Energy Efficient Communication using Wake-Up Receivers

vorgelegt von

M.Sc.
Johannes Blobel
ORCID: 0000-0002-0591-2944

an der Fakultät IV - Elektrotechnik und Informatik
der Technischen Universität Berlin

zur Erlangung des akademischen Grades

DOKTOR DER INGENIEURWISSENSCHAFTEN
– Dr. Ing. –

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender:	**Prof. Dr.-Ing. Thomas Sikora**
Gutachter:	**Prof. Dr.-Ing. habil. Falko Dressler**
	Prof. Dr. Jiannong Cao
	Prof. Dr. Archan Misra
Tag der wissenschaftlichen Aussprache:	**23. März 2021**

Berlin 2021

Abstract

Wireless communication is the key enabler for the vision of an interconnected world. In many application scenarios like Wireless Sensor Networks (WSNs), Internet Of Things (IoT) or Industry 4.0 battery operated nodes monitor the environment and send the collected data over a wireless link. In order to ensure a long lifetime it is essential to design such systems as energy efficient as possible. In the past years a new technology called Wake-Up Receivers (WURs) has been proposed in the literature that promises ultra-low power communication without the drawbacks of traditional, duty cycling based protocols. In this thesis the WUR concept is studied towards its applicability in different scenarios: Wildlife animal tracking with small sensors, low power Wireless LAN (WLAN) for IoT devices and a more general communication system for devices with tight energy constraints. Also a new addressing scheme is proposed which allows a more flexible use of a WUR. The findings of this thesis show how the WUR concept can be used for energy efficient communication and how it can be integrated into existing systems with little additional overhead and in a way that is compatible with legacy devices. It offers very low energy consumption without degrading the performance or increasing delays in centralized and decentralized networks.

Kurzfassung

Drahtlose Kommunikation ist ein wichtiger Grundbaustein für die Vision einer immer besser vernetzten Welt. Das Vermessen der Umgebung und das Senden der gesammelten Daten über eine drahtlose Verbindung findet in vielen Bereichen, wie zum Beispiel in drahtlosen Sensornetzwerken, dem Internet Of Things (IoT) und in der Industrie 4.0 Verwendung. Um eine lange Laufzeit solcher Systeme zu gewährleisten, ist es von größter Bedeutung sie so energieeffizient wie möglich zu gestalten. In den letzten Jahren wurde zu diesem Zweck das Konzept der Weckrufempfänger (Wake-Up Receiver (WUR)) entwickelt, das sparsame Drahtloskommunikation ohne die Nachteile herkömmlicher zyklischer Protokolle verspricht. In dieser Arbeit wird das Weckrufempfängerkonzept auf seine Anwendbarkeit in unterschiedlichen Szenarien hin überprüft: Beobachtung von frei lebenden Fledermäusen, energieeffizientes Wireless LAN (WLAN) für IoT und ein allgemeines Kommunikationssystem für Geräte mit stark begrenzten Energieressourcen. Des weiteren wird ein neues Addressierungsschema entwickelt, das eine wesentlich flexiblere Nutzung von Weckrufempfängern ermöglicht. Die Ergebnisse dieser Arbeit zeigen, dass das Konzept von Weckrufempfängern für energieeffiziente Kommunikation genutzt werden kann und wie es sich in bestehende Systeme integrieren lässt und dabei kompatibel zu vorhandenen Geräten ist. Es vereint einen sehr geringen Energieverbrauch mit geringer Latenz und ohne eine Verschlechterung der Leistung sowohl in zentralisierten als auch in dezentralisierten Systemen.

Contents

Chapter 1

Introduction and Motivation

Wireless communication has changed our world dramatically in the past decades. It enables the exchange of information "over the air" which has transformed the way we communicate in an unprecedented manner. While the first step in connecting computers with each other like the ARPANET [1] was based on wired communication over telegraph landlines, it soon became clear that a wireless connection can be beneficial to also connect hard to reach locations like an island to a network. The first demonstration of a packet based wireless network was AlohaNet [2], a network that was intended to connect the different locations of the University of Hawaii with each other so that everyone was able to access the computer resources available at the main campus. The random access scheme that was used to share the common radio resources in this network was called ALOHA, a name which is still used today to describe the simplest form of Time Division Multiple Access (TDMA): Each node in the network just sends at a random time, hoping that no other node transmits at the same time. This first network was later extended with a satellite link to connect it to the ARPANET, which eventually led to the development of the Transmission Control Protocol and Internet Protocol, the foundation of today's internet.

The real revolution in wireless communication, however, came with the ongoing decrease in the size of electronics, facilitated by the development of the Integrated Circuit (IC). This reduction in size also led to a reduction in power required to operate a device, eventually enabling devices that can operate on battery. Stripped from wires that tie it to a fixed location, computers became truly mobile which dramatically changed the way we exchange information today. With smartphones widely available we can communicate wherever we are with the rest of the world.

A crucial requirement for such mobile applications, however, is that the communicating devices also have a mobile source of power. Even though battery technology has also improved significantly over the past years, eventually the energy stored in a battery will be depleted and the device needs to be recharged. Since the battery lifetime is an important factor for the usefulness of a mobile device, it is crucial to reduce the energy consumption as much as possible. This optimization goal can be achieved on many different levels, from the design of the computing device, the software running on the device and, of course, by reducing the energy required for wireless communication.

In this thesis I focus on the latter aspect: The energy consumption of the wireless radio chip. There are many different factors that influence the energy consumption (I will give a brief overview in this chapter) that can be optimized individually or using a cross-layer approach. Some solutions require to trade-off energy consumption and performance (in terms of throughput and delay) and some are optimized for a particular network topology. In this thesis I can only cover a subset of all the research problems and will focus on the level of medium access.

Even though I will explain important concepts and protocols used in this thesis, I assume that the reader has some basic knowledge about computer networks in general and wireless networks in particular. This includes digital modulation techniques and the characteristics of wireless signal propagation, fundamental communication concepts like ISO/OSI-Layers, packet based networks, etc.

1.1 Wireless Networks

Before introducing the concept of Wake-Up Receiver (WUR) based communication, I will give a broad overview on basic concepts of wireless networks that are the foundation of this thesis. Wireless communication is based on the propagation of electromagnetic waves[1] and the modulation of the wave's properties to convey information. As a wave is determined by its frequency (or wavelength), amplitude and phase, possible (digital) modulation techniques are Frequency-Shift Keying (FSK), Amplitude-Shift Keying (ASK), and Phase-Shift Keying (PSK) or a combination of them. In high data rate systems, like modern WLAN, the available bandwidth is divided into multiple, orthogonal carrier frequencies that are then modulated individually (Orthogonal Frequency-Division Multiplexing (OFDM)). Such a complex modulation

[1] We are here focusing on electromagnetic wave in the frequency spectrum between 300 MHz and 300 GHz, called microwaves.

technique requires complex transmitters and receivers (or short: transceivers) that consume a lot of energy. A very simple modulation like On-Off-Keying (OOK), a version of ASK with just two symbols, can only support low data rates, but the hardware to send and receive these signals is significantly simpler and needs much less energy. Between these two extremes there are a lot of different modulation techniques, all with their own strength and weaknesses. A more detailed discussion about these techniques is out of the scope of this introduction, the simple, but important observation, however, is this: *High data rate transceivers require more energy than low data rate transceivers.*

One of the main differences between wired and wireless communication is the broadcast nature of the wireless channel. A signal that is transmitted over a wired link is only received by the receiver at the other end of the line. Radio Frequency (RF) waves, that are emitted by an antenna, propagate *omnidirectional*[2] through space, which means that the signals from multiple transmitters could interfere with each other, if they send at the same time. It is therefore necessary to coordinate the channel/medium access across multiple devices in order to built a reliable and high performance wireless communication network. Multiple access can be achieved by dividing the wireless resources in time, frequency, code or space and in a variety of patterns (centralized/distributed, contention based/reservation based, etc.). The protocol that determines who is allowed to send when is called a Medium Access Control (MAC) protocol. It is responsible for multiplexing the common medium, provide flow control and detection and handling of failures. In order to coordinate the access between multiple nodes of a network, the nodes typically exchange some sort of control information over the wireless link, i.e. an Acknowledgement (ACK) or RTS/CTS (Request To Send/Clear To Send). The MAC has a huge influence on the performance of a wireless communication system and highly depends on the topology of the network. A centralized network,for example, that relies on an existing infrastructure, like cellular networks, can schedule the available (radio) resources in a centralized way, while Ad-hoc networks like Wireless Sensor Networks (WSNs) have to coordinate in a decentralized fashion. Again, a detailed overview on MAC protocols is beyond the scope of this thesis, but I will introduce some important examples later on. The important observation here is: *The coordination between multiple nodes through a MAC protocol is specific to the characteristics and topology of the network and has a significant influence on the performance and energy consumption of a wireless network.*

[2] Real antennas are not perfectly omnidirectional but show a typical radiation pattern. Also by using directed antennas or antenna arrays one can steer the radiated electromagnetic energy towards a more narrow direction. But this only reduces the amount of devices in a *collision domain*, a real one-to-one communication (unicast) is still not achievable

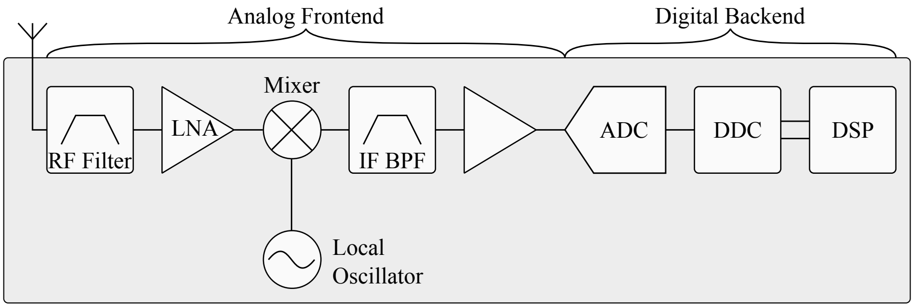


FIGURE 1.1 – Simplified architecture of a superheterodyne receiver.

There are many other aspects that affect the performance of a wireless network, which are not covered in this thesis. One important aspect is routing, meaning the forwarding of packets via multiple nodes (hops) from a source to a destination. Especially in decentralized networks, a well performing routing protocol is essential for the whole network as it ensures connectivity within the network. Routing protocols usually try to find routes through the network that minimize (delay, number of hops, energy) or maximize (throughput) a certain metric.

Energy Consumption in Wireless Networks

The transmission and reception of RF signals requires energy, which is a limited resource in mobile, battery operated systems. Therefore one important optimization goal for many wireless systems is the reduction of energy consumption. While there are approaches that reduce the energy consumption by adapting the *transmit* power like in [3], the main focus lies on the *energy consumption of the receiver*. The reason for that is, that even though the transmission usually requires more energy than reception[3], the transmitter only has to be enabled very briefly during transmission, but the receiver needs to be active constantly in order to receive any incoming signals.

Modern wireless receivers are commonly based on the principle of a superheterodyne receiver [5, Chapter 6] as illustrated in Figure 1.1. These types of receivers are based on the down-conversion from a high carrier frequency to a lower Intermediate Frequency with a local oscillator and a mixer. The signal received by the antenna is first amplified with a Low Noise Amplifier and filtered with a Band Pass Filter that is tuned to the frequency range of the receiver (i.e. the 2.4 GHz band). The mixer then multiplies the incoming signal with a signal from a local oscillator. The resulting Intermediate Frequency

[3] For example, the CC3200 WLAN System On Chip [4] draws a current of 166–229 mA for transmission and 59 mA for reception.

signal is then once again filtered and amplified before transformed to a digital signal using a high bandwidth Analog-To-Digital Converter. In modern system the demodulation of the signal is then implement in the digital domain on a dedicated Digital Signal Processor. This architecture is very flexible and provides a high sensitivity and performance, but it also has a high power consumption. There are many active components like the amplifiers, mixers and of course the Digital Signal Processor that all draw a significant amount of current.

If the receiver is enabled there are five possible scenarios from which four actually waste energy

Idle Listening There is no other device sending, still the receiver is enabled and listens to the medium.

Overhearing Reception and decoding of a frame that is intended for another device.

Collision Two or more transmitters send at the same time using the same channel.[4] Depending on the ratio of the signal strengths at most one of them can be received. The other/s is/are lost.

Failed Reception The received signal is intended for us but due to errors in the transmission the decoded signal is erroneous.

Successful Reception The received signal is intended for us and can be decoded successfully.

The prevention of collisions is a typical task of a MAC protocol as collisions not only waste energy but also degrade the performance of a wireless system. Also the detection of failed transmissions and following re-transmissions is usually done by MAC protocols. The main waste of energy, however, happens during idle listening and overhearing because in most scenarios the time during which there are no transmissions or only transmissions for other nodes is significantly larger than the time during which actual data is sent to and received by a specific node. Therefore many protocols that are optimized towards energy efficiency employ a duty cycling scheme in which the node changes between a low power sleep mode, where the main radio is disabled, and an active mode in which the radio is enabled and data can be exchanged. The ratio between active time and total time is called the *duty cycle*. The challenge here is to synchronize all nodes in the network, such that they can communicate with each other with as little overhead as possible. I will

[4] The word channel is commonly used to describe a certain frequency band. It could also mean a different spatial stream for Multiple-Input Multiple-Output systems or a different code for Direct-Sequence Spread Spectrum modulation.

introduce some existing protocols for WSNs in Section 3.1 and explain relevant parts of the IEEE 802.11 Wireless LAN (WLAN) standard in Section 4.1.1.

