leftarrow$ENQUEUE(ETx $i$ address)
        SEND($\texttt{REP}_{\texttt{PWR}}$, ETx $i$)
        STATE $\leftarrow S_{\text{WAIT}}$
upon $E_{\text{VoltageAbove}}$ (ETx $i$)                           ▷ Note (c)
    SEND($\texttt{REP}_{\texttt{PWR}}$, ETx $i$)
    STATE $\leftarrow S_{\text{CHRG}}$
upon $E_{\text{RxPwrPr}}$ ()                                     ▷ Note (d)
    SEND($\texttt{REP}_{\texttt{PWR}}$, ETx $i$=$Q_{\text{TX}}$ (first))
upon $E_{\text{TimeoutWaitForPwr}}$ ()                           ▷ Note (e)
    STATE $\leftarrow S_{\text{IDLE}}$
upon $E_{\text{RmvOldest}}$ ()                                   ▷ Note (f)
    $Q_{\text{TX}} \leftarrow$DEQUEUE($Q_{\text{TX}}$ (last))
upon $E_{\text{VoltageBelow}}$ ()                                ▷ Note (g)
    STATE $\leftarrow S_{\text{IDLE}}$
```

[(a)] Every $t^{\text{ERx}}_{\text{Ping}}$ s if ERx is in S_{IDLE}
[(b)] When $\texttt{REQ}_{\texttt{PWR}}$ received from ETx i and ETx i address $\notin Q_{\text{TX}}$
[(c)] When ERx in S_{WAIT} receives power and load voltage exceeds $t^{\text{ERx}}_{\text{VoltageTh}}$ V
[(d)] Every $t^{\text{ERx}}_{\text{PwrProbe}}$ s if ERx is in S_{CHRG}
[(e)] When ERx in S_{WAIT} and no power from ETx for more than $t^{\text{ERx}}_{\text{WaitForPwr}}$ s
[(f)] When oldest address in Q_{TX} has been stored longer than $t^{\text{ERx}}_{\text{RmvLast}}$ s
[(g)] When in S_{CHRG} and attached load voltage drops below $t^{\text{ERx}}_{\text{VoltageTh}}$ V

Currently, the protocol is already in the Charging Phase. $\texttt{REP}_{\texttt{PWR}}$ packets are transmitted periodically as long as there is energy received by ERx.

In the second scenario considered (see Figure 3.11(b)), the execution of the protocol is very similar to the optimistic scenario. The difference starts when the energy transmission from the first ETx to ERx is not successful (in the Power Probing Phase). Since ERx never received the energy transfer it was waiting for, it did not send an unsolicited $\texttt{REP}_{\texttt{PWR}}$ packet to ETx, but moved back to the Charging Request Phase instead. If ETx does not receive unsolicited $\texttt{REP}_{\texttt{PWR}}$ within $t^{\text{ETx}}_{\text{turnOff}}$, it will stop transmitting energy. In the new Charging Request Phase previous, unsuccessful ETx is excluded (it is excluded by ERx for $t^{\text{ERx}}_{\text{RmvLast}}$). In this phase, even though the $\texttt{REQ}_{\texttt{CRG}}$ is broadcasted, the response from the previous ETx (ETx1) will always be ignored, thus this exchange is not displayed in Figure 3.11(b). Out of all remaining ETx devices the one will associate with the ERx, from which the ERx will hear the $\texttt{REQ}_{\texttt{PWR}}$ first, and from then the exchange proceeds further in the same way as in the optimistic scenario.

The reason for the exclusion of an unsuccessful ETx for a certain period

Algorithm 3 Probing—ETx events