1.2 Traffic Pattern and Use Cases

This thesis focuses on the reduction of energy consumption in wireless networks by reducing the energy overhead caused by common duty cycling based protocols. This is especially important in networks with long idle times and little and sporadic traffic patterns. In such networks the energy required to transmit the actual payload is often not the main determining factor. Instead, a lot of energy is used to keep the network nodes synchronized either by regular exchanging packets[5] or by asynchronous mechanisms involving long preambles like in B-MAC [6].

The solutions presented in this thesis are built for such low-data scenarios, where nodes can stay inactive for a long time and only have to send or receive data sporadically. A typical scenario for this are Internet Of Things (IoT) networks that we investigate in Chapter 4 and Chapter 5, where sensors and actors may only have to be active once every hour or so and should ideally work with a single battery for many years.

1.2.1 BATS project

Another application example that has been investigated in Chapter 2 and Chapter 3 is a challenging interdisciplinary project called BATS[6]. The BATS project focused on the application of WSNs for wildlife monitoring. Together with biologists and electrical engineers, we developed lightweight sensor nodes that were attached to mouse-eared bats (*Myotis myotis*) in order to study their social behavior. This is a very challenging scenario, because bats can fly up to 50 km/s, which makes the network topology change rapidly [7]. Also, the sensor nodes have to be extremely lightweight (at most 2 g), since the bats themselves only weigh around 20 g. This results in an extremely low power budget. With these sensor nodes we were able to a) track bats during flight maneuvers, and b) to identify social interactions by monitoring contacts between individuals.

[5] i.e. the Access Point (AP) in a WLAN network sends out regular beacons that need to be received by the Stations (STAs) to keep in sync

[6] Dynamic Adaptable Applications for Bats Tracking by Embedded Communicating Systems, `http://www.for-bats.de/`

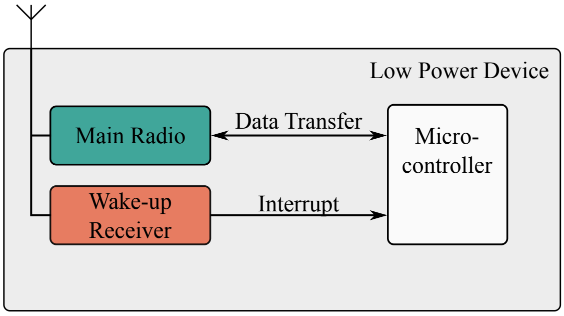


FIGURE 1.2 – Overview of a WUR capable wireless device. The main radio can be turned off most of the time. Upon the reception of a wake-up signal the microcontroller is woken up with a hardware interrupt and can commence communication over the main radio.

1.3 Principal Operation of Wake-up Receivers

Duty cycling based protocols can already reduce the energy wasted by idle listening and overhearing significantly. However, they require some form of synchronization among the nodes, either via schedules or long preambles. Such synchronization adds additional overhead to the communication, which requires energy and makes the protocols more complex. In a highly mobile scenario, such a synchronization might even be infeasible (see Section 3.3). Also, duty cycling protocols always trade off energy consumption and performance: A low duty cycle saves more energy, but increases delay and decreases throughput, while a larger duty cycle has better performance, but needs more energy. The main problem of duty cycling based protocols is that a node in sleep mode has its radio disabled and therefore cannot know if any other node in the neighborhood might want to send a packet. It has to wake up periodically and listen to any incoming signals. In the same way, if a node has data it wants to send to another node, it first has to wait until the other device wakes up.

Wake-Up Receivers (WURs) aim to solve this synchronization problem by offering a way to wake up a remote node with a radio signal. To this end, the devices of a wireless network are equipped with an additional radio receiver (see Figure 1.2), that can generate an interrupt to wake up the device upon the reception of a *wake-up signal*. In order to achieve energy efficiency with this approach, it is crucial that the energy Consumption of a WUR is *significantly lower* than that of a normal radio receiver. Current implementations only require a few µW or even less power (see Section 2.1). This can be achieved by using a much simpler receiver design than that of a superheterodyne receiver

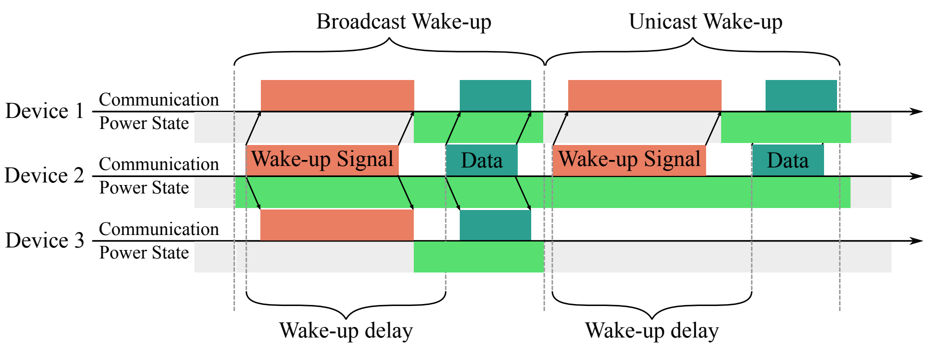


FIGURE 1.3 – Communication using a wake-up receiver: Device 1 and 3 are in sleep mode (gray); Device 2 wants to send data to all nodes and sends a wake-up signal; upon reception Device 1 and 3 wake up and can receive the packet from Device 2 via the main radio.

with much less active (i.e. energy consuming) components. I will describe how such a receiver can be built in Chapter 2. The reasoning behind such an additional receiver is the following: In contrast to normal data exchange, a wake-up signal does not have to convey a lot of data and can therefore use a much simpler modulation (usually OOK modulation) and receiver architecture. The advantage is, that the *receiving* node does not have to employ a costly idle listening or periodic wake-up scheme in order to be informed about a transmission attempt from another node. Instead, it follows a simple protocol as illustrated in Figure 1.3: All devices are by default in a sleep mode (depicted by the grey bar in the figure) and have their main radio disabled. The WUR can be enabled continuously, because it only draws very little power from the battery. If a node now has new data that it wants to transmit (Device 2 in the example), it will first send out a wake-up signal. The device or the devices that receive this wake-up signal change to an active state, enable their main radio and normal data exchange can commence. After the transmission, the devices can go back to sleep mode. If the WUR has addressing capabilities, a wake-up signal can be specific for a single node. With simpler systems a wake-up signal will cause all nodes in communication range to wake up.

The asynchronous nature of the WUR greatly simplifies protocol design. Instead of sending long preambles or agreeing on common schedules (which also requires some means to compensate clock drift), all nodes can stay in a low power mode until they are woken up remotely or decide on their own to wake up via a timer or some external interrupt.[7] However, this technique

[7] In WSNs for example, the nodes are equipped with sensors and monitor their environment.

also has some limitations. First and foremost, the sensitivity[8] of WURs is usually well below that of more complex receivers. This is due to the fact that a high sensitivity requires some form of amplification, which costs energy. This, however, limits the wake-up range because the transmission power can not be arbitrarily increased.[9] In such a case, the set of nodes that can be communicated with using the main radio can be different from the set of nodes that can be woken up. Also the low data rate of WURs can become a problem if many wake-up signals need to be send, because the long signals block the wireless channel and also require a transmitter to consume energy for a longer time.

1.4 Research Questions and Contributions

The design of energy efficient communication systems can be achieved on many different levels. In this thesis I will focus on the MAC layer that coordinates the communication between direct neighbors, i.e. between nodes in direct communication range. The design of a MAC protocol can be based on different objectives, that can even be conflicting with each other. Typical goals are: maximizing the efficiency of resource utilization (throughput), minimizing signal collisions, minimizing the latency/delay, simplicity (small overhead), fairness and *energy consumption*. It also has a huge influence on the design of a MAC protocol if there is a central entity like a base station or an access point available, that can coordinate the medium access, or if all nodes are considered equal and have to coordinate the medium access in a decentralized way.

In order to achieve energy efficiency *and* a good performance I will tackle the following problems:

- **Development of a novel addressing scheme for WURs** that increases the flexibility of waking-up specific nodes or group of nodes in an energy efficient way. The concept of wake-up receivers is based on the assumption that it is possible to build a receiver with a power consumption that is magnitudes lower than that of regular radio receivers and can therefore listen to the channel continuously. The challenge here is to also have a good sensitivity. A high sensitivity is preferable as it increases

[8] The sensitivity determines how strong a signal has to be at least so that the receiver can detect and decode the signal. The sensitivity is typically given in dBm, a unit that describes the strength of the signal in relation to 1 mW

[9] Usually the maximum power is limited by regulations, especially in the Industrial, Scientific And Medical bands that are shared with other systems. Also a higher transmission power would increase the energy consumption, defeating the purpose of a WUR

the possible communication range. Ideally the sensitivity of the WUR should be as good as the one of the main radio, even though this is very hard to achieve. and a high data rate. These optimization goals are usually conflicting with each other since a high sensitivity requires some form of amplification of a weak signal (and amplifiers are active components and thus consume energy) and high data rates require high bandwidth[10] which in turn requires more energy to process. A WUR should also have an addressing functionality so that only specific nodes can be woken up, saving energy on all other nodes. In this thesis I will give an overview on existing solutions and present a new addressing scheme that enables very flexible communication protocols while keeping the additional energy consumption low.

- **Design of WUR based MAC protocol for highly dynamic networks.** The ability to wake up nodes with a wireless signal can be used in different ways, depending on the network topology and the requirements of the system. In networks with little or no mobility we can apply different strategies than in highly mobile networks. In homogeneous networks all nodes also have to *transmit* wake-up signals, which requires a different MAC design than in heterogeneous networks where only a central entity emits wake-up signals. While the basic concept of wake-up receiver based communication as shown in Figure 1.3 is simple and straightforward, the details and the properties of specific networks require a more complex design and evaluation. In this thesis I will present a wake-up based solution for a challenging, highly mobile WSN in the realm of wildlife monitoring.

- **Wake-Up Receiver based Wireless LAN** Even though the idea of WURs was first discussed in the realm of WSNs, its basic operation principle can also be applied in different domains. In centralized networks like normal consumer WLAN multiple mobile nodes (the Stations (STAs)) are connected to a central node (the Access Point (AP)). In such a scenario the AP is not energy constrained and does not have to perform any power saving techniques. Only the mobile STAs have to save energy, which is why the IEEE 802.11 standard already defines a duty cycling based Power Save Mode (PSM). In this thesis I will show how a WUR can be used to reduce the energy consumption of mobile devices without the drawbacks of the existing PSM.

[10] According to the Shannon-Hartley theorem the capacity C of a communication channel is given by $C = B \cdot log_2(1 + S/N)$, where B is the bandwidth (in Hz), and S/N is the signal to noise ratio.

- **Low Power Downlink using Wake-Up Receiver** Even though the solutions from the first chapters and from other researchers already show promising results for specific applications, it is desirable to have a more general communication paradigm that is applicable to a broad range of applications. Therefore, in the final technical chapter, I will introduce an extension of the WUR concept that treats the additional low power receiver as general purpose receiver that can receive not just wake-up signals, but arbitrary information in an energy efficient way. This dual radio communication paradigm also supports the basic wake-up pattern, but it can be used in a more flexible way and can therefore be used for a wide range of applications with little adaption needed.

1.5 Publications

Chapters 2–5 of this thesis are based on the results from the following peer-reviewed publications:

1. J. Blobel, C. Sommer, and F. Dressler, "Protocol Options for Low Power Sensor Network MAC using Wake-up Receivers with Duty Cycling," in *IEEE International Conference on Communications (ICC 2016)*, Kuala Lumpur, Malaysia: IEEE, May 2016, pp. 3925–3930. DOI: 10.1109/ICC.2016.7511318

   In this publication I created a detailed simulation model to investigate the performance of the different protocols. The model includes an energy model to simulate using a capacitor as energy source, and the wake-up enabled MAC models for the communication protocols. In conjunction with the movement model from [7] I run an extensive simulation campaign and evaluated the results.

2. J. Blobel, J. Krasemann, and F. Dressler, "An Architecture for Sender-based Addressing for Selective Sensor Network Wake-Up Receivers," in *17th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2016)*, Coimbra, Portugal: IEEE, Jun. 2016. DOI: 10.1109/WoWMoM.2016.7523516

   In this paper I introduced my idea of sending an address *and* a mask within the wake-up signal. I developed the matching logic and implemented it with a hardware prototype. The simulation results are based on the bachelor thesis from Janis Krasemann, which I supervised.

3. J. Blobel and F. Dressler, “Sender-Triggered Selective Wake-Up Receiver for Low-Power Sensor Networks,” in *36th IEEE Conference on Computer Communications (INFOCOM 2017), Demo Session*, Atlanta, GA: IEEE, May 2017. DOI: 10.1109/INFCOMW.2017.8116522

   I presented the prototype of the selective WUR at the Demo session of INFOCOM 2017.

4. J. Blobel, F. Menne, D. Yu, X. Cheng, and F. Dressler, “Low-power and Low-delay WLAN using Wake-up Receivers,” *IEEE Transactions on Mobile Computing (TMC)*, Oct. 2020. DOI: 10.1109/TMC.2020.3030313

   In this publication I applied the WUR concept to WLAN based systems. For this, I first implemented the normal WLAN Power Save Mode, since it was not yet available in INET. I then created a simulation model that adds a WUR to a normal WLAN and adapted the MAC protocol to support this new mode of operation. The prototype was build by Florian Menne in his Master thesis, which I supervised. I was also responsible for the final experiments and the evaluation of the collected data.

5. J. Blobel, V. H. Tran, A. Misra, and F. Dressler, “Low-Power Downlink for the Internet of Things using IEEE 802.11-compliant Wake-Up Receivers,” in *40th IEEE International Conference on Computer Communications (INFOCOM 2021)*, Virtual Conference: IEEE, May 2021

   For this publication I defined the LPD protocol format and created the simulation model. The model includes the wireless energy transfer and an extended wireless model to simulate the WLAN compatible Centrally Controlled Channel Access. I conducted the simulations and evaluated the results. I also helped with the development of the prototype and the evaluation of the measurement data.

During my Ph.D. studies I further co-authored the following peer-reviewed papers:

6. Q.-H. Nguyen, J. Blobel, and F. Dressler, “Energy Consumption Measurements as a Basis for Computational Offloading for Android Smartphones,” in *14th IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (EUC 2016)*, Paris, France: IEEE, Aug. 2016. DOI: 10.1109/CSE-EUC-DCABES.2016.157

7. F. Klingler, J. Blobel, and F. Dressler, “Agriculture meets IEEE 802.11p: A Feasibility Study,” in *15th IEEE International Symposium on Wireless Communication Systems (ISWCS 2018)*, Lisbon, Portugal: IEEE, Aug. 2018. DOI: 10.1109/ISWCS.2018.8491239

8. J. Blobel, “Energy Efficient Communication using Wake-up Receivers,” in *International Conference on Networked Systems (NetSys 2019), PhD Forum*, Munich, Germany, Mar. 2019

9. F. Dressler, M. Mutschlechner, M. Nabeel, and J. Blobel, “Ultra Low-Power Sensor Networks for Next Generation Wildlife Monitoring,” in *11th IEEE International Conference on Communication Systems and Networks (COMSNETS 2019)*, Bengaluru, India: IEEE, Jan. 2019. DOI: 10.1109/COMSNETS.2019.8711475

10. M. S. Amjad, G. S. Pannu, A. Memedi, M. Nabeel, J. Blobel, F. Missbrenner, and F. Dressler, “A Flexible Real-Time Software-based Multi-Band Channel Sounder,” in *31st IEEE Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2020)*, Virtual Conference: IEEE, Aug. 2020. DOI: 10.1109/PIMRC48278.2020.9217369

1.6 Structure of this Thesis

The remainder of this thesis is structured as follows: In Chapter 2 I will introduce typical hardware concepts that are used to build a WUR and give an overview on the current state of research. I will then introduce the concept of a *Selective Wake-Up Receiver (SWUR)*, a flexible, sender-based addressing solution, and explain how such a receiver can be build and used in an application example.

Following, in Chapter 3, I will introduce duty-cycling and WUR based MAC protocols for WSNs and explore different WUR based MAC protocol options in a highly mobile, ultra-low energy WSN in the realm of wildlife monitoring. Chapter 4 focuses on a centralized wireless network, namely WLAN with a central AP and multiple mobile STAs. I will show how WURs can be used to make WLAN more energy efficient than with existing, duty-cycling based solutions while maintaining a good performance in terms of delay and throughput. In Chapter 5 I introduce a more generalized concept, called Low Power Downlink (LPD), that can further reduce the energy consumption of wireless devices while maintaining compatibility with existing WLAN networks. Finally, in Chapter 6, I will conclude the findings of this thesis and give an outlook on open research questions and possible extensions to the concepts introduced.

Chapter 2

Wake-Up Receiver Hardware

BEVORE exploring the possible applications for Wake-Up Receivers (WURs) in different domains we shall first discuss how such a WUR can be built. This chapter includes an overview of different WUR technologies and their capabilities and limitations. I will then present a hardware prototype that extends existing WURs with a flexible, sender-based addressing scheme.

2.1 Wake-up Receiver Hardware

Modern Radio Frequency (RF) receivers, that use a superheterodyne architecture, include many active components like Low Noise Amplifier, Phase-Looked Loop and mixers, that operate on high frequencies and therefore consume a lot of power, regardless if the later stages are receiving and decoding a signal or not. It becomes apparent that for battery operated devices it is prohibitive to listen to the channel continuously (*idle listening*) as this would drain the battery in a short time. This problem has so far been tackled by so called duty-cycling protocols, that disable the main transceiver for a certain time and just switch it on briefly to communicate with other devices. An overview of these protocols and a discussion about their strength and weaknesses is given in Section 3.1.

Another approach to solve the problem of high energy consumption of superheterodyne radio receivers are *Wake-Up Receivers*. The main idea of these receivers is not to replace existing radios, but to complement them. By choosing a much simpler architecture one can use mostly passive components that require much less energy. This, however, leads to a reduced sensitivity and lower data rate which is why WURs cannot replace normal radios. Instead they are used – as the name suggests – to wake up remote nodes using a radio signal

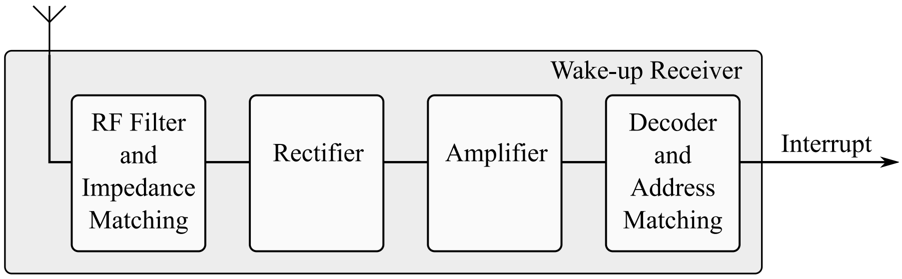


FIGURE 2.1 – Block diagram showing the main components of a generic wake-up receiver.

called *wake-up signal*. Because of the low power consumption the wake-up receiver can be in reception mode all the time and wait for such a wake-up signal. Upon reception the WUR can then signal this to the Microcontroller with a hardware interrupt which can then turn on the main radio to exchange data.

The concept of wake-up radios was first introduced in the realm of Wireless Sensor Networks (WSNs) [18], where the authors propose a "Remote Activated Switch (RAS)" in addition to the existing radio. While it was called a different name, the concept is that of a wake-up receiver and was inspired by passive devices known as Radio Frequency Identification (RFID):

> "Based on RF tags technology, we can develop a switch that can be used to remotely activate a radio device while being in sleep state. In this way, nodes are woken up when necessary rather than getting active periodically to verify whether there is pending traffic." (Chiasserini and Rao [18])

The general architecture of most wake-up systems is depicted in Figure 2.1. To only receive signals from a certain frequency band and to optimize energy transmission to the later stage a filter and impedance matching circuit are usually the first stage after the antenna. Many of the proposed systems are based on mainly passive components using one or more diodes to rectify the incoming signal. The rectified signal can then further be amplified using a low power amplifier or a passive voltage multiplier as proposed in [19], [20]. A voltage multiplier has the advantage that it does not increases the power consumption because it only contains passive components (diodes and capacitors), it does, however, reduce the possible datarate because the circuit does not react immediately to the incoming signal. Using an amplifier increases the energy consumption of the receiver because it is an active component but allows for higher data rates. The received signal can directly be used to generate the interrupt that wakes up the microcontroller. It would therefore just serve as a RF power detector and would lead to many false wake-ups because other signals on the same channel would also trigger the circuit.

To prevent such false wake-ups, an additional stage has been integrated that can detect if an incoming signal is just noise or a valid wake-up signal. One of the first papers on WURs already included a simple addressing scheme based on the utilization of multiple radio frequencies [21]. Another approach was presented in [22] where a simple energy detector circuit would first enable a more complex stage including an Low Noise Amplifier and an address decoder that can receive and check a Pulse Width Modulation modulated signal.

An example of the analog radio frontend of a wake-up receiver including an impedance matching network, a full wave diode rectifier, amplification and signal digitalization stage can be seen in Figure 2.2. This circuit has been simulated using SPICE whose output is shown in Figure 2.3. In the first stage the incoming signal (*SIG*) is fed to an impedance matching network to maximize the energy that is transferred from the antenna to the circuit. This alternating current (AC) signal is fed to a full wave rectifier circuit that uses the SMS7630 low barrier detector Schottky diode from Skyworks to create a direct current (DC) signal (*Detect*). To improve the sensitivity this signal

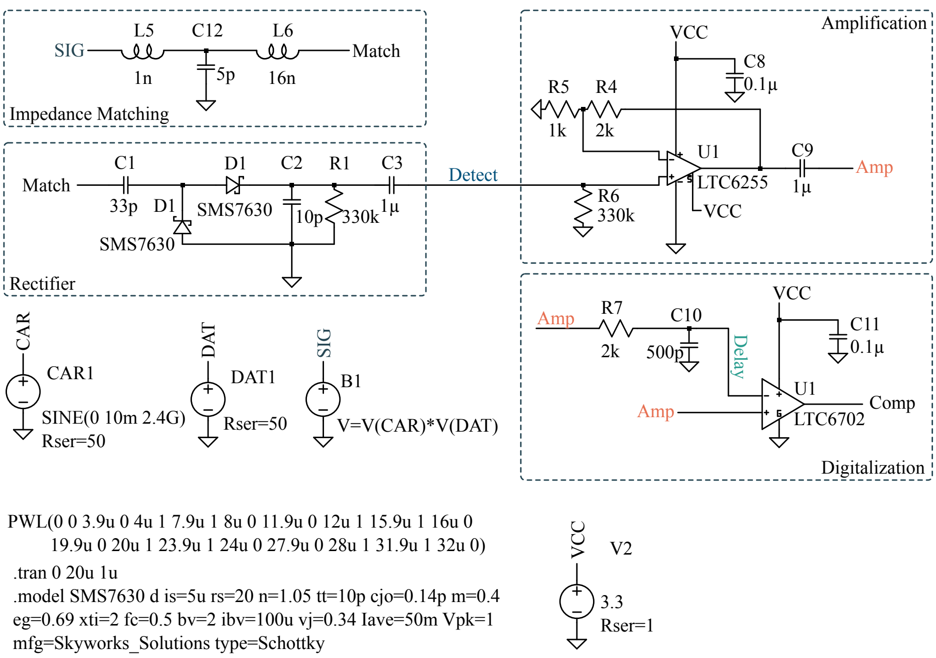


FIGURE 2.2 – Circuit diagram of a simple semi-passive wake-up receiver. It includes an impedance matching network, a full wave diode rectifier, amplification and signal digitalization stage.

is then amplified using a low power LTC6255 operational amplifier (*Amp*). As a final step the signal is converted to a digital signal using an LTC6702 comparator (*Comp*). The threshold of the comparator is not set to a fixed value here, instead the amplified signal is delayed using an RC filter and used as threshold (*Delay*). Using this technique the wake-up receiver works over a wide range of signal strengths. In Figure 2.3 it can be seen that if the amplified signal is larger than the delayed one the output of the comparator turns to HIGH (3.3 V). If it falls below the delayed signal the output of the comparator is LOW (0 V).

Since the first wake-up receivers many systems have been proposed, that differ in sensitivity (which determines the wake-up range), power consumption, delay, and addressing capabilities. An overview and comparison of different WUR implementations along with a discussion of their benefits and drawbacks can be found in [23] and [24].