```
upon $\mathsf{E}_{\mathsf{TxChrgReq}}$ ($\texttt{REQ}_{\texttt{CRG}}$ , ERx $i$)                ▷ Note (a)
    if STATE = $\mathsf{S}_{\mathsf{OFF}}$ then
        waitRandom($t^{\text{ETx}}_{\text{RandWait}}$ )                    ▷ Uniform distribution
        send($\texttt{REQ}_{\texttt{PWR}}$ , $i$)
        STATE $\leftarrow$ $\mathsf{S}_{\mathsf{PROBE}}$
upon $\mathsf{E}_{\mathsf{PwrProbeAbove}}$ ($\texttt{REP}_{\texttt{PWR}}$ , ERx $i$, $V$)          ▷ Note (b)
    if STATE = $\mathsf{S}_{\mathsf{PROBE}}$ then
        STATE $\leftarrow$ $\mathsf{S}_{\mathsf{OFF}}$
upon $\mathsf{E}_{\mathsf{PwrProbeBelow}}$ ($\texttt{REP}_{\texttt{PWR}}$ , ERx $i$, $V$)          ▷ Note (c)
    if STATE = $\mathsf{S}_{\mathsf{PROBE}}$ then
        turnOnPowerTransmission()
        STATE $\leftarrow$ $\mathsf{S}_{\mathsf{ON}}$
upon $\mathsf{E}_{\mathsf{TimeoutPwrProbeRsp}}$ ()                              ▷ Note (d)
    STATE $\leftarrow$ $\mathsf{S}_{\mathsf{OFF}}$
upon $\mathsf{E}_{\mathsf{TimeoutOnPwrProbe}}$ ()                               ▷ Note (e)
    turnOffPowerTransmission()
    STATE $\leftarrow$ $\mathsf{S}_{\mathsf{OFF}}$
```

(a) When $\texttt{REQ}_{\texttt{CRG}}$ received from ERx i

(b) When $\texttt{REP}_{\texttt{PWR}}$ received from ERx i with $V \geq t^{\text{ERx}}_{\text{VoltageTh}}$

(c) When $\texttt{REP}_{\texttt{PWR}}$ received from ERx i with $V < t^{\text{ERx}}_{\text{VoltageTh}}$

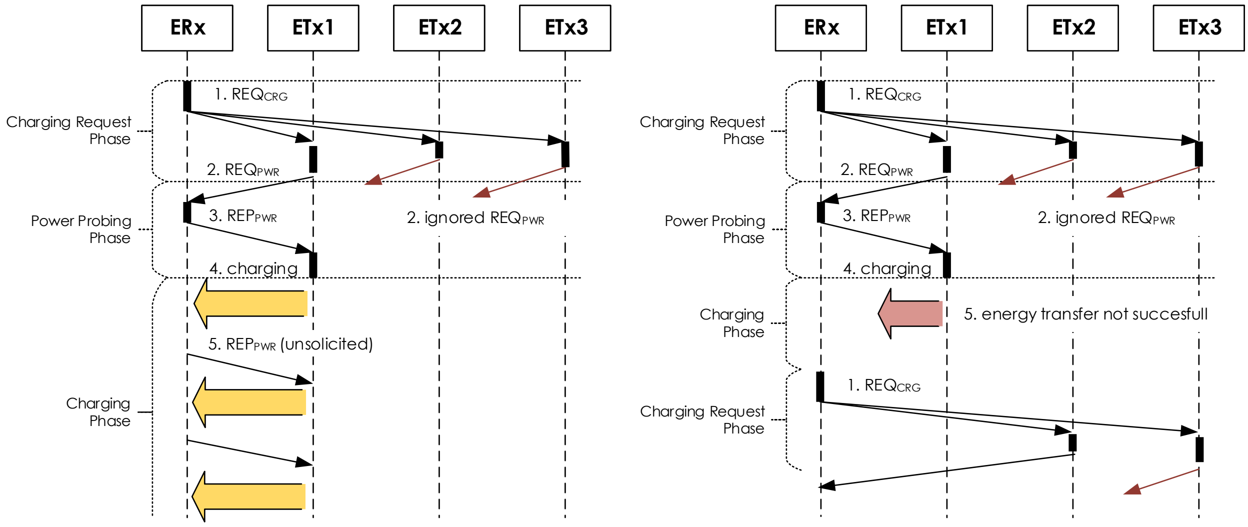
(d) When in $\mathsf{S}_{\mathsf{PROBE}}$ after sending $\texttt{REQ}_{\texttt{PWR}}$ the $\texttt{REP}_{\texttt{PWR}}$ from the ERx not received for more than $t^{\text{ETx}}_{\text{PwrProbeRsp}}$ s

(e) When in S_{ON} and no $\texttt{REP}_{\texttt{PWR}}$ from the ERx has been received for more than $t^{\text{ETx}}_{\text{PwrProbe}}$ s

of time is the need to avoid the resource starvation problem (see [44, Section 6.7.4, p. 459]). This is a well known problem in dynamic, concurrent systems, such as our WPTN. In case of scenario from Figure 3.11(b), if no exclusion mechanism would be in place, it could happen that every time after a Charging Request Phase the ERx associates itself with ETx1 (which cannot transmit energy successfully). This could happen if ETx1 has a faster processor than other ETx devices or in case of unfavorable race conditions between ETx devices (which are not synchronized with each other). The reason ETx is excluded for a finite time are the changing propagation conditions in the network, such as the mobility of an ERx device. Propagation conditions could enable earlier excluded device to charge the ERx successfully at some future point in time.

In case of probing-based protocols one can notice that the communication range does not have to be so precisely matched with the power transmission range. In case energy transfer is not possible, the protocol will back off and start a similar procedure with other ETx devices. If any of ETx devices is within communication and energy transfer range, it will eventually start charging the ERx.

The probing-based protocol generally assumes that one ETx charges one



(a) Optimistic scenario

(b) Pessimistic scenario

Figure 3.11: Example Message Sequence Charts (MSCs) of probing-based protocol: (a) optimistic scenario, and (b) pessimistic scenario.

ERx. That was the design thought. It could however happen that in the charging range of an ETx device that is transmitting energy to a specific ERx (ERx1) device, another ERx (ERx2) appears. ERx2 will start its protocol execution in the Charging Request Phase. However, the transmitted energy will be received through the ERx antenna and delivered to the load. In this situation any attempt to associate ERx2 with another ETx will fail, because during Power Probing Phase ERx2 will report that it is harvesting energy (although officially it is not in the Charging Request Phase). It is thus possible in the protocol to receive energy from a transmitter without being associated with it. This is the heuristic solution that, together with the proper design of the locations of ETx devices, should minimize the energy consumed by ETx devices while maximizing charging at the ERx devices.

3.5 Mathematical analysis

In this section we answer the question how long does it take from an ERx appearing in the network to the ERx receiving energy. The reason we investigate this parameter is the fact that beaconing-based and probing-based protocols are fundamentally different during this initialization phase, where charging did not start yet. We do basic analysis for the beaconing-based protocol and extended, Markov chain analysis for the probing-based protocol.

3.5.1 Description of the scenario

In the analysis scenario we consider one ERx and N ETx devices. Out of those N ETx devices, the ERx is in charging range of K of those devices $(0 \leq K \leq N)$. It means that only K out of N devices can charge the ERx. Also, we assume a static situation - the ERx does not move, thus N and K do not change and we consider only the expected time to start charging the ERx for a given N and K. The specific values of N and K depend on the design and topology of the network, that were not considered in this thesis. We also assume ERx is not being charged by any external sources and it requires charging. This is an important assumption for a probing-based protocol, because it means the protocol will always change its state from the Power Probing Phase to the Charging Phase (and will stay in the Charging Phase as long as charging is successful).

3.5.2 Time analysis of protocol phases

The time analysis of the protocol is being done based on the phases of the protocols and the MSC diagrams in Figures 3.9 and 3.11. Our goal in the following analysis is to analyze the duration of the phases of both beaconing-based and probing-based protocols and derive formulas for expected time to charge an ERx device in the network. In order to do that, we need to define certain time intervals of events happening in the protocol. Let us assume certain time variables, that enable us to model the delays in the system:

- $T_{chargeRequest}$ — a random delay connected with the $t^{\mathrm{ERx}}_{\mathrm{Ping}}$ parameter, symbolizing the time from the random appearance of ERx in the network to the moment when it sends the first $\mathtt{REQ}_{\mathtt{CRG}}$ packet; it is a random time distributed uniformly: $U(0,t^{\mathrm{ERx}}_{\mathrm{Ping}})$;

- T_{pckt} — the latency of sending a packet in the network, because of the fact all packets in our implementation have similar size, we do not differentiate between latencies of different types of packets; more detailed analysis could include this type of differences too; it is expected for this delay of be of the order of milliseconds;

- T_{proc} — the processing delay that happens between reception of a packet and response to it; it is expected for this delay of be of the order of milliseconds;

- T_{power} — a random variable symbolizing the time it takes from a decision of an ETx device to turn on power transmission to an ERx device efficiently harvesting power; this value is affected by the instruction execution timing of an Arduino, switching time of the transistor T1 (see Figure 3.5(a)), internal workings of Powercast TX91501-3W device, propagation delay, characteristic of Powercast P1110 energy harvester and input delay of ERx Arduino; it is expected for this delay to be of the order of milliseconds or lower.

Using above random variables it is possible to model the behavior of an ERx—ETx pair of devices. For each protocol we have an optimistic and pessimistic scenario. The optimistic scenario depicts a situation when exchange of communication between ERx and ETx results in successful energy transfer. The pessimistic scenario is the one in which such an exchange of communication results in unsuccessful energy transfer. Important difference between the beaconing-based and the probing-based protocol is the fact that the probing-based protocol has a chance to **recover from the pessimistic scenario**, see Figure 3.11(b), while the beaconing-based scenario will stay in that scenario indefinitely, see Figure 3.9(b). For both protocols we can consider two different times: T_{opt} — optimistic time it takes from an ERx appearing in the network to receiving energy for optimistic scenario; and T_{pes} — the pessimistic time it takes from an ERx appearing in the network to unsuccessful attempt of energy transmission (and potential recovery in case of probing-based protocol).

Those delays are presented below:

Beaconing-based protocol

- $T^{B}_{opt} = T_{chargeRequest} + T_{pckt} + T_{proc} + T_{power}$
- $T^{B}_{pes} = T_{chargeRequest} + T_{pckt} + T_{proc} + T_{power}$

Probing-based protocol

- $T^{P}_{opt} = T_{chargeRequest} + 3(T_{pckt} + T_{proc}) + T_{power}$
- $T^{P}_{pes} = T_{chargeRequest} + 3(T_{pckt} + T_{proc}) + t^{\text{ERx}}_{\text{WaitForPwr}}$

In further analysis we assume that T_{pckt}, T_{proc} and T_{power} are negligible compared to $T_{chargeRequest}$ and $t^{\text{ERx}}_{\text{WaitForPwr}}$. Moreover, for beaconing-based protocol we have only one round - as soon as ERx is within communication range, ETx devices will attempt to charge the device. Thus introducing T_{pes} for beaconing-based protocol is not practical. The new equations become:

Beaconing-based protocol

- $T^{B}_{opt} = T_{chargeRequest} = U(0, t^{\text{ERx}}_{\text{Ping}})$

Probing-based protocol

- $T^{P}_{opt} = T_{chargeRequest} = U(0, t^{\text{ERx}}_{\text{Ping}})$
- $T^{P}_{pes} = T_{chargeRequest} + t^{\text{ERx}}_{\text{WaitForPwr}} = U(0, t^{\text{ERx}}_{\text{Ping}}) + t^{\text{ERx}}_{\text{WaitForPwr}}$

In the analysis of the time taken to start charging we consider the mean values of T^B_{opt}, T^P_{opt} and T^P_{pes} — $\overline{T^B_{opt}}$, $\overline{T^P_{opt}}$ and $\overline{T^P_{pes}}$ respectively.

In further analysis we use a notion of rounds. One round of the protocol is one exchange of messages between ERx and ETxs. In the beaconing-based protocol one round constitutes of the ERx broadcasting $\texttt{REQ}_{\texttt{CRG}}$ and the ETxs responding with power transfer. The duration of that round is denoted as $T^{\mathrm{B}}_{\mathrm{opt}}$. In the probing-based protocol one round consists of an ERx broadcasting a $\texttt{REQ}_{\texttt{CRG}}$, one of ETxs responding with $\texttt{REQ}_{\texttt{PWR}}$, ERx responding with $\texttt{REP}_{\texttt{PWR}}$ and finally ETx transmitting power to ERx. We define this round to be optimistic if the power is received by ERx and pessimistic when ERx does not receive power transmitted by ETx in that round. The duration of an optimistic round in the probing protocol is denoted as $T^{\mathrm{P}}_{\mathrm{opt}}$; the duration of a pessimistic round as $T^{\mathrm{P}}_{\mathrm{pes}}$. Note that we do not consider pessimistic rounds in the beaconing-based protocol. The reason for that is that in the beaconing-based protocol there is always one round, regardless of the fact if the ERx received power or not, since there is no mechanism to provide feedback on the power received.

3.5.3 Time to start charging

Beaconing-based protocol

For the beaconing-based protocol the analysis is simple. There is only one round of charging, after which all ETx devices that are within communication range of the ERx will be turned on. If an ERx randomly appears in the network, the time to charge will be a random variable: $T^B_{start} = T_{opt}$. We define two different cumulative distribution functions (CDF): the first one in order to compare the beaconing-based protocol with the probing-based protocol — $F_{\overline{T}^B_{start}}(t)$ — and the second one in order to compare the beaconing-based protocol behavior with the measurements — $F_{T^B_{start}}(t)$. $F_{T^B_{start}}(t)$ is defined under the assumption that $T^B_{start} = U(0, t^{\mathrm{ERx}}_{\mathrm{Ping}})$:

$$F_{T^B_{start}}(t) = \frac{t}{t^{\mathrm{ERx}}_{\mathrm{WaitForPwr}}} \quad \text{for } t \in [0, t^{\mathrm{ERx}}_{\mathrm{WaitForPwr}}] \tag{3.1}$$

This matches the real world experiment, where the ERx appears randomly in the network with this distribution. However, in order to compare the CDF of the beaconing-based protocol with the CDF of the probing-based protocol, we want to show the effect of how many rounds the protocol takes to initialize. In such a case, the random distribution of T^B_{start} is not so important, thus we simplify the equations using the average value of $\overline{T}^B_{start}$:

$$F_{\overline{T}^B_{start}}(t) = \begin{cases} 0 & \text{if } t \leq \overline{T^B_{opt}} \\ 1 & \text{if } t > \overline{T^B_{opt}} \end{cases} = H(t - \overline{T^B_{opt}}) \tag{3.2}$$

where $H(x)$ is the Heaviside step function.

Probing-based protocol

The probing based protocol works in rounds. A successful round starts with the Charging Request Phase, continues to the Power Probing Phase and ends in the Charging Phase, in which the protocol stays, successfully charging the ERx. However, if the charging is not successful, the protocol goes back to the Charging Request phase. In the new Charging Request phase the previous ETx, that unsuccessfully attempted to charge ERx, is excluded from the network. Therefore, for the first round, we have N transmitters and K transmitters that could charge the ERx. If we choose one of the $N-K$ ETx devices that could not charge the ERx, we exclude it in the next round, which starts with $N-1$ transmitters and K transmitters that could charge the ERx. The process is repeated until a correct ETx is chosen (the choice is being made on a first-come, first-served basis). We can use a Markov chain to describe this random process. This Markov chain is defined using equations (3.3) and (3.4). Equation (3.5) shows cumulative distribution function (CDF) of time to charging for the probing-based protocol.

The probability that successful charging happens in the first round equals $P(X_1') = \frac{K}{N}$. Since eventually we are interested in the number of unsuccessful rounds, we look closer into the probabilities of unsuccessful charging. This probability for the first round equals $P(X_1) = \frac{N-K}{N}$. Subsequently, the probability that charging will be unsuccessful for two rounds is equal to $P(X_2) = \frac{N-K}{N}\frac{N-1-K}{N-1}$. The second component in this equation is changed from N to $N-1$, because in the second round one unsuccessful ETx is eliminated and we start with a network of $N-1$ ETx devices within communication range and K devices within charging range. To generalize this result, the probability that for i rounds ERx will not be successfully charged is

$$P(X_i) = \frac{N-K}{N}\frac{N-1-K}{N-1}...\frac{N-(i-1)-K}{N-(i-1)} = \prod_{j=0}^{i-1}\frac{N-j-K}{N-j} \tag{3.3}$$

Finally, we can consider a probability of successful charging in round i, which is equal to $P(X_i') = P(X_{i-1}) * \frac{K}{N-(i-1)}$. The full equation is

$$P(X_i') = \frac{K}{N-(i-1)}\prod_{j=0}^{i-2}\frac{N-j-K}{N-j}, i \in [1, N-K+1] \tag{3.4}$$