Addressing functionality is often implemented using a microcontroller or with a correlator within the WUR [23]. A different approach for address matching using the timing between two consecutive signals and the help of the microcontroller was proposed in [25]. The address is encoded within the time between two consecutive signals. Each signal wakes up the microcontroller very shortly to measure the timing and check for a match. The authors compare their addressing scheme to the correlator-based address matching and microcontroller-decoding of the signal. While this scheme only uses very

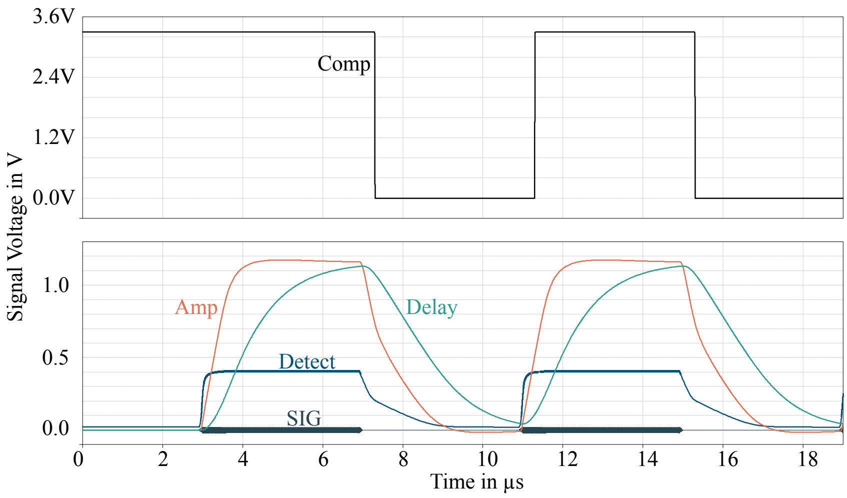


FIGURE 2.3 – Results of the SPICE simulation: The incoming signal is rectified, amplified (lower part) and converted to a digital signal (upper part).

little power, the duration of a wake-up signal is rather long (48 ms to 68 ms), since each address bit is encoded using a delay between two signals.

The idea of using Bloom-filters for group based ID matching in a WUR has been presented in [26]. The transmitted wake-up signal includes a wake-up id which is based on the intended groups that should be woken woken up. Each node has its own bloom-filter, that specifies the groups it belongs to. If the received signal matches, i.e., the node belongs to one of the transmitted groups, the node is woken up. This scheme cannot wake-up single nodes, but only groups of nodes.

Many WUR implementations only have a very poor sensitivity, which limits the reception range to less then 10 m. The WUR presented in [27], however, can reach a sensitivity of −83 dBm. This allows for a wake-up distance of 1200 m (when sending with 10 dBm). However, the low current consumption of 3 µA and the good sensitivity comes at the cost of a very high latency of 484 ms.

2.2 Selective Wake-up Receiver (SWUR)

Most wake-up receivers that are commercially available today (i.e. AS3933 from AMS) can decode an On-Off-Keying (OOK) modulated signal and correlate that with a predefined pattern. The main objective of such a correlation is to prevent false wake-ups by other signals on the same channel. In such a configuration all nodes would be configured to have the same pattern, hence only allowing to wake up all nodes at the same time (*broadcast*). If, however, all nodes get their own pattern it also acts as a unique address that allows to wake up each node individually (*unicast*). A mix of these two options is possible if groups of nodes have common patterns configured which would allow a multicast wake-up. In [28] the authors presented an architecture that allows nodes to have more than one address where one could serve as a unicast address and one as a broadcast address.

While such solutions can already offer a certain degree of flexibility they are still limited when it comes to addressing a subset of nodes (*multicast*). The architecture for a selective wake-up receiver that we introduced in [9] adds the possibility to choose the communication scheme by the *sender* of the wake-up signal without the need of reconfiguring the receiving nodes. Before explaining the logic design, let us consider the basic working principle:

- Each node has a unique pattern configured which serves as an address.
- The wake-up signal contains an address *and* a mask.
- A node is woken up if the bits of the received address determined by the mask match the configured pattern.

Figure 2.4 illustrates the pattern matching for a multicast wake-up. The unique pattern (node address) of the receiving node is shown in the first row (blue). The received signal (address and mask) is shown in the next two rows (green). The first two bits of the mask are set to 1, therefore, the stored pattern and the received address have to be equal at this positions in order to get a match. The last two bits of the mask are 0, indicating that these bits are not relevant for the address matching, therefore these two bits will always match.

By specifying the mask that is sent with the wake-up signal, the sender can determine the communication scheme. If all mask bits are set to 1 only one node with the matching address will wake up, which is a unicast wake-up. For a broadcast all bits of the mask are set to 0 all nodes receiving this signal wake up. For a multicast wake-up a subset of relevant bits are set to 1, which determine the parts of the address that have to match with the pattern of the receiver. This allows waking up a subset of nodes.

2.2.1 Logic Design

To add the aforementioned functionality to an existing WUR only little additional hardware is required. In fact, only a couple of additional standard logic gates are sufficient to implement the matching logic. A conventional correlator simply compares the received address with the pattern that is stored in a register. If all bits match, an interrupt pin is triggered, which will wake up the microcontroller [23].

By using additional logic gates the correlator can be extended to also check the mask of the received signal. To do so the received data first has to be shifted

Configured Pattern	1	0	1	1
Received Address	1	0	0	0
Received Mask	1	1	0	0
Match	1	1	1	1

FIGURE 2.4 – Example of the pattern matching for a *multicast* wake-up

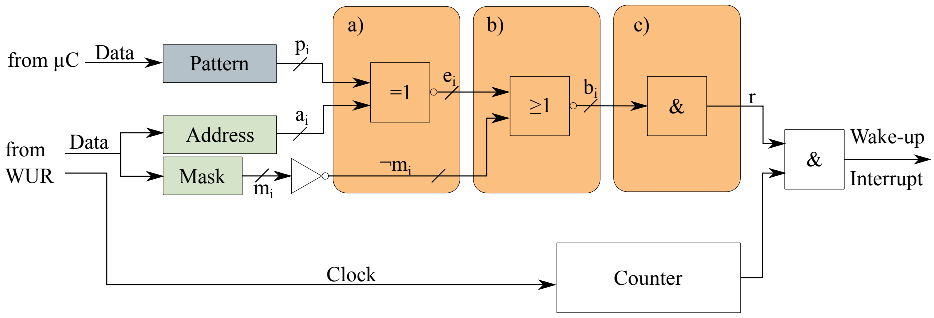


FIGURE 2.5 – Logical Structure of the Selective Wake-Up Receiver. The data is shifted into the registers. From there the logic combinations are done in parallel for each bit and finally combined to the resulting signal r. After n clock cycles, the output of the counter goes high, signaling that r is now stable.

into a register in order to correlate the received signal with the predefined pattern with parallel logic gates.

Figure 2.5 shows the conceptual structure of the Selective Wake-Up Receiver (SWUR). The pattern $P = [p_1, p_2, \ldots, p_n]$ is set by the microcontroller and stored in a register. The address $A = [a_1, a_2, \ldots, a_n]$ and mask $M = [m_1, m_2, \ldots, m_n]$ are received by the wake-up receiver (AS3933) and are also shifted into registers. For all bits $i \in \{1 \ldots n\}$, the following logical combinations are performed in three steps:

(a) $e_i = \text{XNOR}(a_i, p_i)$: True if pattern and address are equal.

(b) $b_i = \text{OR}(e_i, \neg m_i)$: True if either e_i is 1 or mask is 0.

(c) $r = \text{AND}(b_i \forall i \in \{1 \ldots n\})$: True if all bits match.

In other words, the output of the correlator for bit i is either 1 if the mask is 1 *AND* the address and pattern are equal or if the mask is 0. In the last step, all results b_i from phase (b) are merged with one n-AND gate. As a result, the wake-up interrupt will only be generated if for all bits the result of step (b) is 1.

While the data is shifted into the registers sequentially the output of the circuit is not stable and changes constantly. Since we are only interested in the state of the logic circuit after all bits are received we also added a counter that is increased with each incoming bit.[11] If all bits are received, the output

[11] This counting is especially easy in this case and does not require any synchronized clock because the wake-up signal is Manchester modulated which enables the AS3933 chip to recover and output a clock signal

of the counter switches to 1 which signifies the end of the reception. The final interrupt signal is then generated by a final *AND* gate that is 1 only if the last stage of the matching circuit (step (c)) as described before and the counter outputs a 1.

2.2.2 Broadcast Optimization and Mask Encoding

As discussed in Section 2.1 the data rate provided by WURs is typically very low compared to the main radio. This has two major implications: First, it means that the channel is blocked for a rather long time, which can have a negative impact on other transmissions on the same channel. Second, it means that in a homogeneous network where the energy constrained nodes also *send* wake-up signals, the transmission can consume a considerable amount of energy (see Section 2.5). Therefore it is desirable to reduce the length of the wake-up signal as much as possible.

We can further optimize the wake-up protocol by only sending information that is strictly necessary. A broadcast signal for example does not require an address to be sent because it is ignored anyway. The architecture can easily be optimized in this case by simply sending the mask bits first and adding another logic test. An interrupt can then be raised immediately after the reception of n 0 (mask-) bits, which would signify a broadcast. With this shortcut we can halve the energy required to send a broadcast wake-up. A similar technique to reduce the length of the wake-up signal is introduced in Section 5.1.4.

2.3 Hardware Prototype

Figure 2.6 shows our hardware prototype which was built using standard logic gates, shift registers and counters. The circuit design can be found in Appendix A. On the right side of the Printed Circuit Board is the analog frontend with an impedance matching circuit, envelope detector and the AS3933 wake-up receiver. The left side contains debugging LEDs[12] and the logic circuits. The logic gates used in this prototype are quadruple gates which is why the address and mask of the prototype have 4 bit. We used standard logic gates from the 74HCxx family, like the 74HC08 quad 2-input AND gate used in step (c).

The analog frontend uses the AS3933 chip to decode the Manchester coded OOK signal. This chip is actually built for RFID applications and expects a

[12] These LEDs can be disabled with a jumper for measuring the power consumption. The red LEDs show the received address, green shows the received mask and the yellow LEDs show the local address of the device

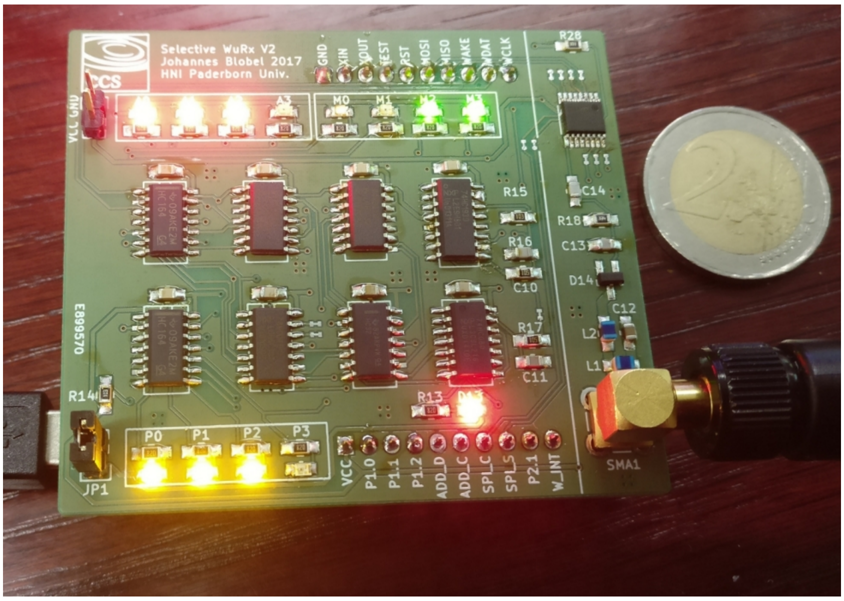


FIGURE 2.6 – SWUR prototype using standard logic ICs. The top LEDs show the received address (red) and mask (green), the lower ones the node's address (yellow).

125 kHz signal as input. We therefore added a diode based rectifier circuit that extracts the envelope of the incoming 868 MHz RF signal. The sender then modulates the 868 MHz carrier with an additional 125 kHz signal to send the symbol 1 (see Appendix B). The data and clock output of the AS3933 is finally fed to the digital logic circuit where the address matching is performed.

The received data is first stored in a 74HC164 shift register whose outputs are connected to the logic gates (U1 in Appendix A). The clock signal is fed to a 74HC193 4-bit binary counter that will output a 1 on its 4th output Q3 after 8 clock cycles. In order to reset the circuit this counter signal is fed to an RC filter that will reset the circuit after 1 ms. The whole system is controlled by an MSP430G2553 microcontroller that can configure the AS3933 via its Serial Peripheral Interface and write the local node address into the corresponding shift register (U2). The microcontroller then enters a deep sleep mode (LPM4) from which it can only be woken up by an external interrupt. If a wake-up signal is received and an interrupt is triggered, the microcontroller will run for 3 ms and then enter sleep mode again.

For sending the wake-up signal we used a Software Defined Radio (Ettus B210 USRP) together with the GNU Radio signal processing framework. The flowgraph that generates the OOK signal with the correct structure can be

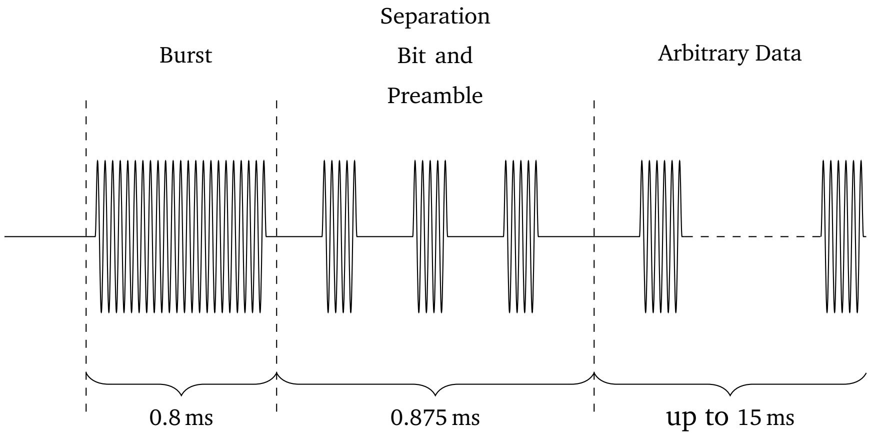


FIGURE 2.7 – The AS3933 expects a 125 kHz signal with the following structure: A burst, followed by a separation bit and a preamble and finally the actual data.

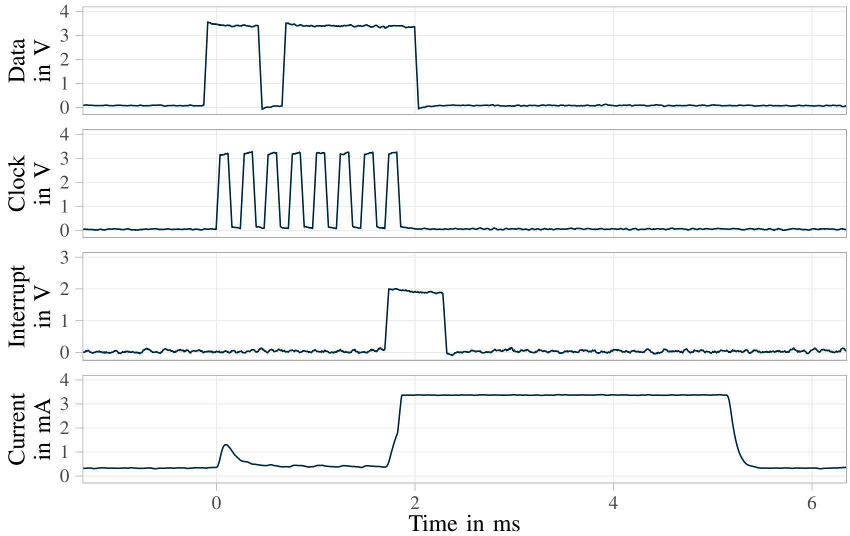
seen in Appendix B. The AS3933 chip expects a certain structure to detect and decode a signal as illustrated in Figure 2.7:

1. Burst: used to detect the signal and perform Automatic Gain Control
2. Separation Bit and Preamble: detect start of frame and synchronization
3. Data which can contain:
    - Pattern: To prevent false wake ups
    - Arbitrary data: In our prototype this is the address and mask of the wake-up signal

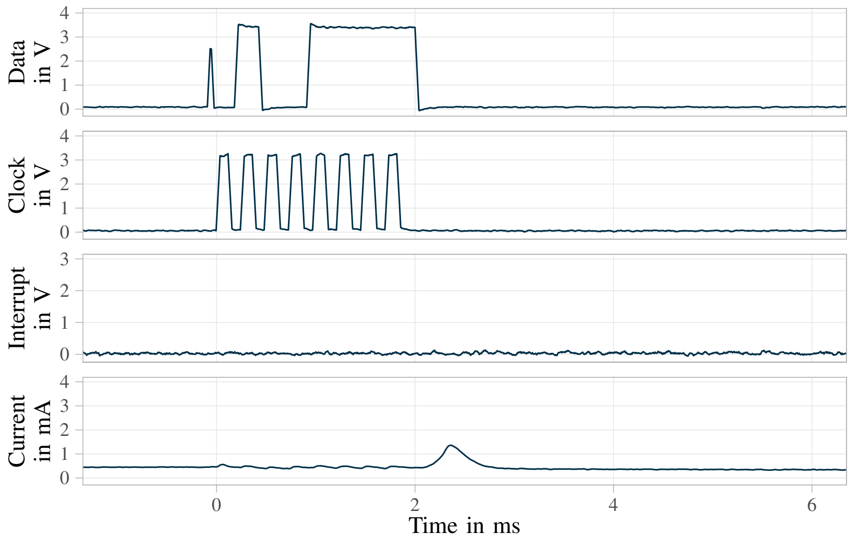
The AS3933 can output any data this is received after the preamble. If we disable the pattern matching functionality of the receiver and send the data right after the preamble this output, however, was not stable and sometimes also included the last preamble bit. As this leads to errors in the logic circuit we had to first send a constant pattern of 16 bit that is checked by the chip and then the actual payload data (address and mask). This makes the wake-up signal longer than strictly necessary but was required for this prototype to ensure a stable operation.

2.3.1 Qualitative Measurements of Prototype

To show how the prototype reacts to different signals we performed initial validation experiments and measured the internal signal levels and the current



(a) The received address *does* match the preconfigured pattern: an interrupt is raised.



(b) The received address does *not* match the preconfigured pattern: no interrupt is raised.

FIGURE 2.8 – Signals and current consumption of the prototype during reception of a unicast wake-up signal.

drawn from the device. For this we set the local address of the receiver to “1101” and then sent different wake-up signals while running the measurements. Figures 2.8a and 2.8b show the voltage trace of the incoming data and clock signals, the wake-up interrupt signal, and the current consumption of the testbed. The first figure shows the system if we send a matching unicast wake-up signal (address is “1101”, mask is “1111”). As the results of the logic circuit is 1 in this case, an interrupt is generated which in turn causes the microcontroller to wake up which causes the rise in the current consumption. The total energy consumed during this experiment was 46 mJ.

If we send a signal that does not match, the interrupt is not generated, as shown in Figure 2.8b. The received address “0100” does not match, thus the microcontroller is not woken up. The power consumption is only raised slightly, which is an artifact of the RC filter. In total only 14 mJ where consumed. The validation measurements show that the logic circuit works as expected and can successfully receive, decode and check the address and mask of the wake-up signal, all without the help of the microcontroller.

2.4 SWUR for Software Updates of Low-Power Sensor Nodes

To demonstrate how the flexibility of the proposed addressing scheme can be used I will introduce an application example in this section and show how the multicast addressing can be applied. It was developed in the context of the BATS (Dynamic Adaptable Applications for Bats Tracking by Embedded Communicating Systems) project [29]. The goal of this project was to enable biologists to observe bats in their natural habitat. This is a difficult task because they are small, nocturnal animals that can fly very fast (up to 50 km/h) which makes detailed observation of their behavior very challenging. In this project we created very small and lightweight (less than 2 g) sensor nodes that were attached to the animals. The sensor nodes sent out a unique id to neighboring nodes at a fixed rate. The neighborhood information collected this way could then be transmitted to a set of ground nodes that were distributed in the forest. This downlink[13] communication was very challenging because the contact times where data could be transferred was typically just within a couple of seconds, and the wireless channel was unreliable due to multipath propagation in the forest environment. We successfully used Erasure Codes as a forward error correction to transmit the collected data from the bats to the ground

[13] Downlink here means the transmission from the flying bats *down* to the base stations.

nodes [30]. Further discussions about possible communication protocols for this scenario can be found in Chapter 3.

Communication in the other direction, however, requires a different protocol to upload a new software to the sensor nodes. As we cannot predict when a bat enters the ground node area we cannot easily schedule the transmission of the data packets. Also, the short contact times do not allow to use a sophisticated protocol that would detect the presence of a bat, determine which packets the node has already received and then transmit any missing packets. Instead, we investigated the usage of rate–less Erasure Codes, namely fountain codes [31], to send firmware updates to the mobile nodes. To make this process as energy efficient as possible we included an SWUR to the protocol. Fountain codes first divide the data that should be transmitted into several source packets. From these source packets an arbitrary amount of packets can be derived (rate less code) using a generator function and transmitted constantly. Each node now has to receive enough of these packets to successfully decode the original data. How many packets are needed to decode all data depends on which packets are received and this cannot be determined beforehand. The advantage of this scheme is that there is no need to retransmit any lost packets. The ground nodes just continue sending packets until all mobile nodes acknowledged that they were able to decode the data.

Since the bats enter the ground node area at different times and packet loss can occur, it is likely that some nodes can decode the data sooner than others. If such a node would now receive further packets this would consume unnecessary energy. We therefore propose to use a Selective Wake-Up Receiver to reduce such false wake-ups. We can use the addressing capability of the receiver to send any further fountain encoded packets only to nodes that have not been able to decode the software update yet.

The protocol to update the software on the mobile nodes is as follows: First, the ground nodes start to send fountain encoded packets that are addressed to all nodes (broadcast). The mobile nodes will start to receive those packets and eventually be able to decode the software update. If the first nodes start to acknowledge the successful decoding of the update to the ground nodes, the addressing scheme will change to split up the receivers into groups using use multicast or even unicast packets. This way we can prevent that nodes which already received the update have to waste energy on receiving any additional packets.

The splitting algorithm, which determines the optimal addressing scheme, tries to minimize the number of false wake-ups by updating the set of recipients every time another node acknowledges the successful decoding of the update by creating new multicast groups. The quality of this algorithm depends on the granularity with which it can select the groups which in turn depends on the address and mask length. As discussed before, one optimization goal is to reduce the length of the wake-up packets, but if we choose a smaller address space, the splitting algorithm cannot create perfect groups which leads to false wake-ups. By changing the length of the address and mask from 1 to $log n$ we can scale the addressing granularity from broadcast only to node individual addressing for n nodes.

To evaluate how the SWUR assisted update protocol performs under different channel conditions and different wake-up lengths we created a simulation model using OMNeT++ [32] and the MiXiM framework [33]. In our evaluation scenario we simulated 128 bats that move according to the mobility model from [7] which means that the nodes can fly in and out the communication region over the course of the simulation. In the middle of the scenario we placed a single ground node that distributes an update using the aforementioned protocol. In order to successfully decode the update each node has to receive at least 10 fountain coded packets. Since Erasure Codes are developed for a binary channel (either a packet is received completely or not at all) we also model the wireless channel by changing the packet loss probability. This also implies that the acknowledgments from the nodes can be lost, which would cause false wake-ups. In contrast to the hardware prototype from Section 2.3, we used a binary encoded mask that determines the number of bits that are masked, not each bit individually.

We evaluated the protocol using the following wake-up lengths:

- 1 bit (1 bit for the mask, 0 bit for the address): Baseline configuration using broadcast only operation.
- 2 bit (1 bit for the mask, 1 bit for the address): This enables the ground node to address two separate groups with 64 nodes.
- 5 bit (2 bit for the mask, 3 bit for the address): Smaller groups of 16 nodes can be addressed.
- 10 bit (3 bit for the mask, 7 bit for the address): Optimal configuration for $2^7 = 128$ nodes. Each node can be addressed individually.

The results from the simulations are shown in Figure 2.9: For each parameter configuration we simulated the scenario 50 times and recorded the number of false wake-ups. The results clearly show that the granularity in addressing introduced by the SWUR can greatly reduce the number of false wake-ups and hence the energy required to receive an update. With longer wake-up addresses and masks the algorithm can address the nodes more precisely, which reduces the amount of false wake-ups. This positive impact of

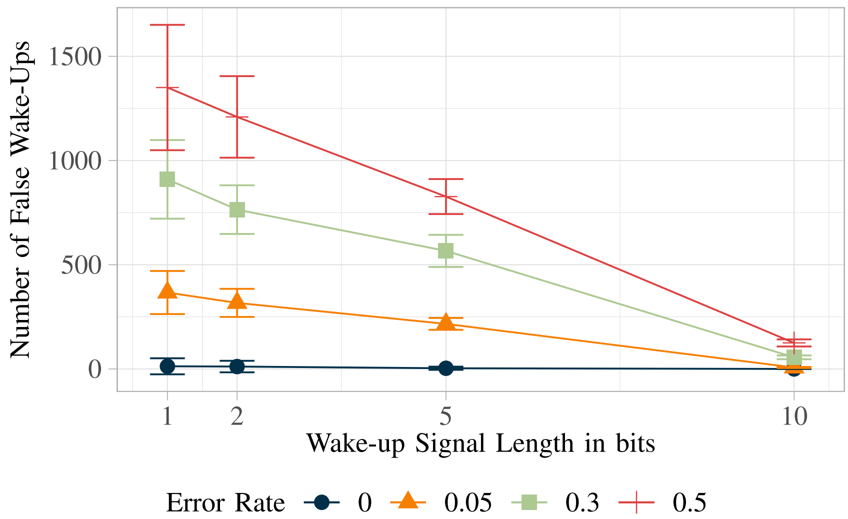


FIGURE 2.9 – Mean and standard deviation of false wake-ups during the update process.

the wake-up length is more distinct if the communication channel shows a higher packet error rate. With larger error rate the relative difference between the scenario with 1 bit and 10 bit increases.