and the CDF of the time to start charging is

$$F_{T^P_{start}}(t) = \begin{cases} 0 & \text{if } t \leq 0 \\ \sum_{i=1}^{\left\lfloor \frac{t-\overline{T^P_{opt}}}{\overline{T^P_{pes}}} - 1 \right\rfloor} P(X'_i) & \text{if } 0 \leq t < (N-K)\overline{T^P_{pes}} + \overline{T^P_{opt}} \\ 1 & \text{if } t \geq (N-K)\overline{T^P_{pes}} + \overline{T^P_{opt}} \end{cases} \tag{3.5}$$

Table 3.2: Protocol parameter values used in the WPTN experiment implementation

Symbol	Type[a]	Description	Set value
$t^{\text{ETx}}_{\text{CrgReq}}$	B	In S_{ON}—feedback timer within which unsolicited $\texttt{REQ}_{\texttt{CRG}}$ packets from ERx need to be received	8 s
$t^{\text{ETx}}_{\text{CommTh}}$	B P	Received signal strength value below which a packet from ERx is ignored when received[b]	–70 dBm
$t^{\text{ERx}}_{\text{Ping}}$	B P	Time between two consecutive $\texttt{REQ}_{\texttt{CRG}}$ packets being broadcasted by ERx (if in $\mathsf{S}_{\mathsf{IDLE}}$)	4 s
$t^{\text{ETx}}_{\text{PwrProbeRsp}}$	P	Time an ETx waits for $\texttt{REP}_{\texttt{PWR}}$ after sending $\texttt{REQ}_{\texttt{PWR}}$	4 s
$t^{\text{ERx}}_{\text{RmvLast}}$	P	Time for which each ETx address is stored in the Q_{TX} queue	30 s
$t^{\text{ETx}}_{\text{turnOff}}$	P	In S_{ON}—waiting time for the first (unsolicited) $\texttt{REP}_{\texttt{PWR}}$ (sent by ERx on transition from $\mathsf{S}_{\mathsf{WAIT}}$ to $\mathsf{S}_{\mathsf{CHRG}}$)	2 s
$t^{\text{ETx}}_{\text{PwrProbe}}$	P	In S_{ON}—feedback timer within which unsolicited $\texttt{REP}_{\texttt{PWR}}$ packets from ERx need to be received	8 s
$t^{\text{ERx}}_{\text{PwrProbe}}$	P	In $\mathsf{S}_{\mathsf{CHRG}}$—time interval between two $\texttt{REP}_{\texttt{PWR}}$ packets sent by an ERx to the ETx currently charging that ERx	4 s
$t^{\text{ETx}}_{\text{RandWait}}$	P	The maximum time ETx waits before sending $\texttt{REQ}_{\texttt{PWR}}$ after receiving $\texttt{REQ}_{\texttt{CRG}}$ from ERx[c]	0.5 s
$t^{\text{ERx}}_{\text{WaitForPwr}}$	P	The maximum time ERx will wait for power from ETx while being in state $\mathsf{S}_{\mathsf{WAIT}}$	4 s
$t^{\text{ERx}}_{\text{VoltageTh}}$	P	Voltage threshold for a load being attached to the microcontroller[d]	0.5 V

[a] Protocol type: B—Beaconing, P—probing

[b] Parameter used to simulate different levels of communication layer power transmission/coverage

[c] Timer used to avoid collisions at ERx when multiple ETx hear the same $\texttt{REQ}_{\texttt{CRG}}$ and send $\texttt{REQ}_{\texttt{PWR}}$ immediately

[d] If the voltage level is above this threshold the power level is considered to be sufficient to initiate charging

State:	$\mathbf{S}_{\textbf{OFF}}$		$\mathbf{S}_{\textbf{ON}}$	
Input:	T	G	T	G
$\mathbf{E}_{\textbf{TxChrgReq}}$	S_{ON}	$\mathsf{A}_{\text{TurnOn}}$	—	—
$\mathbf{E}_{\textbf{TimeoutTxChrgReq}}$	—	—	S_{OFF}	$\mathsf{A}_{\text{TurnOff}}$

Table 3.3: ETx FSM — beaconing-based protocol

Input:	$\mathbf{S}_{\textbf{IDLE}}$	
State:	T	G
$\mathbf{E}_{\textbf{RxChrgReq}}$	S_{IDLE}	$\mathsf{A}_{\text{SendChrgReq}}$

Table 3.4: ERx FSM — beaconing-based protocol

State:	$\mathbf{S}_{\textbf{OFF}}$		$\mathbf{S}_{\textbf{PROBE}}$		$\mathbf{S}_{\textbf{ON}}$	
Input:	T	G	T	G	T	G
$\mathbf{E}_{\textbf{TxChrgReq}}$	$\mathsf{S}_{\text{PROBE}}$	$\mathsf{A}_{\text{SendPwrProbeReq}}$	—	—	—	—
$\mathbf{E}_{\textbf{PwrProbeAbove}}$	—	—	S_{OFF}	$\mathsf{A}_{\text{TurnOff}}$	—	—
$\mathbf{E}_{\textbf{TimeoutPwrProbeRsp}}$	—	—	S_{OFF}	$\mathsf{A}_{\text{TurnOff}}$	—	—
$\mathbf{E}_{\textbf{PwrProbeBelow}}$	—	—	S_{ON}	$\mathsf{A}_{\text{TurnOn}}$	—	—
$\mathbf{E}_{\textbf{TimeoutOnPwrProbe}}$	—	—	—	—	S_{OFF}	$\mathsf{A}_{\text{TurnOff}}$

Table 3.5: ETx FSM — probing-based protocol

State:	$\mathbf{S}_{\textbf{IDLE}}$		$\mathbf{S}_{\textbf{WAIT}}$		$\mathbf{S}_{\textbf{CHRG}}$	
Input:	T	G	T	G	T	G
$\mathbf{E}_{\textbf{RxChrgReq}}$	—	$\mathsf{A}_{\text{SendChrgReq}}$	—	—	—	—
$\mathbf{E}_{\textbf{NewPwrProbe}}$	S_{WAIT}	$\mathsf{A}_{\text{SavePwrProbeAddr}}$, $\mathsf{A}_{\text{SendPwrProbe}}$	—	—	—	—
$\mathbf{E}_{\textbf{VoltageAbove}}$	—	—	S_{CHRG}	$\mathsf{A}_{\text{SendPwrProbe}}$	—	—
$\mathbf{E}_{\textbf{RxPwrPr}}$	—	—	—	—	—	$\mathsf{A}_{\text{SendPwrProbe}}$
$\mathbf{E}_{\textbf{TimeoutWaitForPwr}}$	—	—	S_{IDLE}	—	—	—
$\mathbf{E}_{\textbf{RmvOldest}}$	—	$\mathsf{A}_{\text{RemoveOldestAddr}}$	—	$\mathsf{A}_{\text{RemoveOldestAddr}}$	—	—
$\mathbf{E}_{\textbf{VoltageBelow}}$	—	—	—	—	S_{IDLE}	—

Table 3.6: ERx FSM — probing-based protocol

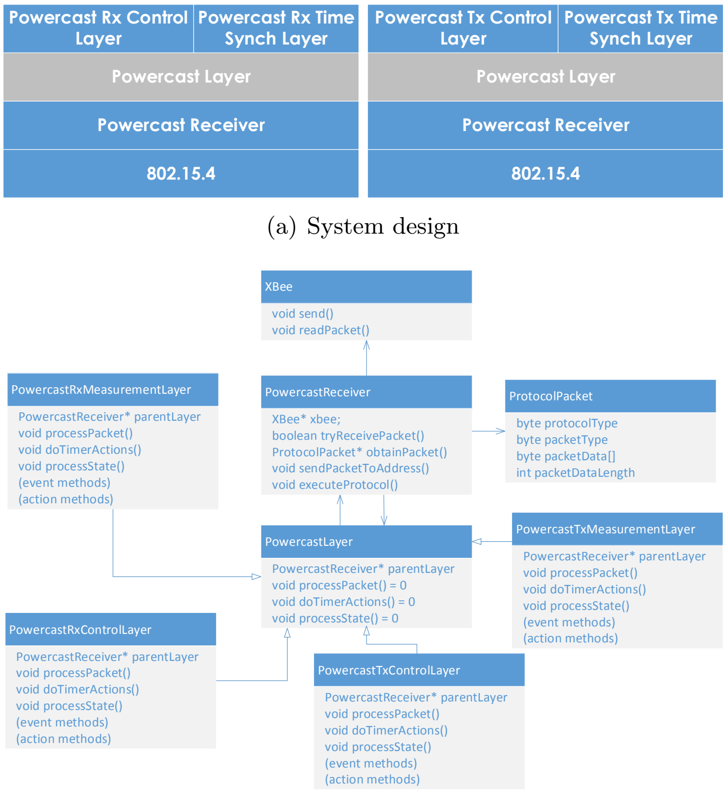
Chapter 4

Implementation

4.1 System design

The WPTN protocol was implemented using a layered design. This type of design is commonly used in telecommunication systems to implement protocol stacks. The system design is presented in Figure 4.1(a). The software design follows the hardware design described in Section 3.2. The bottom layer is the IEEE 802.15.4 standard we use for communication. Above this physical and media access control (MAC) layer our custom Protocol Receiver layer is created. This layer is responsible for sending and receiving messages from the lower 802.15.4 layer. It translates 802.15.4 protocol data units (PDUs) to custom Powercast Receiver PDUs that are being transferred to the upper layers (and vice versa). The reason for this layer is to abstract the physical layer from the implementation of the specific WPTN protocol. In our design if the physical layer is changed, only the Powercast Receiver layer needs to be adopted to facilitate translation of Powercast Receiver PDUs to the new type of MAC layer PDUs. The alternative forms of physical layer could be, for example, IEEE 802.15.1, IEEE 802.11 or custom built solutions. Alternatively, the physical layer could be abstracted away to a simulation environment. The architecture of the system enables easy integration of the software stack with simulation tools such as NS-3, OPNET and OmNET++.

The higher layer is a Powercast Layer. This layer is in fact an abstract layer that is being executed by the Powercast Receiver layer. This is the layer where protocols running in the WPTN are being implemented. There are two protocols running in the network - the WPTN control protocol and the time synchronization protocol. The time synchronization protocol is placed there only to enable measurement synchronization in the network. It is required for processing the results in Chapter 5. The WPTN control protocol (Powercast Rx Control Layer and Powercast Tx Control Layer on Figure 4.1(a)) are implementations of the WPTN control protocol. We have



(a) System design

(b) Software architecture

Figure 4.1: Implementation of WPTN system: (a) system design, and (b) software architecture.

tested two implementations: the beaconing-based implementation and the probing-based implementation.

4.2 Implementation

4.2.1 Software architecture

The software architecture is presented in the UML diagram in Figure 4.1(b). It reflects the system design principles described in Section 4.1. The main three classes are the XBee class, the Powercast Receiver class and the PowercastLayer class. They reflect the 802.15.4 layer, Powercast Receiver layer and the Powercast Layer from Figure 4.1(a), respectively.

The XBee class contains two functions that are important from this project point of view: send() and readPacket(), responsible for sending and receiving packets from the physical medium.

The main object that runs the WPTN device is the PowercastReceiver class object. It contains the XBee class object pointer, which is being called to receive and to send packets. This is the main object that is being instantiated in the main loop of the microcontroller of a WPTN device. Its function executeProtocol() is being executed continuously in the main loop of the microcontroller. This function will try to receive a packet from the XBee object using the tryReceivePacket() function. If a packet is available (tryReceivePacket() returns true), the available packet will be converted to the ProtocolPacket object using the obtainPacket() function and passed to the higher layers of the protocol (control or measurement layer, depending on the protocolType field that is part of every message). In order to achieve that, the executeProtocol() function triggers the processPacket() functions of both the measurement and control layer. Subsequently, the doTimeActions() functions of both the measurement and control layer are triggered (the ProtocolLayer functions are explained below). You can see the details of the implementation of this function in Listing A.1.

The PowercastLayer is an abstract layer that defines three abstract methods that need to be implemented in child classes: processPacket(), doTimerActions() and processState(). The goal of the processPacket () function is to generate events associated with a packet that was being passed from the PowercastReceiver layer and then trigger the processState () function. The processState() function is a direct implementation of the finite state machines of the beaconing-based or probing-based protocol (see Figure 3.8 and 3.10 respectively). doTimeActions() is responsible for generating events that happen with a fixed time interval, regardless of received packets. An example of such an event can be an event for sending a $\mathsf{REQ}_{\mathsf{CRG}}$ packet every $t^{\mathrm{ERx}}_{\mathrm{Ping}}$ if ERx is in $\mathsf{S}_{\mathsf{IDLE}}$ (see Section 3.4.2). The processState() function invokes actions associated with the current state and current event.

Each child classes of the PowercastLayer class implements those three functions and implements the event methods and action methods (see example Listing A.2).

4.2.2 Implementation of control protocol

A control protocol in our WPTN system is a protocol that enables wireless power charging. The beaconing-based and probing-based protocols described in Chapter 3 are examples of WPTN control protocols, in contrast to the measurement protocol described in Section 4.2.3, that is designed to enable measurements of the control protocol. We have implemented two control protocols: beaconing-based and probing-based, which will be described below.

Implementation of beaconing-based protocol

The implementation of the beaconing-based protocol follows directly the FSMs described in Tables 3.3 and 3.4. The characteristics of the events in both ETx and ERx devices do not require any additional data structures (unlike the probing-based protocol, which requires the implementation of an extra queue).

Implementation of probing-based protocol

The implementation of the probing-based protocol follows the FSMs described in Tables 3.5 and 3.6. Compared with the beaconing-based protocol, in the probing-based protocol ERx devices have a capability to measure and report the harvested power to the ETx. Thus, apart from a $\texttt{REQ}_{\texttt{CRG}}$ message, which is present in both beaconing-based and probing-based protocols, additionally $\texttt{REQ}_{\texttt{PWR}}$ and $\texttt{REP}_{\texttt{PWR}}$ messages are implemented. Those messages enable the ETx to send a request to the ERx to which the response contains the value of the currently harvested energy. This facilitates creation of the harvested energy based feedback mechanism that is at the core of the probing-based protocol.

Yet another additional complexity of the probing-based protocol is associated with the fact that events defined in the probing-based protocol require memory of certain events in the system, namely addresses of ETx devices that were failing to associate themselves with the given ERx. Those addresses are being stored in the Q_{TX} queue for $t^{\mathrm{ERx}}_{\mathrm{RmvLast}}$ before removing it from the queue through the $\mathsf{E}_{\mathsf{RmvOldest}}$ event. The capabilities of the Q_{TX} are to store addresses of ETx devices for a fixed time ($\mathsf{E}_{\mathsf{RmvOldest}}$) and check if a given ETx address is in the queue. In the current implementation we needed to limit the size of this queue (due to limited memory available at Arduino platform). It means that an ERx can keep track only of a limited number of ETx devices that were trying to charge this ERx. For our experiment we have chosen the size of four to be the maximum size of the queue. The reason for this choice is the fact that in our experiments we were not using more than four ETx devices.

From the implementation point of view, Q_{TX} uses one array of unsigned 16-bit integers - txAdressesQueue - to store addresses of ETx devices (that are 16-bit integers themselves). There is another array that stores timestamps corresponding to the time of removal of ETx device address from the Q_{TX} - timestampsForRemovalQueue. Addresses are saved in an unsigned long queue with a timestamp value after which they should be removed from the queue (function saveAddressToQueue).

The implementation of this queue is presented in Listing A.4.

4.2.3 Implementation of measurement protocol

In order to make meaning out of measurements of the network, we need to have a method of aligning measurements made on different devices in the time domain. Our system can be seen as an asynchronous system. We have a number of ERx devices and a number of ETx devices that communicate with each other to achieve a certain goal, i.e. efficient wireless power charging. Regardless of the hardware platform used, in a setting with multiple devices the clocks of those devices will differ (even when initially they would be set to the same value, which is not possible to achieve in our platform, where we start devices manually).

Even though clock skew does not affect the performance of the WPTN control protocol, it makes it difficult to make sense of measurements done on different devices. The type of measurements made in our WPTN system are time-based measurements. It means what we will want to make a measurement of time-domain signals at a number of devices (e.g. harvested energy in ERx devices and consumed energy at ETx devices). In order to make meaning out of those measurements, we need to align them in the time domain. For that we need time synchronization between devices.

To enable that, we have implemented a simple time synchronization protocol. The idea of the time-synchronization protocol is simple:

- ERx is a clock source;
- other devices in the network receive clock values from ERx.

As an implementation, a simple FSM has been defined. The behavior of the ERx FSM is to periodically broadcast clock update messages; the behavior of the ETx FSM is to update its clock on the reception of a clock update message from the ERx. The problem of accuracy of time synchronization is described further in Section 4.3.

The reason of implementing a separate clock-synchronization protocol is the principle of separation of concerns. An alternative could be using the beaconing-based or probing-based protocol itself to synchronize the ETx devices with the ERx device. Such a design would, however, result in different synchronization quality for different parameters of the WPTN control protocol. Implementing clock-synchronization as an independent module minimizes the effect the WPTN control protocol has on it and allows for independent control of the time synchronization according to the needs of the experiment.