2.5 Analysis of Energy Savings in Homogeneous Networks

So far we have investigated how Wake-Up Receivers can be used to reduce the energy consumption in a WSN by avoiding idle listening and overhearing, both problems related to the *reception* of RF signals. But also the *transmission* of either data or wake-up packets consumes energy which has to be taken into consideration. In the application example from the previous section the ground nodes are equipped with large batteries that can also be recharged. The energy consumption of such nodes for transmitting and receiving is therefore not of particular interest. But in homogeneous networks, where all participating nodes are energy constrained, not only the reception but also the transmission of wake-up signals has to be taken into consideration.

As mentioned before, the data rate that can be achieved by WURs is low compared to more complex radios. This also means that the duration to send a certain amount of data is rather long and, therefore, consumes more energy.[14]

[14] This is one of the reasons why wake-up signals should be kept as short as possible as discussed in Section 2.2.2.

The selectivity of our SWUR can save energy by preventing false wake-ups, but the costs for sending the additional data (address and mask) also have to be taken into consideration when designing a communication protocol for homogeneous networks. To save energy within the network, the additional energy to send a longer wake-up signal must be smaller than the energy that we can save by not waking up nodes unnecessarily.

To analyze this tradeoff more precisely, we will consider the following scenario: We assume a homogeneous WSN where each node is energy constrained and is able to send and receive wake-up signals. If a node wants to communicate with any other node, it first sends a wake-up signal and then uses the main radio to exchange data. Depending on the length of the wake-up signal we can control which nodes are woken up with a different granularity as shown before. The shortest signal is a broadcast wake-up, the longest can wake up each node individually. If a node is woken up and afterwards receive a message that was not intended for it, it will discard this message and go back to sleep. Such a false wake-up consumes a certain amount of energy. How many of such false wake-ups occur depends on the density of the network and the length of the wake-up signal. The energy required for sending the wake-up signal depends on the length of the address and mask, on the data rate and on the chip that is used.

The energy E that can be saved for one wake-up by using a SWUR depends on the number of false wake-ups n that are prevented by using a longer wake-up signal and on the energy that is required to send this longer signal:

$$E = \underbrace{(P_{RX} \cdot T_{RX} + P_{\mu C} \cdot T_{ON}) \cdot n}_{\text{energy saved by not waking up}} - \underbrace{P_{TX} \cdot T_{\text{WUR}}}_{\text{wake-up signal}} \ , \qquad (2.1)$$

where P_{RX} and P_{TX} are the power required for receiving and transmitting using the main radio respectively, T_{ON} is the time a falsely woken up node is powered on and uses a power of $P_{\mu C}$. T_{RX} determines the time that a node requires for receiving a message after wake-up. The energy required to send a wake-up signal is given by the transmit power consumption P_{TX} and the duration of the wake-up signal T_{WUR} As the energy for receiving a wake-up signal is orders of magnitude lower than the energy used by the microcontroller and main radio (see Section 2.1) we can ignore it in this analysis.

The WUR that we used for our prototype supports a data rate of 4 Manchester coded symbols per second which corresponds to a symbol duration of 250 µs. The wake-up signal for our prototype consists of multiple parts as depicted in Figure 2.7. The signal starts with a burst of 0.8 ms, followed by a separation bit and a preamble of 3 symbols (additional 0.875 ms). We finally

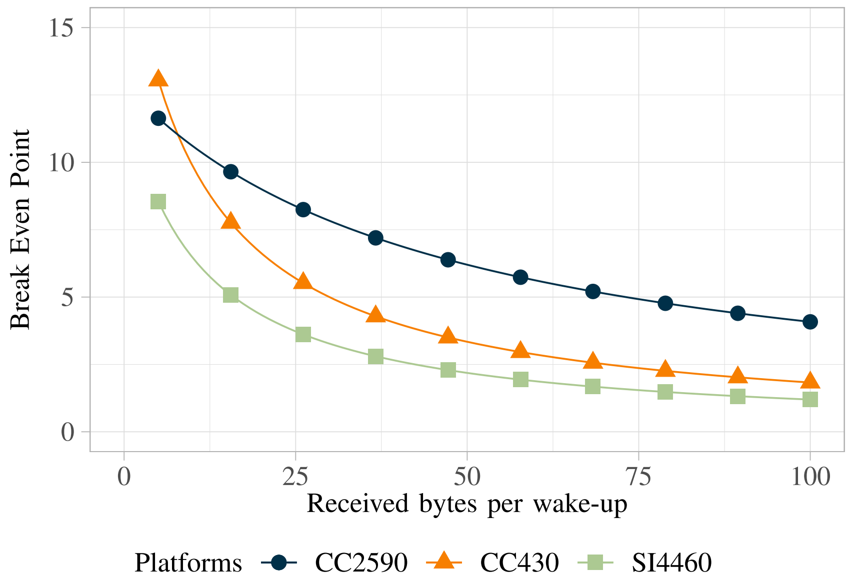


FIGURE 2.10 – Minimal number of prevented wake-ups in the network that are required to save energy (break even point).

have to send 8 bit for the address and mask which takes another 2 ms. In total it takes $T_{\text{WUR}} = 3.675\,\text{ms}$

Following Equation (2.1), the minimum number of false wake-ups that have to be prevented by using a longer wake-up signal in order to save energy in the network (break even point) is:

$$n = \left\lceil \frac{P_{TX} \cdot T_{\text{WUR}}}{P_{RX} \cdot T_{RX} + P_{\mu C} \cdot T_{ON}} \right\rceil \tag{2.2}$$

As an example we use the values from the system in [34] where we used a CC340[15]. This chip consumes $P_{\mu C} = 10.5\,\text{mW}$ in active mode, $P_{TX} = 99\,\text{mW}$ for sending and $P_{RX} = 45\,\text{mW}$ for receiving. The time T_{RX} to receive a data packet depends on the data rate of the main radio and the number of received bytes. Here we assume a data rate of 200 kbit/s and that a node will be active

[15] This is a System On Chip which consists of the low power MSP430 microcontroller and a CC1101 sub-1 GHz transceiver

Table 2.1 – Minimal number of prevented wake-ups required in order to reach the break even point.

Received Bytes per wake-up	2	8	16	32
Number of prevented false wake-ups required	16	10	7	5

for $T_{ON} = 2\,\text{ms}$ if woken up. Depending on the amount of received data per wake-up we can now calculate the minimum number of wake-ups that are required to reach the break even point, i.e., to save energy by sending a longer wake-up signal. Figure 2.10 shows the results of this analysis and a comparison with two other chips. We can see that if we send only very few data the energy loss at other nodes by falsely receiving this data is small compared to the energy that was required to send a wake-up signal. We therefore would have to prevent a larger amount of false wake-ups by using a longer wake-up signal in order to save energy within the network. In our example, if we just send 5 B per wake-up the energy to send the wake-up signal is as much as it would take 13 nodes to receive the data. From the results we can conclude that for scenarios with low density and if only a small amount of data is transmitted per wake-up it would be better to use a wake-up system without addressing but with a shorter wake-up signal. Another option would be to use the proposed broadcast shortcut from Section 2.2.2 or the frame format introduced in Section 5.1.4. This would reduce the energy for sending the wake-up signal and make the addressing scheme more suitable for low-density homogeneous networks.

2.6 Conclusion

In this chapter we have discussed how wake-up receivers can be built and what their strengths and weaknesses are. The main metrics to describe these receivers is the energy consumption, the sensitivity (which determines the range for a given transmit power) and the data rate. In general, the very low power consumption of WURs is achieved by choosing a simple, signal power based receiver with lower sensitivity and data rate than high performance heterodyne receivers.

Another important aspect of wake-up receivers is the capability to encode an address in the wake-up signal to wake up only certain nodes. Already the earliest wake-up receivers implemented such a functionality by encoding the address in the frequency of the wake-up signal. Today, OOK modulation is widely used to transmit an address and nodes can correlate the received data with their own address. We have developed a novel addressing scheme that extends this concept by sending an *address* and a *mask*, allowing the sender of the wake-up signal to flexibly select an addressing scheme (unicast, broadcast or multicast) without reconfiguring the nodes. This is beneficial for many usage scenarios and can be implemented with little effort.

Chapter 3

Wake-Up Receivers in Wireless Sensor Networks

Wireless sensor networks were the first field where Wake-Up Receivers (WURs) have been used to provide energy efficient communication. The main solution to prevent the wasting of energy due to overhearing and idle listening so far was to employ duty cycling schemes where the nodes would switch between inactive times with low power consumption and active times where messages could be exchanged. In this chapter some important duty cycling based Medium Access Controls (MACs) protocols are introduced along with a discussion on the inherent drawbacks and problems caused by their synchronous nature. I will then give an overview on other research in the realm of Wireless Sensor Networks (WSNs) that tries to solve these problems by including the WUR concept to achieve low power communication. Finally, I will describe the work we did in the BATS project where we combined duty cycling and WUR concepts in a very challenging WSN scenario.

3.1 Duty-Cycling based Medium Access in Wireless Sensor Networks

In the last decades a myriad of duty-cycling based MAC protocols have been developed that try reduce the energy consumption while maintaining a good performance [35], [36]. The main performance metrics and optimization goals are typically energy consumption, delay, and fairness. Some algorithms focus on static scenarios, while others also support a dynamic network environment.

The main idea of all protocols is to disable the main radio for a maximum amount of time to save energy. In order to retain connectivity, such a MAC protocol must provide a synchronization method to ensure that sender and receiver are awake at the same time [6]. MAC protocols for WSNs can be classified by different properties like synchronous/asynchronous, communication patterns, adaptivity to changes, etc. One of the first of such energy efficient MAC protocols was S-MAC [37], [38]. This protocol synchronizes the sleep/wake schedules among neighboring nodes during an initial synchronization phase. While this greatly reduces energy consumption, the distributed scheduling can lead to a fragmented network. Also, the delay for multi-hop message passing is very high, which can be improved by an approach called adaptive listening [39]. The T-MAC protocol that was presented in [40] introduced an adaptive active period to better handle load variations, and a future request-to-send (FRTS) to improve multi-hop performance. Another duty cycling based protocol is TRAMA [41] which uses a slotted Time Division Multiple Access (TDMA) based mechanism to avoid collisions in an energy efficient way. The system benefits in terms of power consumption of the receiver, but increases the latency of detection and also implies higher energy consumption of the transmitter.

Another class of MAC protocols are asynchronous protocols. The idea behind these protocols is that most energy is wasted by idle listening. B-MAC therefore employs a low power listening (LPL) approach that periodically samples the wireless channel to detect ongoing transmissions instead of keeping the receiver enabled constantly [42]. When a node wants to send data it first sends a preamble that is eventually detected by the receiving node. The preamble length must be at least as long as the interval in which the nodes check for channel activity. The receiving node can detect this preamble during the LPL phase and then stay active until the actual data message is transmitted. WiseNET [43] improved this scheme by adapting the preamble length based on traffic conditions and by learning the sampling schedule of neighbors. It can be seen as a combination of the asynchronous B-MAC and the synchronous S-MAC protocol. Furthermore, predictive solutions have been studied in the literature and show promising results when it comes to data communication patterns instead of physical contacts. In PW-MAC [44], the sender makes use of a wake-up time algorithm to predict the next wake-up of the receiver. This allows the sender to stay in sleep mode as long as possible and wake-up right before the receiver in order to conserve energy. In general, the asynchronous approach has the advantage of a simpler rendezvous procedure[16] that does

[16] Sender and receiver have to be active at the same time to exchange data, so they have to "meet" in time. Hence the term rendezvous.

not require a common schedule but comes at the cost of a higher energy consumption for sending the long preambles.

3.2 Wake-up Receivers in WSNs

In Section 2.1 we discussed how WURs can be built. In this section we will now have a closer look on how they can be used in WSNs.

An overview of MAC protocols using a WUR is given by Djiroun and Djenouri in [45]. The authors present a taxonomy grouping these protocols into three categories: a) path reservation wake-up MAC, b) non-cycled wake-up MAC and c) duty cycled wake-up MAC. The first category includes protocols that also take routing into account. Category c) includes multi radio protocols like [46] that use two normal radios from which one is used as a (duty cycled) wake-up radio on a different channel. In this work I only consider such solutions wake-up systems that have a secondary ultra-low power radio that is active continuously. Still, the solutions from category c) show an important aspect that is fundamental to the principal of wake-up receivers: *Data transmission and controlling/signaling are two different aspects of MAC protocols with different requirements. It therefore makes sense to use different hardware that is optimized for each aspect.* Other aspects such as routing and cross-layer optimization also have an important influence on the performance and energy consumption of wireless networks. In this thesis, however, I will focus on the Medium Access Control level.

Another categorization of WUR based MACs has been proposed by Piyare et al. [47]. The authors differentiate between a) transmitter initiated, b) receiver initiated, and c) bidirectional protocols. In its simplest form, a *transmitter initiated* wake-up protocol in a WSN works as follows: If a node A wants to communicate with node B it first sends a wake-up signal.[17] The WUR at node B then generates an interrupt that wakes up the microcontroller from its deep sleep state which then enables the main radio to receive data from node A. After a timeout or if no more data needs to be exchanged both nodes go back to an energy preserving sleep mode [48].[18] An example for a *receiver initiated* protocol is presented in Section 3.3.