4.3 Accuracy of time synchronization

As described above, we need to have a method of aligning measurements made on different devices in the time domain. The main parameter of the

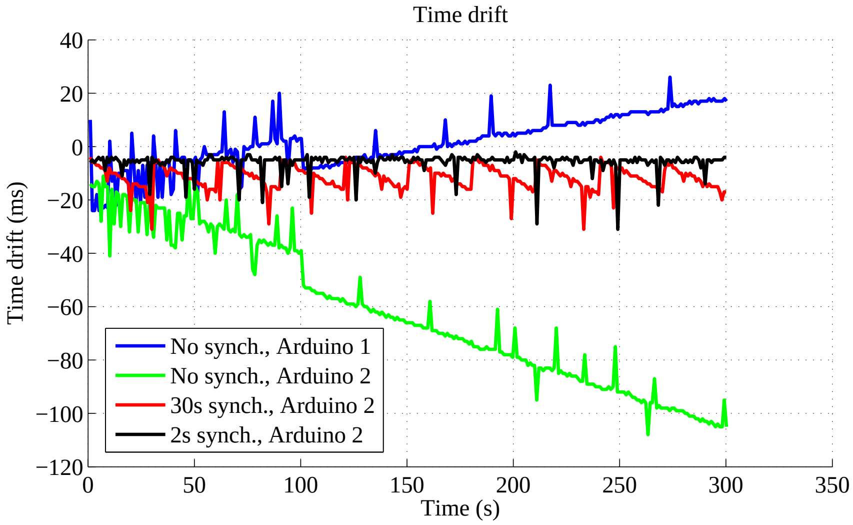


Figure 4.2: Time drift for multiple Arduino devices (Arduino 1 and Arduino 2) for no synchronization, 30 s synchronization interval, and 2 s synchronization interval.

time synchronization protocol is t_{synch}— variable saying how often do we perform time synchronization. Figure 4.2 shows the results of time drift measurements we performed to prove time synchronization is needed and to choose the right t_{synch}value.

The measurement set up is the following: we used one ETx device and one ERx device for the measurement. We configure beaconing-based protocol with $t_{\text{Ping}}^{\text{ERx}}$ of 1 s. The value of $t_{\text{Ping}}^{\text{ERx}}$ is chosen so that it is more frequent than the most frequent timer in the beaconing-based or probing-based protocol. Timestamps of sending a $\texttt{REQ}_{\texttt{CRG}}$ message are logged both on the ERx and ETx side. The measurement duration is 300 s. The measurements were done for three values of t_{synch}: 2 s, 30 s and >300 s. The value >300 s is marked on Figure 4.2 as *No synch.* The exact value of t_{synch}in that case is not important, since from an experiment point of view the effect is equal to this value being set to infinity — the clocks at ERx and ETx are synchronized only once at the beginning of the experiment. In other cases the time synchronization is performed every 30 s and 2 s, respectively. The measurement was done for two different ETx devices (marked *Arduino 1* and *Arduino 2*). The purpose of that is to show that different ETxs, even if equipped with the same hardware, will have different characteristics of time drift. That depends on the precision of the oscillators that are being used by the microcontroller.

The time synchronization is performed in the following manner: the ERx broadcasts its current timestamp. On reception of this broadcast message each ETx device[1] takes this timestamp as his own. After the experiment timestamps registered from the ERx are subtracted from the ETx. The result is a value that is the sum of T_{pckt}, T_{proc} and the actual time drift (see Section 3.5). Calculated in such a way the time drift is therefore the difference between the time of the reception of a $\texttt{REQ}_{\texttt{CRG}}$ at th4 ETx and the time of the ERx broadcasting it. It is not possible to eliminate T_{pckt} and T_{proc} from the measurement in a simple way. All the measurements were made under the assumption that the values of T_{pckt} and T_{proc} are negligible compared with other events in the network (order of microseconds for those values and order of seconds for other events in the WPTN).

The results of the measurements show the result of time drift for two different ETx devices and no synchronization. It can be noticed that the time drift of one of the ETx devices is positive and increasing, while the other one is negative and decreasing. A positive time drift means that the ETx is faster than the ERx. A negative time drift means the opposite. In case of no synchronization the measurement starts with time drift close to zero. As the experiment progresses, the time drift increases in a linear fashion. For the time synchronization of 30 s we can see that approximately every 30 s the time drift returns to its correct, close to zero value. These are the moments where time synchronization between ERx and ETx happens. During the 30 s time when no time synchronization happens the effect of time drift is still noticeable (characteristic saw-like plot). The last measurement is made for time synchronization done every 2 s. For this interval the time drift is constant and close to zero. The reason for that is that 2 s is too short to notice the effect of time drift between clocks on different microcontrollers.

Throughout every measurement made we can see certain spikes in time drift of values of approximately 10-20 ms. The reason for that is connected with all the measurements being logged and saved to the SD card. During every run of the loop there is usually one measurement or event logged. Sometimes it may happen though that during one run there is one measurement and one event to save to SD card. There is also certain cache between SD card and writing it using Arduino API. When this cache is written to the SD card, the run of the loop takes slightly longer. That is the reason for single spikes happening over the measurement,

An important conclusion to this analysis is the importance of time synchronization. If there is no time synchronization implemented, time drift is linearly increasing or decreasing. If we introduce periodical time synchronization, we make sure that time drift will arrive at this value every t_{synch}. The key is to choose a value of t_{synch}so that the effect of time drift is not noticeable in further experiments. For further experiments $\boldsymbol{t_{\text{synch}}=}$ **2 s** was chosen. The reason for that is the fact that all the time intervals for

[1] Note that in described measurement only one ETx at the time took part in the experiment.

which we tested beaconing-based and probing-based protocols (namely $t^{\mathrm{ERx}}_{\mathrm{Ping}}$, $t^{\mathrm{ERx}}_{\mathrm{PwrProbe}}$, $t^{\mathrm{ETx}}_{\mathrm{CrgReq}}$, $t^{\mathrm{ETx}}_{\mathrm{PwrProbe}}$ and $t^{\mathrm{ERx}}_{\mathrm{WaitForPwr}}$) are larger than t_{synch}value. This way we ensure that every time any of the control protocol events happen, the time is correctly synchronized (time synchronization happens more often than the most frequent events in the WPTN system).

Chapter 5

Simulation and measurements

5.1 Measurement scenario

Several ETxs and one ERx emulator were placed on a cardboard boxes 50 cm tall—allowing for equal positioning in the vertical plane. Four ETxs (ETx1, ETx2, ETx3 and ETx4) were placed at the edges of a 1.5 m × 3.5 m rectangular plane. The angles of the front of the antennas were regulated and initially aligned with the diagonals of the measurement plane, with their center axis unchanged during the entire experiment. Conversely, the ERx emulator was allowed to be placed in ten different positions separated in vertical and horizontal axes by 1 m and 0.5 m, respectively. The front of the ERx emulator panel antenna was always parallel to one of the axes. A schematic representation of the ETxs and ERx emulator positions is presented in Fig. 5.1. The measurement setup has been built inside the master student office of the TU Delft Embedded Systems Lab, see Fig. 5.2, with minimal movement of humans during the experiment.

Within this setup, the experiment simulated the random appearance and disappearance of the ERx in a controlled and repeatable fashion. The experiment was started by placing the ERx emulator at position '1', see Fig. 5.1, and initializing a measurement manually. The experiment was initialized by turning on or pressing the reset button of each device in the network, in the following order[1]: ETx1, ETx2, ETx3, ETx4, ERx. From that moment the ERx emulator advertises itself to the WPTN and starts collecting measurements. The ERx emulator is placed at this position for a random time chosen uniformly between 40 s and 44 s. This behavior is introduced to simulate random appearances and arrivals of the ERx emulator within one

[1] Order of initializing the devices has no influence on the experiments. It was done to ease the debugging process at the stage of developing the software and designing the experiment

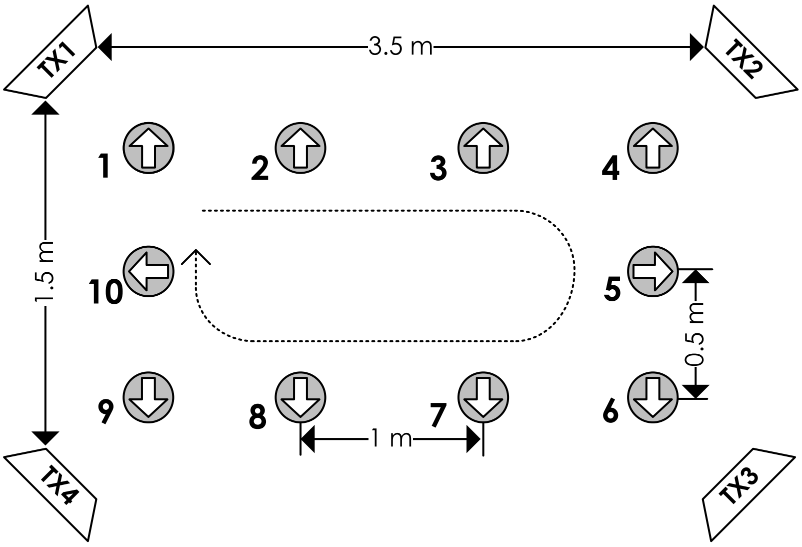


Figure 5.1: Experiment setup

time period of sending $\mathtt{REQ}_{\mathtt{CRG}}$ messages. After that time it signals the end of the single measurement through a buzzer, see Fig. 3.5(b). Consequently, the protocol execution is paused for 15 s allowing the experiment operator to move the ERx emulator to a second measuring position. One round of data collection is finished when the ERx emulator reaches position '10', with the movement pattern depicted in Fig. 5.1. Each round of movements has been repeated five times for statistical significance. The duration of a single experiment was ten minutes. Therefore, results presented in the following section are based on approximately nine hours of constantly running measurements.

5.2 WPTN Performance Metrics

5.2.1 Choice of performance measures

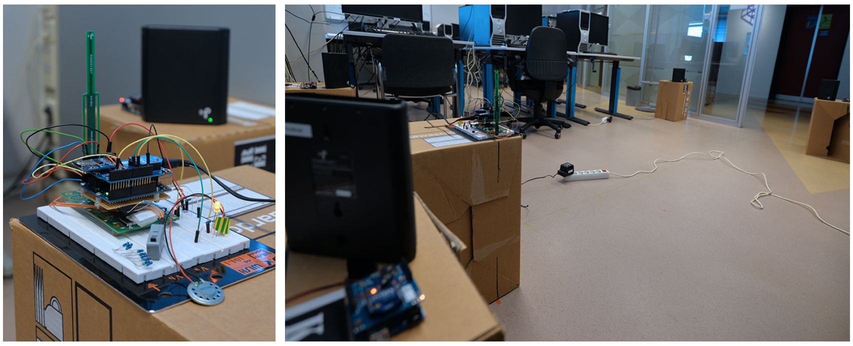
We consider three WPTN performance metrics, that will be described in the following sections. Those metrics relate to one instance of an experiment. For a different experiment (e.g. with a different ERx movement scenario) those results would be different. It is only possible to compare results for different sets of protocol parameters and the same experiment procedure.

Harvested Energy

Energy that has been harvested during the whole experiment - $\mathrm{E}_{\mathrm{h}}^{\mathrm{Rx}}$.

Efficiency

Efficiency is the ratio between energy harvested by the ERx and energy



(a) Energy receiver (b) Experiment photograph

Figure 5.2: WTPN experiment photograph: (a) ERx, (b) photograph of the entire setup in the laboratory setting.

consumed by all ETxs: $\eta = \frac{\mathrm{E_h^{Rx}}}{\sum_{i=1}^{N}\mathrm{E_i^{Tx}}}$, where $\mathrm{E_i^{Tx}}$ - energy consumed by ETx i.

Energy Consumption

We use a simplified model of energy consumption. We assume that a passive wakeup radio is used to wake up the receiver from off state to start communicating with the chargers. The receiver consumes power only for transmitting and receiving. Suppose N_t and N_r are the number of packets that the receiver transmits and receives in a round of experiment. The simplified power consumption model for a round of experiment is as

$$E_c = E_{T_x} + E_{R_x} = N_t \frac{S_p}{R_d}(UI_{T_x}) + N_r \frac{S_p}{R_d}(UI_{R_x}). \tag{5.1}$$

The explanation of variables used in the model can be found in Table 5.1.

Accuracy

Accuracy is divided in three sub measures: receiver accuracy, transmitter accuracy during charging, and transmitter accuracy during periods where efficient charging is not possible. Receiver and transmitter accuracy are analyzed separately and combined into one metric called charging accuracy.

5.2.2 Accuracy

Harvested energy and efficiency describe the wireless power transfer characteristic. This is dependent, not only on the specific protocol, but also hardware, environment and experiment scenario. We were trying to come

Table 5.1: Variables in the hardware experiment: Arduino Uno and Xbee

Symbol	Description	Value
U	Working voltage	3.3 V
I_{T_x}	Transmitting current	35 mA
I_{R_x}	Receiving current	50 mA
$I_{S,x}$	Sleep current (Xbee)	0.9 mA
$I_{S,a}$	Sleep current (Atmel)	0.01 mA
R_d	Data rate	9600 bytes/s
S_p	Packet size	120 bytes
T_a	Appear time at a position	40 s
T_d	Disappear time at a position	15 s
T_s	Voltage sample period	0.1 s

(a) Arduino Uno values taken from [45, Table 29-7]

(b) Xbee values taken from [46]

up with a measure that would describe the accuracy of behavior of the ETx devices in the network without describing how much power they actually deliver. In order to define such a metric, we need to define types of events that can happen in the system.

Let us assume a certain environment, where we have one ERx device and N ETx devices. The ERx can be within the charging range of WPTN or outside of the charging range. If the ERx is within the charging range of the WPTN, it means that **at least one** ETx can charge this ERx successfully. Conversely, if the ETx is outside of the charging range of the WPTN, it means there is no ETx that could charge this ERx at this moment. We consider an ERx being charged successfully when it can harvest energy, that results with the voltage equal or higher than $t^{\mathrm{ERx}}_{\mathrm{VoltageTh}}$ to appear on the load of the ERx. While the ERx is in charging range of the WPTN two situations can happen: a situation when ERx is indeed charged (correct behavior of the protocol) and a situation when ERx is not charged (incorrect situation, called *missed charging*).

On top of those relationships we have a behavior of ETx devices. Each ETx device can be in the state of transmitting energy at any point of time (both when the ERx is in the charging range of the WPTN and when it is outside of this range). It is important to point out that apart from two trivial situations, where the ETx is transmitting energy and the ERx is receiving energy successfully (correct behavior of the protocol) and the ETx is transmitting energy while the ERx is outside of charging range of WPTN (incorrect behavior, called below *false alarm*), there is one situation that is not trivial, and that is the ETx transmitting energy and the ERx being within the charging range, but not receiving energy successfully. This

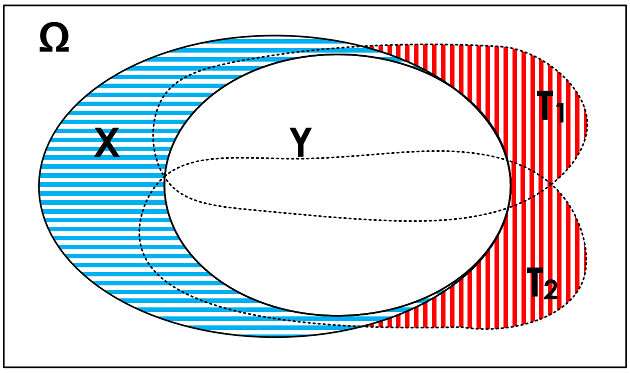


Figure 5.3: Illustration of states of an example WPTN system and relationships between those states in the form of an Euler diagram: Ω - space of all events; X - ERx was in charging range of at least one ETx; Y - ERx was charged by at least one ETx; T1 - ETx1 was transmitting energy; T2 - ETx2 was transmitting energy. *Missed charging* is marked with blue horizontal lines and *false alarm* with red vertical lines. Events for only two transmitters are depicted for clarity. In general, there may be more than two transmitters in the system. Note that for any number of the ETx devices $X \setminus Y \cap \bigcap_{i=1}^{N} T_i = \emptyset$ holds.

situation means that a wrong ETx device was chosen by the protocol to transmit energy. We also do not consider a situation when the ERx is outside of the WPTN charging range, but still receiving energy (e.g. from external sources). All those events are depicted in Figure 5.3:

- Ω — state of all events of the WPT Network;
- X — ERx was within charging range of the WPTN;
- Y — ERx was being charged by the WPTN;
- T_1 — ETx1 was transmitting energy;
- T_2 — ETx2 was transmitting energy.

There are a couple of events of interest for further analysis, namely:

- $X' \cap \bigcup_{i=1}^{N} T_i$ — *false alarm*;
- $X \cap \bigcup_{i=1}^{N} T_i$ — a complement of a *false alarm* set of events, correct behavior of the network when ERx is within charging range of WPTN;
- $X \setminus Y$ — *missed charging.*

Moreover, it should be noted that Figure 5.3 shows an example where only two transmitters are presented in the WPTN. Note that the set $X \setminus$

$Y \cap T_1 \cap T_2 = \emptyset$. In general, the set $X \setminus Y \cap \bigcap_{i=1}^{N} T_i = \emptyset$ due to the fact that if the ERx is within the charging range of WPTN, at least one of the ETx devices should be able to charge it. Thus, it is **not possible** that all the ETx devices are transmitting energy ($\bigcap_{i=1}^{N} T_i$) and the ERx is within charging range of the WPTN but **not** receiving power ($X \setminus Y$).

Receiver Accuracy

Receiver accuracy could be defined as the accuracy of charging the ERx device. In that measure we are not interested in the behavior of the individual ETx devices. The only interest is whether ERx device was charged by the WPTN when it should have been. The metric is presented in Equation 5.2.

$$A_{ERx} = \frac{|Y|}{|X|}, \text{see missed charging} \tag{5.2}$$

Notice that $A_{ERx} = 0$ when $|Y| = 0$, which means the ERx was not at all charged when it was within charging range of the WPTN. $A_{ERx} = 1$ when behavior of the network was fully correct — the ERx was charged at all times it was within charging range of WPTN and no *missed charging* occurred.

Transmitter Accuracy

Receiver accuracy (A_{ERx}) does not take into consideration the behavior of ETx devices. However, in reality the behavior of ETx devices is crucial for the efficiency of the WPTN. That is the reason we consider the additional metric of transmitter accuracy. The metric is presented in Equation 5.3.

$$A_{ETx} = \frac{|Y|}{\sum_{i=1}^{N} |X \cap T_i|} \tag{5.3}$$

This accuracy metric assesses the correctness of the behavior of ETx devices while the ERx device could have been receiving energy (the ERx device was within charging range of the WPTN).

The first effect this measure accommodates is the effect of **wrong ETx devices** being chosen to charge the ERx device while it was within charging range of WPTN. An example of such a situation can be a situation when the ETx1 is on while the ERx is within the charging range of the WPTN, but the ERx device is not receiving energy (it means that some other ETx should have been turned on at that time). The more of such behavior exists in the network, the lower the A_{ETx}. This effect will be lower as the value of A_{ERx} approaches 1 (the value of $|Y|$ approaches the value of $|X|$). We make further analysis assuming $|X| = |Y|$.

Another effect this measure includes is the effect of multiple ETx devices charging the same ERx at the same time. The desired behavior that optimizes received energy and efficiency is when only one ETx is charging the given ERx at a time. If we assume, that the ERx was charged during all times it was within charging range of the WPTN ($|X| = |Y|$), then $A_{ETx} = 1$ when for the whole period when the ERx was being charged only by one ETx. The lowest value of A_{ETx} is $1/N$, because in the worst case scenario all N ETx devices were transmitting energy when the ERx was receiving it. If $A_{ETx} = x$ it means that performance of the ETx devices was 1/x times worse than it could have been if only one ETx device at the time would charge ERx, thus it means that 1/x times more ETx devices were charging the ERx device at the same time.

Charging Accuracy

We consolidate two above metrics in one metric, defined in Equation 5.4.

$$A = A_{ETx} A_{ERx} \tag{5.4}$$

In this metric, A_{ETx} can be seen as a normalization factor for more general, receiver-level accuracy measure — A_{ERx}. The above metric has the following characteristic:

- If $A = 1$, the behavior of the system while the ERx was within charging range of the WPTN was fully correct — the ERx was charged by one transmitter at the time ($A_{ETx} = 1$) and the ERx was charged for the whole time it was within charging range of the WPTN ($A_{ERx} = 1$);

- If $A = 0$, the behavior of the system while the ERx was within charging range of the WPTN was completely incorrect — the ERx was not charged at all when it was within charging range of the WPTN ($A_{ERx} = 0$), in this case $A_{ETx} = 0$ too, since $|Y| = 0$;

- If $A = x$, $x \in (0, 1)$, the behavior of the system while the ERx was within charging range of the WPTN was partially correct — either the ERx was charged fraction x of all possible moments it could have been charged ($0 < A_{ERx} < 1; A_{ETx} = 1$), ERx was being charged by 1/x ETx devices more than it should have been ($0 < A_{ETx} < 1; A_{ERx} = 1$), or a combination of those two factors ($0 < A_{ETx} < 1; 0 < A_{ERx} < 1$).

No-charging Accuracy

Both receiver and transmitter accuracies (A_{ERx} and A_{ETx} respectively) as well as the consolidated accuracy measure A take into consideration situations where the ERx was within charging range of the WPTN (set of events symbolized by X). What is however as crucial to consider as this set of events is a complementary set of events $X' = \Omega \setminus X$. This set of events represents situations when the ERx was outside of the charging range of the WPTN. We are interested in seeing how ETx devices behave during that time. The desired situation takes place when no ETx is transmitting energy while the ERx is outside of the charging range of the WPTN ($\sum_{i=1}^{N} |X' \cap T_i| = 0$). However, due to delays in the control pane of the protocol this will not always be the case. If $\sum_{i=1}^{N} |X' \cap T_i| \neq 0$, then we have a *false alarm* situation, when an ETx transmits energy even though there is no ERx it could charge in the network. The measure is presented in Equation 5.5.

$$A_{NoCharging} = \frac{1}{N} \frac{\sum_{i=1}^{N} |X' \cap T_i|}{|X'|}, \text{see false alarm} \tag{5.5}$$

5.3 Results

5.3.1 ERx Time to Charge

To verify the theoretical analysis of time to charge in both protocols, we have conducted an experiment, where we have placed the ERx less than 50 cm to each of the ETx devices (to ensure ERx is within charging range of all ETxs). To emulate the ERx being within charging range of a given ERx, we would connect or disconnect the Powercast device from the Arduino microcontroller. For each value of K from $K = 1$ (one ETx connected) to $K = 4$ (four ETxs connected) we have performed an experiment where ERx appears randomly in the network 50 times. Afterwards, we measured the time it takes from appearing in the network to being charged. A CDF of those experiments are presented in Figures 5.4 and 5.5. In this figure experimental results (solid lines) are compared against theoretical results (dashed lines). For Figure 5.4 three CDFs are presented: the theoretical one corresponding to the equation (3.2) (used to compare with the probing-based protocol results); uniform distribution of a random variable $T^{\mathrm{B}}_{\mathrm{start}}$ corresponding to the equation (3.1) (used to compare with experimental result); and the experimental result.

We see that the beaconing-based protocol is faster in reaching an ERx than the probing-based protocol, however with increasing K the time to charge for the probing-based protocol becomes very low as well (almost instant connection after approximately two seconds). For the beaconing-based protocol irrespective of number of ETx the time to charge stays constant. The discrepancy between experimental and numerical results is due to approximation of not taking into account the propagation and processing time. Nevertheless the analysis follows the trends of the experimental results.

For the case of the probing-based protocol, we can notice that changing the number of the ETx devices within charging range of the ERx affects the number of rounds the protocol will take to reach the successful round (it is reflected in the number of 'steps' in Figure 5.5). The experimental results match the theory in the number of rounds it takes to successfully start charging the ERx. However, there are some differences between specific values of experiments and theory, with experiment showing generally higher values of the time to charge. The reason for that is underestimation of the processing and the packet transmission time of the packet. As we can judge from Figure 5.4, the transmission and processing time of a single packet is approximately 0.5 s. The reason for that conclusion is the fact that the experimental curve is shifted by approximately 0.5 s compared to the theoretical uniform distribution. This means it takes an additional 0.5 s to send and process the packet. In the probing-based protocol we have three packets like that, which amounts to approximately 1.5 s. That is indeed the delay between theoretical values and experiment in most cases. The reason for the increased steepness of the experimental curve of the probing-based protocol compared to the beaconing-based protocol is the fact that there is much more data traffic and processing at the ERx (e.g. making decision of answering the $\texttt{REQ}_{\texttt{PWR}}$ packet, checking the Q_{TX} for addresses). All those initial tasks take approximately 3 s as we can see from Figure 5.5.

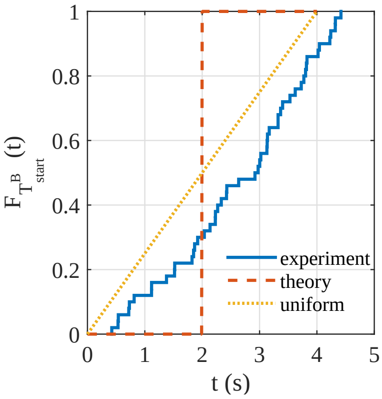
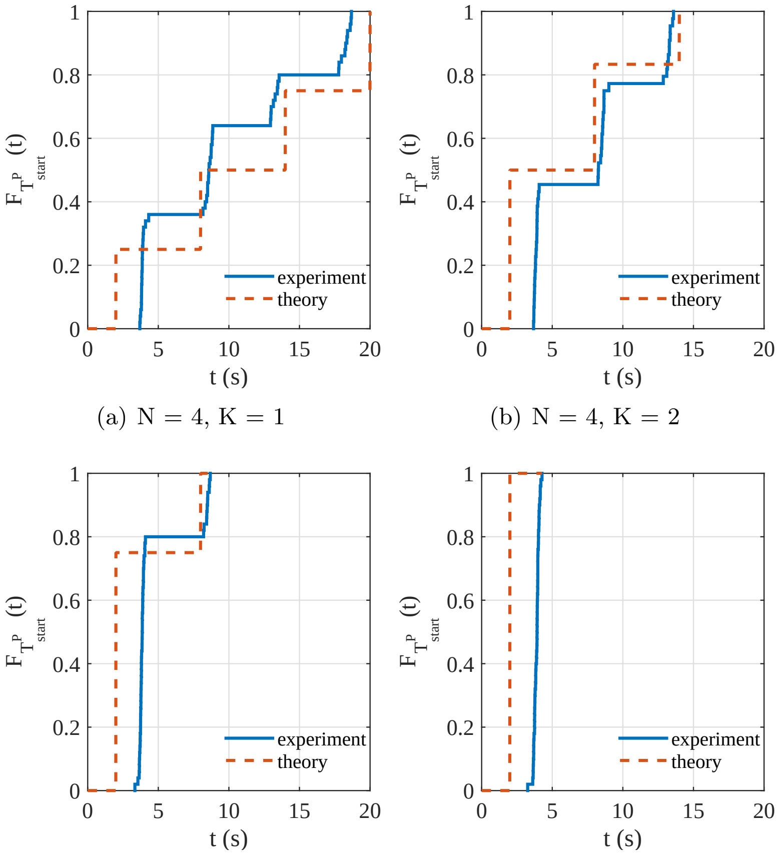


Figure 5.4: WTPN time to charge CDF - beaconing-based protocol, solid line–experiment, dashed line–analysis. N = 4, K = 1 scenario. Changing K has no effect on the result.

5.3.2 Reference measurement

In order to calculate the accuracy of the protocol, we needed to conduct a reference measurement. The goal of this measurement was to find out which of the 10 measurement positions are within the charging range of which ETx device. The scenario is the following:

- Only one ETx device is turned on for each measurement (from ETx1 to ETx4);



(a) N = 4, K = 1

(b) N = 4, K = 2

(c) N = 4, K = 3

(d) N = 4, K = 4

Figure 5.5: WTPN time to charge CDF - probing-based protocol, solid line–experiment, dashed line–analysis.

- Power received by the ERx from the given ETx device is measured in each of the ten measurement positions, see Figure 5.1;
- The ERx appears in the each position for 20 s and disappears for 10 s (the shortened time of the measurement does not influence the conclusion);
- For each measuring position the average received power is calculated out of all collected samples towards the given ETx;
- If the received power is on average higher than $t_{\text{VoltageTh}}^{\text{ERx}}$ at least from one ETx, then is it considered that in this position the ERx is within charging range of the WPT network;
- Otherwise (average received power lower than $t_{\text{VoltageTh}}^{\text{ERx}}$), the ERx is considered to be outside of the charging range of a WPTN.

The output of the measurement is a binary vector of length 10, where the first position corresponds to the first measurement position and the last position in the vector corresponds to the last measurement position in the experiment (see Figure 5.1). If the value of the vector in a given position is ‘1’, it means that in this measurement position the ERx is within the charging range of the network; if this value is ‘0’, it means the ERx is outside of the charging range of the network. This vector was used to calculate accuracy values of the protocol.

5.3.3 Line-of-sight scenario

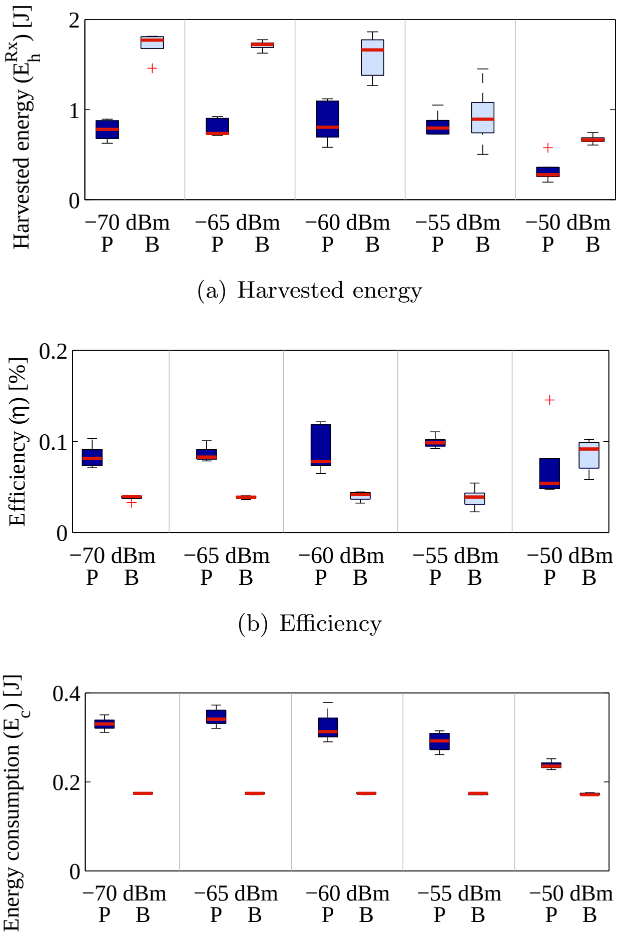
For each measure of performance (harvested energy, efficiency, energy consumption and accuracy) we have performed the experiment for five different communication threshold values ($t^{\text{ETx}}_{\text{CommTh}}$). The value of $t^{\text{ETx}}_{\text{CommTh}}$ can be interpreted as a parameter of the protocol, changed in software, or as a value corresponding to the physical maximum range of communication device. For a given threshold $t^{\text{ETx}}_{\text{CommTh}}$ all the communication below this threshold was ignored in software, but the same would happen if the transmit or receive power would be lower due to physical properties of the communication devices.

We have performed two sets of experiments - for the beaconing-based protocol (marked with the letter ‘B’ on the figures) and the probing-based protocol (marked with the letter ‘P’).

The result is presented in Fig. 5.6, Fig. 5.7 and Fig. 5.8. Results were plotted using MATLAB’s boxplot function.

Figure 5.6(a) shows the energy harvested for different values of $t^{\text{ETx}}_{\text{CommTh}}$ and both the beaconing-based protocol and the probing-based protocol. For every value of the threshold the energy harvested is higher for the beaconing-based protocol. The reason for that is that in the beaconing-based protocol there are less restrictions connected with charging. On average more ETx devices will charge the ERx device (sometimes multiple ETx devices at the same time), which — in this particular scenario — results in more energy being harvested (this does not always have to be the case, see Section 3.3).

In order to make a complete comparison of the beaconing-based and probing-based protocols, we also need to evaluate the efficiency, energy consumption and accuracy of the protocols. The efficiency of both protocols is presented on Figure 5.6(b). In this case we see that for each value of $t^{\text{ETx}}_{\text{CommTh}}$ the efficiency of the probing-based protocol stays on a similar level, while the efficiency of the beaconing-based protocols starts to increase for higher values of $t^{\text{ETx}}_{\text{CommTh}}$ (in our experiment the efficiency of the beaconing-based protocol starts to increase for -50 dBm). The reason for that is that as we raise the $t^{\text{ETx}}_{\text{CommTh}}$, we reach the situation where the communication threshold closer matches situations in which the ERx is in charging range. When we increase the $t^{\text{ETx}}_{\text{CommTh}}$ in the beaconing-based protocol, we avoid



(a) Harvested energy

(b) Efficiency

(c) Communication cost

Figure 5.6: WTPN experiment results for the two proposed WPTN charge control protocols and various reception threshold parameters: (a) harvested energy, (b) charge efficiency, and (c) cost of communication for ERx; B—Beaconing, P—Probing.

the situation where ETx devices are charging the ERx device, even though the ERx device does not receive any power. That is the reason for the efficiency increase. However, even though the efficiency increases, increasing the $t_{\mathrm{CommTh}}^{\mathrm{ETx}}$ also decreases the harvested energy.

Figure 5.6(c) shows the energy consumption in case of the beaconing-based and probing-based protocols. The energy consumption is related to the number of messages that are being sent and received in the protocol.

The execution of the probing-based protocol results in two or three messages being sent before reaching the Charging Phase, see Figure 3.11, compared to one message in case of the beaconing-based probocol, see Figure 3.9.

In case of the beaconing-based protocol, for both the optimistic and the pessimistic scenario it is only a $\texttt{REQ}_{\texttt{CRG}}$ message. For the probing-based protocol it is a $\texttt{REQ}_{\texttt{CRG}}$ message, followed by a $\texttt{REP}_{\texttt{PWR}}$ message and finally an unsolicited $\texttt{REP}_{\texttt{PWR}}$ in case of the optimistic scenario. That is the reason why the communication cost of the probing-based protocol is approximately two–three times higher than the beaconing-based protocol for low values of $t^{\text{ETx}}_{\text{CommTh}}$. Another reason for the higher communication cost for low values of $t^{\text{ETx}}_{\text{CommTh}}$ is the fact that in that case there is a higher probability that the ERx will take multiple rounds of trying to associate with ETx devices unsuccessfully to finally associate with an ETx that can charge it successfully. For the higher values of $t^{\text{ETx}}_{\text{CommTh}}$ the communication cost of the probing-based protocol becomes lower because for these values of $t^{\text{ETx}}_{\text{CommTh}}$ the ERx is on average within charging range of fewer ETx devices, so there is less unsuccessful initializations of the probing-based protocols (the first ETx with which ERx tries to associate is the only one it can associate with).