If the WUR uses another channel than the main radio, the secondary channel can also be used to transmit further control information such as

[17] This could be a simple analog signal, like a busy tone that wakes up all nodes, or can carry addressing information, thus only waking up specific nodes, as illustrated in Section 2.2.

[18] The decision about when to go back to sleep depends on the scenario. Node A could inform node B about any remaining data. Node B may have to stay active to further forward the received data.

which channel to use as shown in [49]. In CMAC, transmissions over the main radio are controlled via *Request* and *Confirm* messages on the wake-up channel that include an agreement on a common channel. The hidden terminal problem is further coped with as a node that is currently receiving data on the main transceiver and gets a request from a third node via the wake-up receiver it signals this node that it is currently busy (via a *Wait* signal). By using a wake-up receiver for asynchronous duty cycling and as a second channel for control messages, CMAC can provide low energy communication without curtailments of latency or throughput. Another integration of the wake-up concept into a low power MAC for sensor networks was presented in [50]. In this paper, a combination of channel assignment, based on a coloring algorithm, together with a wake-up receiver to achieve low power communication was used in a network with static and mobile nodes. The BLITZ protocol from [51] is optimized for event-triggered WSNs that exhibit a sporadic data generation pattern. In such networks nodes generate and forward new data in a non-deterministic way, which make them a perfect fit for wake-up receivers. The protocol combines an asynchronous wake-up scheme with a synchronous dissemination protocol, which can provide low latency and low power consumption at the same time.

3.3 BATS Project

In contrast to traditional wireless networks, WSNs have additional requirements like self-adaptivity, low power consumption and long network lifetime [36], [52]. As discussed in Section 3.1 there are many protocols available that try to achieve one or more of these requirements [35], [37], [41], [43], [44], [49].

We will now have a closer look at the BATS application example that was already introduced in Chapter 1 and discuss why traditional, duty-cycling based protocols are not well suited for this scenario. While we focused on the uplink from the ground nodes to the mobile sensor nodes in the last chapter, we will now have a closer look at the downlink. Figure 3.1 shows the basic scenario that we focus on in this chapter: The mobile nodes regularly send out a wake-up signal followed by their unique ID that is received only by nodes in close proximity.[19] The reception of such beacons is stored as meetings, containing the other node's ID, the start time, and the meeting duration. The collected data must then be send to the ground nodes that are located in

[19] The low sensitivity of the WUR is in this case beneficial, because only signals from bats that are closer than 10 m are received. This allows us to gain information about the neighborhood of the bats.

the forest and can receive the Erasure Code coded data as well as track the flight of the bats by analyzing the transmitted signals. To achieve this the ground nodes regularly emit wake-up signals that are received by the mobile nodes. Upon reception of such a signal the mobile node knows that it is within communication range of a ground node and starts to send the collected data.

This scenario is challenging in multiple ways: First, the time that a bat is in communication range (contact time) is usually very short (less than 3 s) because of their high speed and the complex forest environment [7]. We can therefore not apply known MAC protocols that require any synchronization and signaling. The high mobility renders any schedule that could be agreed upon obsolete after a very short time. Second, the weight restriction of 2 g forces us to use a lithium coin cell battery which is light and has a high energy density, but cannot supply a sufficiently high current to power the microcontroller and radio transceiver. We therefore first have to charge a capacitor with 330 µF and then power the system from this capacitor [34]. This severely limits the maximum active time because the capacitor can only store enough energy to power the system for 12 ms. After that, the node has to go back to sleep mode to recharge the capacitor as illustrated in Figure 3.2.

In the following sections we will discuss how we can still reliably send the collected meeting data from the mobile nodes to the ground nodes. We will employ a combination of duty cycling with a wake-up receiver to provide a robust downlink connection without complex synchronization and in a very challenging scenario.

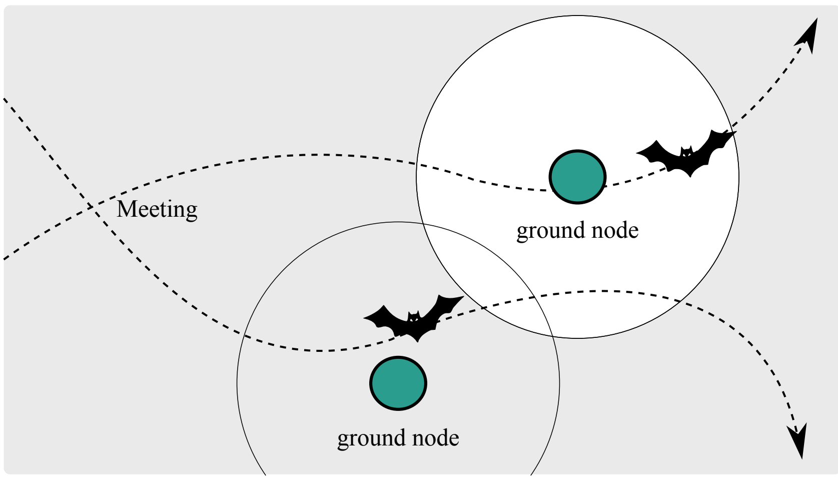


FIGURE 3.1 – BATS scenario: The bats (*Myotis myotis*) are equipped with lightweight wireless sensor nodes that can detect meetings and send the collected data to ground nodes.

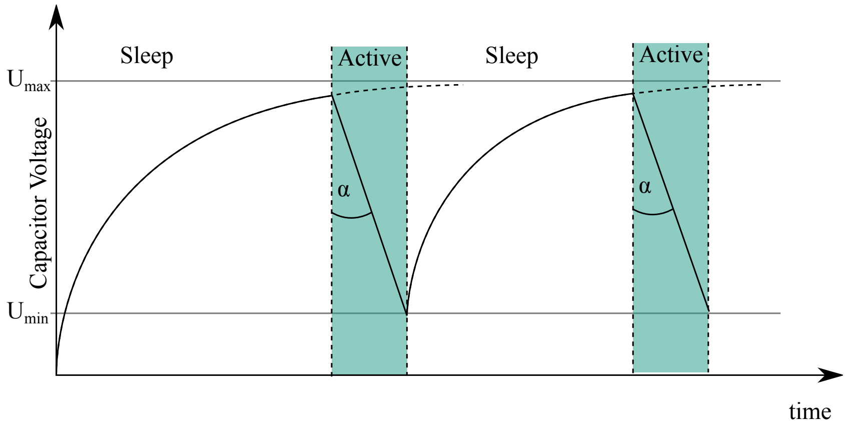


FIGURE 3.2 – Qualitative overview of charging and discharging of the capacitor.

3.4 MAC Protocol Options

Existing MAC protocols for WSNs are often designed for static scenarios where the nodes are at a fixed location. In such scenarios we can use duty cycling with schedules that are negotiated between the nodes and stay more or less fixed during the lifetime of the network.[20] Even protocols that are designed to support network changes and mobility of nodes as known from the realm of Mobile Ad Hoc Networks are not suited for the very low contact times and high variability of the BATS scenario. Together with the tight energy constraints and the maximum active time forced by the capacitor we have to fall back to very simple protocols: random access (*ALOHA*), slotted random access (*Slotted ALOHA*) and Time Division Multiple Access (*TDMA*). We also use the WUR of the nodes to provide a synchronization method within the network.

The protocols we evaluated are based on the following principle: First, the mobile nodes on the bats need to detect when they are in communication range of the ground nodes so that they can transmit the collected data. To achieve this, the ground nodes that are located in the observation area broadcast periodic wake-up signals (beacons). If a bat now enters this area it will start to receive theses beacons and will start to send the collected data to the ground nodes. If all nodes would start transmitting immediately after the reception of a beacon, it would create a synchronized collision. Instead, we use the reception time as a reference for all nodes in communication range that we can use to schedule the downlink transmissions. We denote the time that the

[20] Some MAC protocols like [53] also support dynamic schedules to better support different load scenarios.

wake-up signal is received by the nodes as t_0.[21] We call the time between two beacons a *superframe* with a duration of t_{SF}. For the following simulations we assume a beacon interval of $t_{\mathrm{SF}} = 100\,\mathrm{ms}$ which corresponds to a wake-up frequency of 10 Hz. Within this superframe we now evaluate the performance of the three MAC protocols ALOHA, slotted ALOHA, and TDMA.

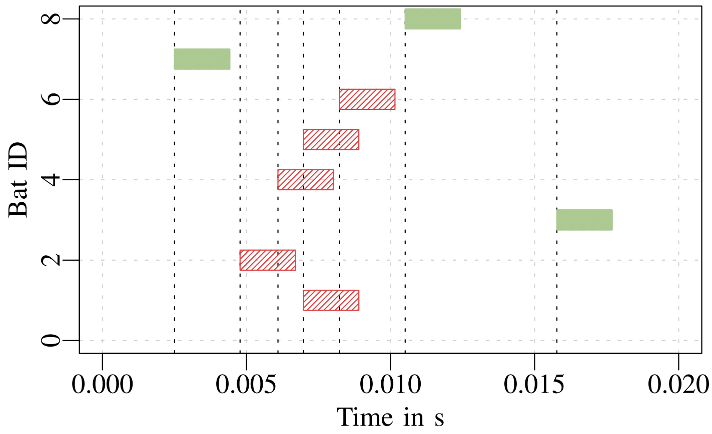
The simplest variant is ALOHA, where all nodes select a random time to transmit from the interval $[0, t_{\mathrm{SF}} - t_{\mathrm{d}}]$, where t_{d} is the duration of one packet. This protocol is known to have a high collision probability as shown in Figure 3.3a. The figure shows the start and end point for the random transmissions of eight bats after the reception of a wake-up beacon at time $t = 0$. Red squares indicate a failed transmission due to a collision, green squares indicate a successful transmission. The problem of the simple ALOHA variant is that a packet that is transmitted at time t_a can collide with all packets that are sent in the interval $[t_a - t_d, t_a + t_d]$

This length of the interval where collisions can occur can be reduced by adopting a slotted variant of the ALOHA protocol. Here we divide the superframe into multiple, discrete sub frames, called slots. Instead of choosing a random transmission time, each node selects a random transmission *slot* as illustrated in Figure 3.3b. Collisions are still possible if two nodes chose the same slot but the probability is reduced.

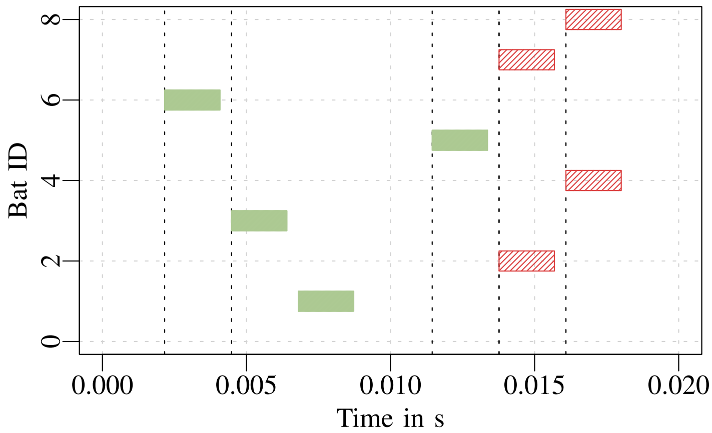
The third variant, TDMA, is shown in Figure 3.3c. For this variant we also divide each superframe into s slots. The assignment of nodes to slots is based on the unique ID of each node. We assume that each node knows its own ID and the number of available slots and can therefore choose the transmission slot ID mod s. Assume we have $s = 8$ slots and $n = 12$ mobile nodes, then nodes 0–8 would use slot number 0, nodes 1 and 9 would use slot number 1 and so on. As long as there are no nodes with the same slot in communication range at the same time, this scheme can prevent any collisions. The selection of a suitable value for s depends on the number of nodes that are within the same collision domain. Due to the high mobility in this scenario this is lower than the total amount of nodes in the network (see Section 3.5).

In our system collisions among packets of different nodes are not the only reason why a transmission can fail. As described before, the system is powered by a capacitor that needs to be recharged between transmissions. In the two random protocols it can happen that a node selects a transmission time/slot at the end of a superframe and then immediately at the beginning of the next superframe. In such a case the time between two transmissions can be too

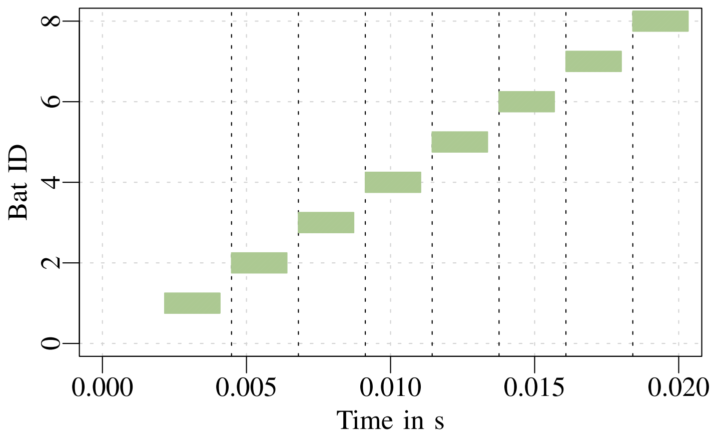
[21] The difference in reception time caused by the different distances of the mobile nodes to the ground node is so small that it can be safely ignored here



(a) ALOHA protocol option



(b) Slotted ALOHA protocol option



(c) TDMA protocol option

FIGURE 3.3 – Example of received frames within a superframe.

small to sufficiently recharge the capacitor and the second transmission will fail because the voltage will drop below the minimum voltage level.