The last measures to consider comparing the beaconing-based and probing-based protocols are the accuracy measures. As discussed in Section 5.2.2, we define four measures: A_{ERx}, A_{ETx}, A (which is the combination of previous two accuracies) and $A_{NoCharging}$. The results of those measures are shown in Figure 5.7 and 5.8.

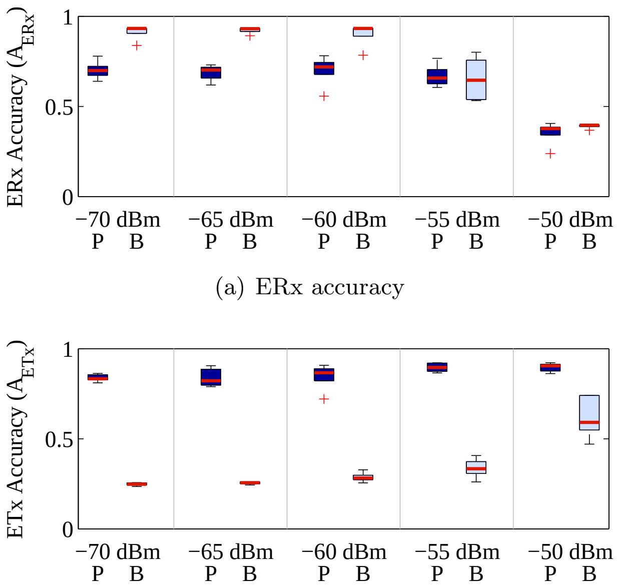
Figure 5.7(a) shows the ERx Accuracy (A_{ERx}). This value says if the ERx device was charged when it was in charging range of WPTN. Since the beaconing-based protocol takes into consideration only the communication layer threshold ($t^{\text{ETx}}_{\text{CommTh}}$), it will attempt to charge the ERx more often than the probing-based protocol, that also takes into consideration the harvested energy ($t^{\text{ERx}}_{\text{VoltageTh}}$). Due to that fact, it is understandable that for most values of $t^{\text{ETx}}_{\text{CommTh}}$, the beaconing-based protocol has a higher value of A_{ERx}. This however, comes with a cost of higher energy consumption (see the lower efficiency of the beaconing-based protocol in Figure 5.6(b)). The trend of A_{ERx} closely resembles the one of $\text{E}^{\text{Rx}}_{\text{h}}$, see Figure 5.6(a). This is due to the fact that the same mechanism is visible in both measures. The advantage of A_{ERx} is the fact that it can be used to compare different designs or be used in different experiments (no need for exact replication of an experiment as required for $\text{E}^{\text{Rx}}_{\text{h}}$).

The beaconing-based protocol allows multiple ETx devices charging the ERx and ETx devices charging the ERx even if there is not sufficient power being delivered (the ERx is outside of the charging range of the WPTN). This behavior is depicted using the ETx Accuracy — A_{ETx}, see Figure 5.7(b). The ETx Accuracy says how precise was the behavior of the ETx devices while the ERx was harvesting energy (while model behavior being the ERx charged by one ETx at the time). Here we can see that for all values of $t^{\text{ETx}}_{\text{CommTh}}$, A_{ETx} is high and constant for the probing-based protocol. For the beaconing-based protocol it is very low (around 0.1, compared with 0.9

for probing-based), and it increases with a decrease of $t^{\mathrm{ETx}}_{\mathrm{CommTh}}$. This is connected with the fact that as we increase $t^{\mathrm{ETx}}_{\mathrm{CommTh}}$, we are closer to the scenario where the communication threshold ($t^{\mathrm{ETx}}_{\mathrm{CommTh}}$) and power threshold ($t^{\mathrm{ERx}}_{\mathrm{VoltageTh}}$) mark the same area. That is also the reason why for these values of $t^{\mathrm{ETx}}_{\mathrm{CommTh}}$ the beaconing-based protocol becomes more efficient, see again Figure 5.6(a).

Figure 5.8(a) shows the overall Charging Accuracy (A), which is simply a multiplication of ERx Accuracy and ETx Accuracy. This can be considered as an overall protocol accuracy during the time the ERx was within charging range of the WPTN. This value is closely related to the value of efficiency — η — see Figure 5.6(b). The reason for that is that the Charging Accuracy is the measure that tells us the effect of the Receiver Accuracy combined with the Transmitter Accuracy. It takes into consideration actual transmitted to received energy as well as a behavior of ETx devices while the ERx could be charged. As we can see in Figure 5.8(a), even though the ERx Accuracy is higher for the beaconing-based protocol, if we also take into consideration the ETx Accuracy, the overall result is a Charging Accuracy being approximately three times higher for the probing-based protocol. For high values of $t^{\mathrm{ETx}}_{\mathrm{CommTh}}$ probing-based protocol becomes less accurate. The reason for that is that for this value of $t^{\mathrm{ETx}}_{\mathrm{CommTh}}$ the behavior of the protocol is not efficient in general — the communication threshold is too high for any successful message exchange to happen, and thus most of the charging opportunities are lost. This threshold results in a very low amount of harvested energy, see Figure 5.6(a).

The three accuracy measures we considered before are measures that inform us about the behavior of the WPTN when the ERx is within charging range of the network. As important as that is the behavior of the network when the ERx is outside of the charging range. This information is shown through the No-charging Accuracy ($A_{NoCharging}$), see Figure 5.8(b). This shows the behavior of the protocol when the ERx was outside of the charging range of the WPTN (so could not be efficiently charged). The beaconing-based protocol is in that case very inaccurate. This happens due to the fact that there is no feedback loop regarding the harvested energy in this protocol. Even though the ERx is within communication range of the ETx devices, they cannot successfully transmit power. For the probing-based protocol this situation is detected and the transmitters stop transmitting power. For the beaconing-based protocol this is not the case, and the transmitters inefficiently transmit power. The result is a low accuracy (and low efficiency, see again Figure 5.6(b)). As in the case of ETx Accuracy, for increasing values of $t^{\mathrm{ETx}}_{\mathrm{CommTh}}$, the beaconing-based protocol becomes more accurate, because the communication range and charging range become of similar size.



(a) ERx accuracy

(b) ETx accuracy

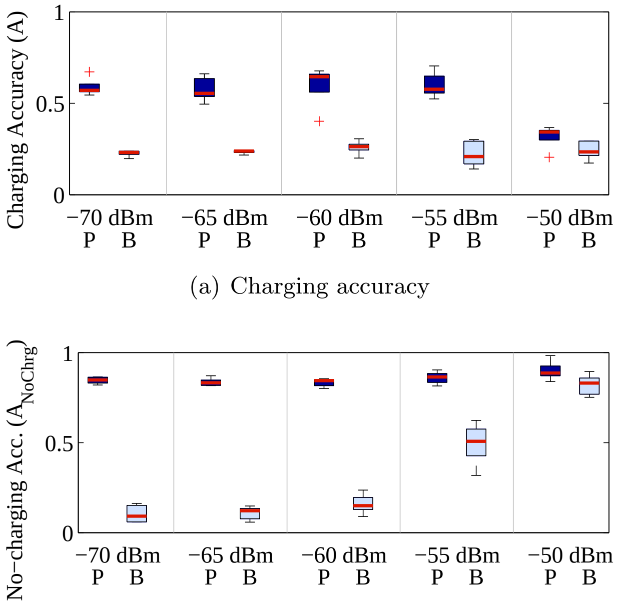
Figure 5.7: WTPN experiment results: (a) ERx accuracy, (b) ETx accuracy.

5.3.4 Non-line of sight scenario

We adapted the original setup by changing reverting the positions of ETx ‘1’ and ‘3’ by 180 degrees. With this we emulated the situation when those ETxs were seen by the control plane of ERx, while the rectified energy at the ERx position was low (the ERx was more often outside of the charging range of the WPTN, while still being within communication range). The result of the experiment is presented in Figures 5.9 and 5.10. Results from this scenario are depicted as *back*. They are compared against results from the previous section, which are marked as *normal*. The experiment was done for $t^{\text{ETx}}_{\text{CommTh}}$ of -70 dBm.

Figure 5.9 shows the results of $\text{E}^{\text{Rx}}_{\text{h}}$, η and E^{c} . We can see that the harvested energy in case of the beaconing-based protocol is almost two times lower — a result of the two ETx devices turned by 180 degrees, which means they are delivering very small amounts of energy to the ERx. In case of the probing-based protocol the behavior of the protocol is slightly better — we could harvest more energy. Moreover, the variance of the values of $\text{E}^{\text{Rx}}_{\text{h}}$ is higher for the *normal* scenario. This is not very intuitive and needs further explanation.

If we consult Figure 5.1, we can see that in the *back* scenario ETx1 and ETx3 were turned back, so that they cannot efficiently deliver energy to the ERx in any of the positions. After conducting many measurements we



(a) Charging accuracy

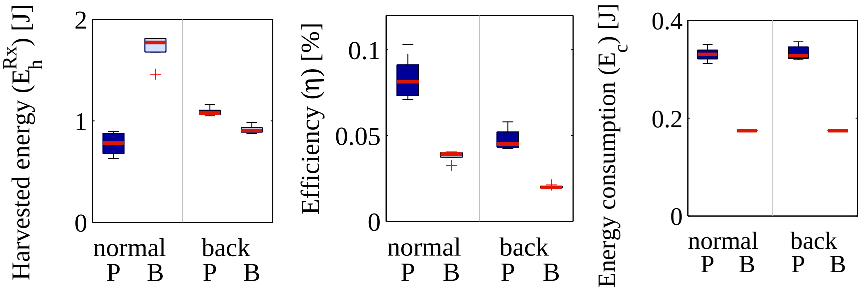
(b) No-charging accuracy

Figure 5.8: WTPN experiment results: (a) charging accuracy and (b) non-charging accuracy.

noticed however, that for the chosen value of $t^{\mathrm{ERx}}_{\mathrm{VoltageTh}}$ sometimes there are two or more ETx devices satisfying the condition of $\mathrm{E}^{\mathrm{Rx}}_{\mathrm{h}} \geq t^{\mathrm{ERx}}_{\mathrm{VoltageTh}}$, that is required for the probing-based protocol to go into the charging phase. As an example position '1' can be given — for the chosen $t^{\mathrm{ERx}}_{\mathrm{VoltageTh}}$ both ETx1 and ETx4 could satisfy the condition, but, ETx1 could deliver more than twice the energy delivered by ETx4. For the probing-based protocol however, the first, most simple idea of first-come, first-served (FCFS) was chosen. It means that when the ERx was in position '1' and if ETx4 contacted it first, the ERx would associate with ETx4, even though ETx1 could transmit more energy. Consequently, in case of the *normal* scenario for some experiments ETx1 would charge the ERx when in position '1', and in some other cases it would be ETx4. Moreover, if energy delivered from ETx4 is very close to the threshold $t^{\mathrm{ERx}}_{\mathrm{VoltageTh}}$, it can go below this threshold due to fast fading that is the consequence of multipath propagation. In such a case, the ERx drops ETx4 and tries to associate with another ETx. This takes time, that could be used to charge the ERx. In case of *back* scenario there are less situations when there are multiple ETx that can charge the ERx at the given position. In that scenario also less switching happens (e.g. in position '1' — if ERx decides to drop ETx4, it will not try to associate itself with another ETx because there is no other ETx device that could charge it successfully). These two reasons combined are causing the higher variance of E_h^{Rx} as well as the lower E_h^{Rx} in case of the *normal* scenario. It is important to remember though that this is a situation characteristic to our experiment scenario — this result may not happen in general scenario. However, it shows the complexity of the problem of choosing the best protocol and set of parameters for the most efficient WPTN.

For the scenario with the ETx devices turned by 180 degrees η decreases approximately twice, Figure 5.9(b). The reason for that in case of the beaconing-based protocol is just approximately two times less energy being harvested (two ETx devices are not charging ERx efficiently). For the probing-based protocol the reason is that for each time the ERx tries to associate with ETx1 or ETx3 (the ones that are turned back), they do not charge efficiently. However, the protocol reacts to that after a certain delay (power-probing phase) and energy transmitted by the ETx devices (but not delivered to ERx) is affecting this result.

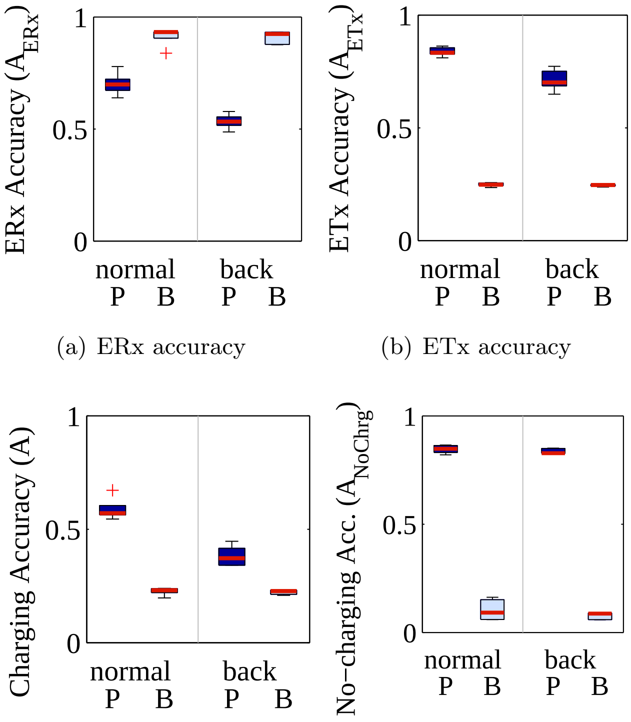
The communication cost is the same for both scenarios, see Figure 5.9(c). The reason is simple — there is the same number of ETx devices with which the ERx communicates.



(a) Harvested Energy (b) Efficiency (c) Communication Cost

Figure 5.9: WTPN experiment results: (a) harvested energy, (b) efficiency and (c) communication cost for the scenario with two transmitters turned the opposite direction (*back* scenario). Measurements done for -70 dBm communication threshold.

Figure 5.10 compares the accuracy results for the *normal* and the *back* scenarios. For all accuracy measures, the beaconing-based protocol results stay the same. The reason for that is the fact that there is no difference in protocol behavior based on energy delivered to the ERx, thus the behavior of the ETx devices in both scenarios was very similar. There is a difference for both ERx Accuracy and ETx Accuracy for the probing-based protocol. The accuracy of ERx Accuracy is lower due to the fact that in the *back* scenario it is more probable that the protocol ends up in unsuccessful power probing phase in comparison with the *normal* scenario. In case of ETx Accuracy the reason is the fact that sometimes the ETx that delivers energy to the ERx will be dropped by the ERx because the energy harvested by the ERx will be (even for a short period of time) lower than $t^{\text{ERx}}_{\text{VoltageTh}}$. A consequence is the Charging Accuracy (Figure 5.10(c)) being lower in the *back* scenario. For No-charging Accuracy, the value is similar for the probing-based protocol. This means that the behavior of the protocol in situations when no energy can be delivered to the ERx is still the same (and much better than in the beaconing-based protocol), see Figure 5.10(d).



(a) ERx accuracy

(b) ETx accuracy

(c) Charging accuracy

(d) No-charging accuracy

Figure 5.10: WTPN experiment results: (a) ERx accuracy, (b) ETx accuracy, (c) charging accuracy and (d) non-charging accuracy for the scenario with two transmitters turned the opposite direction (*back* scenario). Measurements done for -70 dBm communication threshold.

5.3.5 Results summary

In this Subsection we summarize the results of all the measurements. We look more closely for measurements done for very low $t^{\text{ETx}}_{\text{CommTh}}$ (a situation where ERx has connectivity to all ETx devices in the network for our experiment) and inspect how the results change when we increase the threshold.

For low $t^{\text{ETx}}_{\text{CommTh}}$ **probing is more efficient** and accurate but it **harvests less energy** than beaconing and has **higher communication cost**. As the threshold increases, the efficiency of the beaconing-based protocol increases (due to the fact that $t^{\text{ETx}}_{\text{CommTh}}$ becomes better tuned to indicate the charging range of the WPTN). The communication cost of the probing protocol is lowered due to the fact that there are fewer ETx devices that take part in Power Probing phase.

The ERx Accuracy is better for the beaconing-based protocol for low $t^{\text{ETx}}_{\text{CommTh}}$. However, the ETx accuracy is much better in that case, resulting in overall Charging Accuracy being approximately three times higher for the probing-based protocol. Also for No-charging Accuracy the probing-based protocol is much better. As $t^{\text{ETx}}_{\text{CommTh}}$ increases, ETx Accuracy and No-charging accuracy of the beaconing-based protocol increases (due to the same reason as before — better match of $t^{\text{ETx}}_{\text{CommTh}}$ to charging range of WPTN).

Chapter 6

Conclusions and Future Work

6.1 Conclusions

Currently, wireless power transfer networks (WPTNs) are gaining more and more attention from both academia and industry. They are believed to have a potential to solve many problems of wireless sensor networks, and advanced monitoring applications (e.g. insect monitoring). Many universities and research institutes are working to make WPT techniques more efficient, and industry is bringing those to the market.

In this thesis we have considered a novel concept of wireless power transfer networks. Since this is a new concept, a lot of new techniques need to be defined, starting from how such network could operate, what kind of hardware could it use, to how to measure and evaluate its operations. We have addressed those problems by defining new WPTN protocols, creating a real-world proof-of-concept system to measure and evaluate different WPTN protocols, proposed experiments and measures to evaluate the performance of such protocols, and finally performed an evaluation of two protocols proposed in this thesis.

We have identified three elements that are necessary to build efficient wireless power transfer networks: a wireless power transfer layer, a communication layer and a controller. The controller controls the behavior of both the wireless power transfer layer and the communication layer to optimize the behavior of the WPTN. This behavior is called a WPTN protocol. There are two types of such protocols: beaconing-based (without feedback from the wireless power transfer layer) and probing-based (with feedback from the wireless power transfer layer). We have proposed one of each type and implemented it in an experimental hardware platform.

In order to evaluate the behavior of WPTNs and compare different protocols, five performance measures were proposed: harvested energy, efficiency, energy consumption, accuracy and expected time to charge. The reason to define multiple measures is the fact that the goals of a WPTN can be contradictory — on one hand it is minimizing the power consumption in the network; on the other — maximizing charging of energy receivers.

We have considered one specific design and implementation of a beaconing-based protocol type and one of a probing-based protocol type. Even though the analyzed protocols are one of many that could be defined, we can already draw some general conclusions regarding the beaconing and probing types. By definition, every beaconing-based protocol will not use feedback regarding power received at the ERx. This allows for the more minimalistic design that uses less resources. However, having less information makes the optimization problem more difficult. On the other hand there are probing-based protocols. Those protocols use the feedback loop in the form of information about the power received at the ERx. This will always consume at least the same amount of resources as the beaconing-based solution (e.g. in terms of exchanged messages), since in order to initialize probing, some kind of a beaconing scheme needs to exist in such a protocol too. Based on that rationale we identify a few key differences between those two types of protocols:

- the harvested energy is higher for a beaconing-based protocol, but the cost is decreased efficiency (compared to a probing-based protocol);

- the efficiency is higher for a probing-based protocol, but the energy consumption (caused by more control traffic) is also higher and the harvested energy (caused by delays connected with control layer performance, see Figure 5.5) is lower compared to a beaconing-based protocol;

It is important to note that for certain scenarios and environments a beaconing-based protocol turns out to be better, while for some other scenarios a probing-based protocol seems to be more beneficial, taking into consideration all the features of the protocol. There are two special cases that can be considered:

- a case with large distances between ETx and ERx devices, where the situation when the ERx device is within charging range of WPTN is **rare** — in that case a probing-based protocol appears to be much more efficient compared to a beaconing-based protocol;

- the opposite case, where distances between ETx and ERx devices are not so large, a situation where the ERx device is within charging range of the WPTN is **frequent** — in that case a probing-based protocol may seem to be an overkill, since a beaconing-based protocol achieves similar efficiency and harvested energy results with lower communication cost.

There is a number of lessons that can be learned from the conducted experiments:

1. For both beaconing-based and probing-based protocols, the charge request rate should be chosen appropriately in order to optimize power consumption and harvested energy in a WPTN. An example could be the charge request rate that adapts to the density of ERx devices or their mobility. If ERx devices move rapidly, the charge request rate should be higher to allow ETx devices to adapt to the changing situation; if ERxs do not move, the charge request rate can be lower, since the topology of the network does not change.
2. Similarly to the charge request rate, also the power probing rate should be optimized in probing-based protocols, based on similar principles.
3. The WPTN should choose the subset of ETx devices that are turned on at any given time in a way that optimizes power consumption and energy harvesting. It should take into account the non-linear efficiency of ERx devices RF-to-DC energy transfer, and locations of ERx and ETx devices. Measuring all the possible combinations of ETx devices that could charge a subset of ERx devices consumes a lot of time and energy, thus novel smart WPTN concepts should be developed.

6.2 Future Work

This thesis was the first step to analyze wireless power transfer networks. It proposed a basic classification of the types of protocols that could be implemented on top of such network and the way to measure their performance in both experiment-specific and generic way. Further work in that field can be divided in three aspects: hardware platform research, protocol research and application research.

The hardware platform used in our experiments was a basic proof-of-concept implementation. There is much room for improvement in investigating usage of off-the-shelve components to build real WPTN hardware (i.e. an ERx device that does not require a power supply connection) or using novel research platforms such as the Wireless Identification Sensing Platform (WISP) developed at the University of Washington [47]. Apart from trying to integrate different hardware platforms, research into optimizing efficiency of power conversion needs to be investigated.

In order to make a WPTN work efficiently, efficient WPTN protocols need to be developed on top of optimized hardware platforms. Thus, protocol research (which was the main contribution of this thesis) needs to be continued further. Specific follow-up tasks could include research to optimize protocol parameters for the specific environment, first statically and then dynamically. Furthermore, more generic research into different types of protocols could be conducted, and how different hardware platforms and protocols affect each other.

Last but not least, research into applications of wireless power transfer needs to be conducted in parallel to all other activities. Since the possibilities of WPTNs are not yet fully known, it is not possible to imagine all the possible applications of this novel technology. Research areas that can benefit the most from our research are those fields where battery life or providing power to multiple devices is a problem. These problems are common, for instance, in wireless sensor networks, where much research into harvesting energy is already being done.

Bibliography

[1] J. Maxwell and T. Torrance, *A Dynamical Theory of the Electromagnetic Field*, ser. The Torrance collection. Wipf & Stock Pub, 1996. [Online]. Available: http://books.google.pl/books?id=eI4pAQAAMAAJ

[2] T. Edison, “Thomas a,” Aug. 22 1882, uS Patent 263,142. [Online]. Available: http://www.google.com/patents/US263142

[3] N. Tesla, “Apparatus for transmitting electrical energy.” Dec. 1 1914, uS Patent 1,119,732. [Online]. Available: http://www.google.com/patents/US1119732

[4] N. Shinohara, *Wireless Power Transfer via Radiowaves*, ser. ISTE. Wiley, 2014. [Online]. Available: http://books.google.pl/books?id=TwegAgAAQBAJ

[5] W. C. Brown, “The History of Power Transmission by Radio Waves,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 32, no. 9, pp. 1230–1242, 1984. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1132833

[6] B. C, G. H, H. Neil, and W. C, “Microwave to dc converter,” Mar. 25 1969, uS Patent 3,434,678. [Online]. Available: http://www.google.com/patents/US3434678

[7] R. Company, *Experimental Airborne Microwave Supported Platform.* Griffiss Air Force Base, 1965. [Online]. Available: http://books.google.nl/books?id=VS2WGwAACAAJ

[8] A. Fisher, “Secret of perpetual flight? beam-power plane.” *Popular Science*, pp. 62–65, 106–107, jan 1988.

[9] M. Cardullo and W. Parks, “Transponder apparatus and system,” Jan. 23 1973, uS Patent 3,713,148. [Online]. Available: http://www.google.com/patents/US3713148

[10] A. R. Koelle, S. W. Depp, and R. Freyman, “Short-range radio-telemetry for electronic identification, using modulated rf backscatter,” *Proceedings of the IEEE*, vol. 63, no. 8, pp. 1260–1261, Aug 1975.

[11] (2014) Wireless Power Consortium Homepage. [Online]. Available: http://http://www.wirelesspowerconsortium.com/

[12] S. He, J. Chen, F. Jiang, D. K. Yau, G. Xing, and Y. Sun, “Energy provisioning in wireless rechargeable sensor networks,” *IEEE Trans. Mobile Comput.*, vol. 12, no. 10, pp. 1931–1942, Oct. 2013.

[13] S. J. Thomas, R. R. Harrison, A. Leonardo, and M. S. Reynolds, “A battery-free multichannel digital Neural/EMG telemetry system for flying insects,” *IEEE Trans. Biomed. Circuits Syst.*, vol. 6, no. 5, pp. 424–435, Oct. 2012.

[14] C. G. D. H. D. Kalp and W. Tauche, “Making wireless sensor networks truly wireless using RF power,” 2010. [Online]. Available: http://www.sensormgmt.com/Articles/Powered%20By%20FireFly60614.pdf

[15] (2014) WiTricity Homepage. [Online]. Available: http://www.witricity.com

[16] (2014) UBeam Homepage. [Online]. Available: http://www.ubeam.com

[17] (2014) Ossia Homepage. [Online]. Available: http://www.ossiainc.com

[18] (2014) Artemis Home Page. [Online]. Available: http://www.artemis.com

[19] (2014) Energous Homepage. [Online]. Available: http://energous.com

[20] (2014) Homepage. [Online]. Available: http://powerbyproxi.com

[21] A. Kurs, A. Karalis, R. Moffatt, J. D. Joannopoulos, P. Fisher, and M. Soljacic, “Wireless power transfer via strongly coupled magnetic resonances,” *Science*, vol. 317, no. 5834, pp. 83–86, July 2007. [Online]. Available: http://dx.doi.org/10.1126/science.1143254

[22] M. Roes, M. Hendrix, and J. Duarte, “Contactless energy transfer through air by means of ultrasound,” in *Proc. IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society*, Nov 2011, pp. 1238–1243.

[23] M. Roes, J. Duarte, M. Hendrix, and E. Lomonova, “Acoustic energy transfer: A review,” *IEEE Transactions on Industrial Electronics*, vol. 60, no. 1, pp. 242–248, Jan 2013.

[24] A. Denisov and E. Yeatman, "Ultrasonic vs. inductive power delivery for miniature biomedical implants," in *Body Sensor Networks (BSN), 2010 International Conference on*, June 2010, pp. 84–89.

[25] K. E. Koh, T. C. Beh, T. Imura, and Y. Hori, "Impedance matching and power division using impedance inverter for wireless power transfer via magnetic resonant coupling," *IEEE Transactions on Industry Applications*, vol. 50, no. 3, pp. 2061–2070, May 2014.

[26] C. Park, S. Lee, G.-H. Cho, and C. Rim, "Innovative 5-m-off-distance inductive power transfer systems with optimally shaped dipole coils," *IEEE Transactions on Power Electronics*, vol. 30, no. 2, pp. 817–827, Feb 2015.

[27] H. Ju and R. Zhang, "Throughput maximization for wireless powered communication networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 1, pp. 418–428, Jan. 2014.

[28] M. Z. J. Li and Y. Yang, "A framework of joint mobile energy replenishment and data gathering in wireless rechargeable sensor networks," *IEEE Trans. Mobile Comput.*, 2014, accepted for publication.

[29] S. Timotheou, I. Krikidis, G. Zheng, and B. Ottersten, "Beamforming with MISO interference channels with QoS and RF energy transfer," *IEEE Trans. Wireless Commun.*, vol. 13, no. 5, pp. 2646–2658, May 2014.

[30] M. Y. Naderi, P. Nintanavongsa, and K. R. Chowdhury, "RF-MAC: A medium access control protocol for re-chargeable sensor networks powered by wireless energy harvesting," *IEEE Trans. Wireless Commun.*, Apr. 2014, accepted for publication.

[31] X. Liu, P. Wang, D. Niyato, and Z. Han, "Resource allocation in wireless networks with RF energy harvesting and transfer," *IEEE Network*, May 22, 2014, accepted for publication. [Online]. Available: http://arxiv.org/abs/1405.5630

[32] G. Yang, C. K. Ho, and Y. L. Guan, "Dynamic resource allocation for multiple-antenna wireless power transfer," Nov. 17, 2013, submitted to IEEE Trans. Signal Processing. [Online]. Available: http://arxiv.org/abs/1311.4111

[33] I. Krikidis, "Simultaneous information and energy transfer in large-scale networks with/without relaying," *IEEE Trans. Commun.*, vol. 62, no. 3, pp. 900–912, Mar. 2014.

[34] C. Wang, J. Li, and Y. Yang, “NETWRAP: An NDN based real-time wireless recharging framework for wireless sensor network,” *IEEE Trans. Mobile Comput.*, vol. 13, no. 6, pp. 1283–1297, Jun. 2014.

[35] H. Dai, G. Chen, C. Wang, S. Wang, X. Wu, and F. Wu, “Quality of energy provisioning for wireless power transfer,” *IEEE Trans. Parallel Distrib. Syst.*, 2014, accepted for publications. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6762897

[36] P. P. Mercier and A. P. Chandrakasan, “A supply-rail-coupled eTextiles transciever for body-area networks,” *IEEE J. Solid-State Circuits*, vol. 46, no. 6, pp. 1284–1295, Jun. 2011.

[37] L. Liu, R. Zhang, and K.-C. Chua, “Wireless information and power transfer: A dynamic power splitting approach,” *IEEE Trans. Commun.*, vol. 61, no. 9, pp. 3990–4001, Sep. 2013.

[38] K. Huang and V. K. N. Lau, “Enabling wireless power transfer in cellular networks: Architecture, modelliing and deployment,” Jul. 24, 2012, submitted to IEEE Trans. Wireless Commun. [Online]. Available: http://arxiv.org/abs/1207.5640

[39] L. Xiang, J. L. K. Han, and G. Shi, “Fueling wireless networks perpetually: A case of multi-hop wireless power distribution,” in *Proc. IEEE PIRMC*, London, UK, Sep. 8–11, 2013.

[40] (2014) Powercast Corp. Homepage. [Online]. Available: http://www.powercastco.com

[41] (2014) Digi International Homepage. [Online]. Available: http://www.digi.com/xbee/

[42] (2014) Arduino Uno Board Homepage. [Online]. Available: http://arduino.cc/en/Main/arduinoBoardUno

[43] Q. Liu, P. Pawelczak, M. Golinski, and M. E. Warnier, “Charger energy-conserving wireless power transfer network,” 2015, published as a TU Delft technical report TUD-ES-2015-01.

[44] A. S. Tanenbaum, *Modern Operating Systems*, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2007.

[45] Atmel Corp. (2014) Atmel 8-bit microcontroller with 4/8/16/32 kbytes in-system programmable flash. [Online]. Available: http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf

[46] Digi International Inc., “Xbee 802.15.4 module specification,” 2015. [Online]. Available: http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/point-multipoint-rfmodules/xbee-series1-module#specs

[47] J. R. Smith, A. P. Sample, P. S. Powledge, S. Roy, and A. Mamishev, “A wirelessly-powered platform for sensing and computation,” in *UbiComp 2006: Ubiquitous Computing*. Springer, 2006, pp. 495–506.

Appendix A

Source code of WPTN implementations

Listing A.1: executeProtocol() function of PowercastReceiver

```
void PowercastReceiver::executeProtocol()
{
  boolean isPacketReceived = tryReceivePacket();

  if (isPacketReceived) {
    ProtocolPacket* packet = obtainPacket();

    if (powercastMeasurementLayer != NULL && packet->protocolType == PROTOCOL_TIMESYNCH) {
      powercastMeasurementLayer->processPacket(packet);
    }
    else if (powercastControlLayer != NULL && packet->protocolType == PROTOCOL_CONTROL) {
      powercastControlLayer->processPacket(packet);
    }
  }

  if (powercastMeasurementLayer != NULL) {
    powercastMeasurementLayer->doTimerActions();
  }

  if (powercastControlLayer != NULL) {
    powercastControlLayer->doTimerActions();
  }
}
```

Listing A.2: An example of definition of PowercastLayer child class

```
class PowercastTxControlLayer: public PowercastLayer
{
public:
  PowercastTxControlLayer();
  void setOutputPins(int pinTxOff, int pinTxOn);

  void processPacket(ProtocolPacket *xBeePacket);
  void doTimerActions();
  void processState(Event currentEvent, ProtocolPacket
      *xBeePacket);

  //EVENTS
  void eventReceivedPing();
  void eventTimeoutPing();

  // ACTIONS
  void actionTurnOn();
  void actionTurnOff();

  void doMeasurement();

private:
  void makeTxOn(bool status);

  unsigned long lastPingReceived;

  int pinTxOff;
  int pinTxOn;
};
```

Listing A.3: Beaconing-based protocol implementation

```
// WPIN beaconing based protocl processes states.
void PowercastRxControlLayer::processState(Event
    currentEvent, ProtocolPacket *xBeePacket)
{
  int nextState = currentState;
  switch (currentState)
  {
  case STATE_IDLE:
    if (currentEvent == EVENT_TIMEOUT_PING)
    {
      actionSendPing();
    }
    nextState = currentState;
    break;
  default:
    break;
  }
}

// Do actions according to different states in beaconing based protocol.
void PowercastRxControlLayer::doTimerActions()
{
  unsigned long currentTime = getCurrentTimeMiliseconds();
  if (currentTime - timerPing >= DELAY_PING) {
    eventTimeoutPing();
    timerPing = currentTime;
  }

  doMeasurement();
}
```

Listing A.4: Implementation of TxAdressesQueue

```
class TxAdressesQueue
  {
  public:
    TxAdressesQueue()
    {
      resetQueue();
    }

    void resetQueue()
    {
      for (int i = 0; i < SIZE_OF_TX_QUEUE; i++)
      {
        txAdressesQueue[i] = 0;
        timestampsForRemovalQueue[i] = 0;
      }

      delayRemoveLastProbeSender = 0;

      currentIndex = 0;
      oldestIndex = 0;
      previousIndex = 0;
    }

    void saveAddressToQueue(uint16_t address)
    {
      txAdressesQueue[currentIndex] = address;
      timestampsForRemovalQueue[currentIndex] = millis
          () + DELAY_REMOVE_LAST_PROBE_SENDER;
      previousIndex = currentIndex;
      currentIndex = (currentIndex + 1) %
          SIZE_OF_TX_QUEUE;
    }

    bool checkOldestTxForTimeout()
    {
      if (timestampsForRemovalQueue[oldestIndex] != 0
          && timestampsForRemovalQueue[oldestIndex] <=
          millis()) {
        return true;
      }

      return false;
    }

    void removeOldestFromQueue()
    {
      txAdressesQueue[oldestIndex] = 0;
      timestampsForRemovalQueue[oldestIndex] = 0;
      oldestIndex = (oldestIndex + 1) %
          SIZE_OF_TX_QUEUE;
    }

    bool checkForAddress(uint16_t address)
    {
      for (int i = 0; i < SIZE_OF_TX_QUEUE; i++)
      {
        if (txAdressesQueue[i] == address)
        {
          return true;
        }
      }

      return false;
    }
```

```

    bool checkForLastAddress(uint16_t address)
    {
      if (txAdressesQueue[previousIndex] == address)
      {
        return true;
      }
      else
      {
        return false;
      }
    }

    uint16_t getLastAddress()
    {
      return txAdressesQueue[previousIndex];
    }

  private:
    uint16_t txAdressesQueue[SIZE_OF_TX_QUEUE];
    unsigned long timestampsForRemovalQueue[SIZE_OF_TX_QUEUE];
    unsigned long delayRemoveLastProbeSender;
    byte currentIndex;
    byte oldestIndex;
    byte previousIndex;
  };
```