3.5 Evaluation

To evaluate the performance of the three proposed protocols options, we developed a detailed simulation model that includes all relevant aspects of the BATS system. This includes a mobility model to simulate the movement of the bats in a realistic way, the transmission and reception of wake-up and normal data signals and the charging and discharging of the capacitor to also capture the effects of insufficient energy. We analyzed the performance of the protocols in terms of Packet Delivery Ratio (PDR) and energy consumption under different scenario parameters.

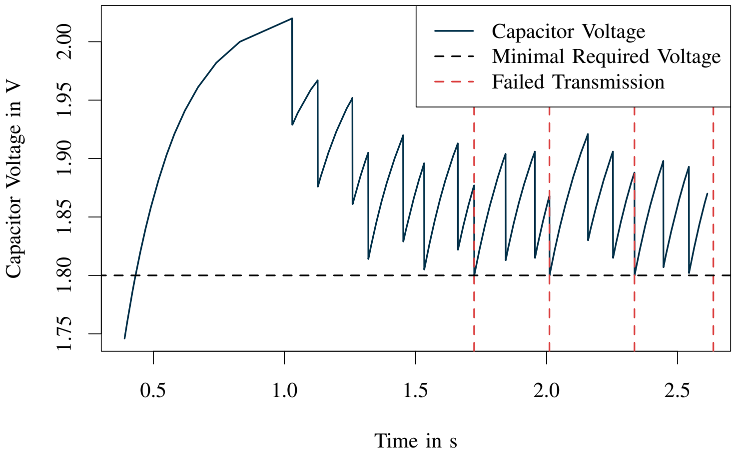
3.5.1 Simulation Model

The simulation model is based on the OMNeT++ network simulator [32] using the MiXiM framework [33] to model the radio communication and MAC protocols. The scenario as shown in Figure 3.1 contains 25 ground nodes deployed in an irregular grid with a inter-node distance of 50 m, distributed over an area of 300 m × 300 m. In the scenario we also simulate a number of mobile nodes attached to bats. The movement of the mobile nodes has an important impact on the simulation results and should resemble the movement of the bats as realistic as possible. We used an adapted Lévi flight model from [7] that models the foraging behavior of bats. In this model the bats first fly to one of the nine hunting areas where they then catch prey, fly in a circular pattern to eat and then go back to hunt. The wireless channel is modeled using a free-space path loss propagation model and log-normal shadowing over an 868 MHz channel.

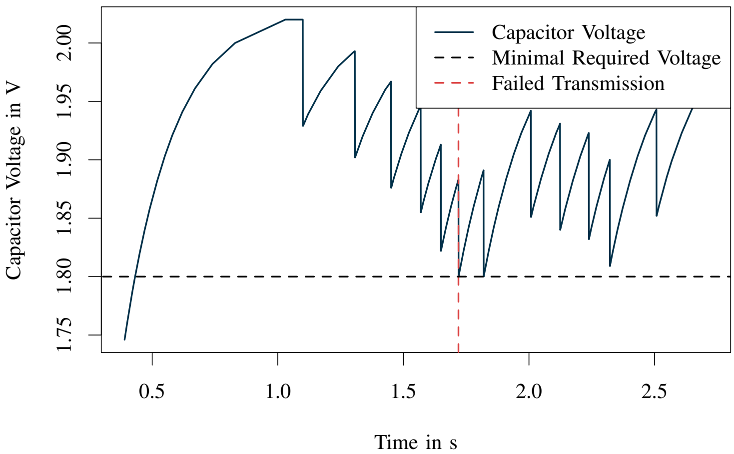
3.5.2 Energy Model Validation

The microcontroller and radio transceiver of our system are powered by a capacitor that first has to be charged, which can cause transmissions to fail as described before. Therefore we also included an energy model that resembles this charging and discharging behavior. The different protocol options show a different susceptibility to failed transmissions caused by an insufficient capacitor charge as illustrated by Figure 3.4.

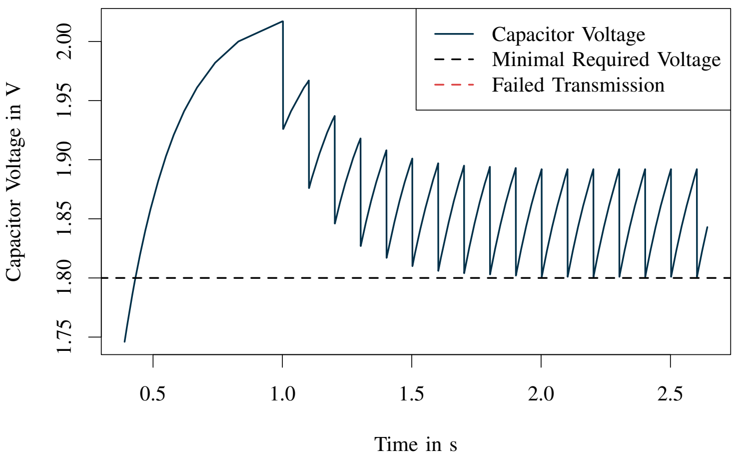
The ALOHA protocol option selects a random transmission time after the reception of each wake-up. This results in an irregular charging/discharging



(a) ALOHA protocol option: Due to the irregular inter-packet times four transmissions fail because of an insufficient capacitor load



(b) Slotted ALOHA protocol option: In one case the transmission fails because of an insufficient capacitor load



(c) TDMA protocol option: All transmissions are successful. If the battery voltage was even lower, no packets could be send anymore

FIGURE 3.4 – Voltage of the buffer capacitor over time.

pattern shown in Figure 3.4a. The figure shows the voltage across the capacitor over time.[22] After each transmission the node goes back to sleep mode and recharges its capacitor. If the time between two consecutive transmissions is not long enough, the second transmission will fail because the voltage drops below the minimum voltage level which causes the System On Chip to switch off. Such failed transmissions are depicted by a red, dashed line in the figure. Also, the slotted ALOHA protocol option can lead to failed transmissions as shown in Figure 3.4b, even though this problem is not as pronounced as in the time continuous version. Figure 3.4c shows the voltage when using the TDMA protocol option. Since each node selects the same slot for each transmission, the pattern shows the regular wake-up interval of 100 ms. In this example there were no failed transmissions because the capacitor could always be charged sufficiently between two transmissions and the voltage never dropped below the threshold of 1.8 V.

The problem of insufficient charge of the capacitor is especially important if the nodes reach the end of their lifetime when the battery voltage gets low. At some point the battery will not be able to recharge the capacitor sufficiently within the 100 ms interval, which would cause all transmissions to fail with the TDMA variant. With the random protocols some packets could still be transmitted successfully if the time between two transmissions is long enough.

3.5.3 Protocol Performance

The following results are all based on simulations with a fixed amount of 25 ground nodes at fixed positions. We then ran the simulations with a varying number of bats (1, 8, 32, 64, 128) and number of time slots per superframe. To ensure fairness between the different variants, all protocols had the same amount of time for sending in each superframe. The simulation starts with the bats moving to different hunting grounds and following the movement given by the model. This means that the number of bats that are in communication range to a certain ground node can differ and is usually lower than the number of available time slots.

On our real hardware sensor nodes we first collect and aggregate a certain amount of data before sending packets to the ground nodes. Since we want to analyze the performance of the different MAC protocols here, we assume in the simulation that each node always has data to send if it detects a wake-up beacon. We used different packet sizes (12 B and 48 B) that have an influence especially for the ALOHA variant because longer packets are more susceptible to collisions than shorter ones.

[22] The voltage V of a capacitor is directly related to the stored energy $E = \frac{1}{2} \cdot C \cdot V^2$

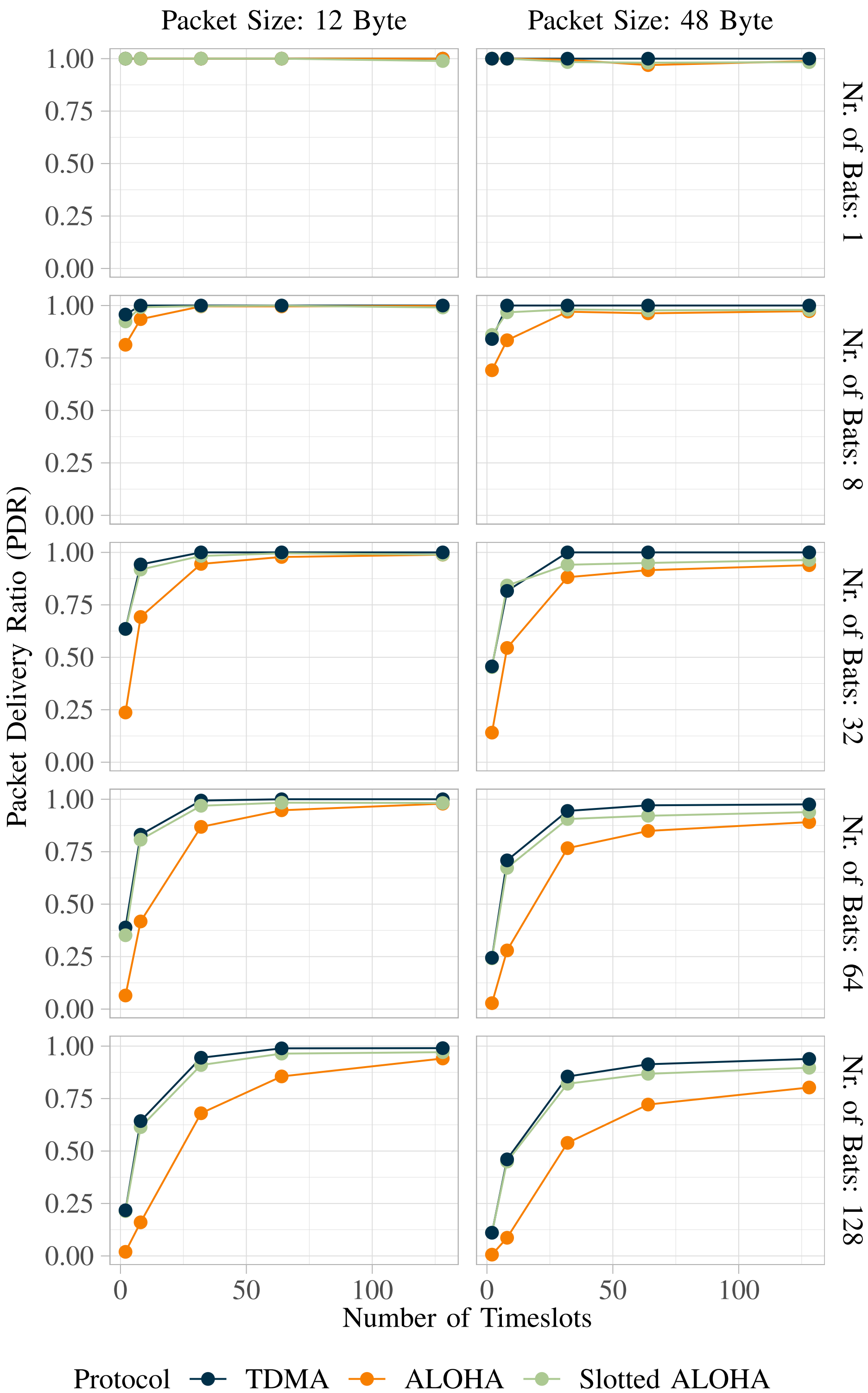


FIGURE 3.5 – PDR of different protocol options as a function of number of time slots and the total number of bats in the scenario.

The results of our simulations are shown in Figure 3.5. The figure shows the Packet Delivery Ratio (PDR) for different number of time slots per super frame and different number of bats in the scenario. The left part shows the results for packet sizes of 12 B, the right part for 48 B. With just a single bat there are no collisions and all protocols can deliver nearly all packets. Only the ALOHA variants fail to transmit a few packets due to the recharging problem discussed before. If we now increase the number of bats, the PDR decreases for the simulation runs where not enough slots where available because in this case collisions can occur. It is clear to see that the slotted variants perform better than pure ALOHA. With many bats in the scenario the probability for collisions increases which leads to a lower PDR. For the simulation runs with packets of 48 B this behavior is even more pronounced. Interestingly the slotted ALOHA variant performs nearly as good as the TDMA variant. That means that in our scenario the probability that two nodes with the same slot in the TDMA variant are present at the same ground node is roughly as high as the probability that two random nodes choose the same slot.

3.5.4 Energy Efficiency

Another important metric in WSNs is the energy efficiency of the communication protocols. All three variants are based on the wake-up receiver providing

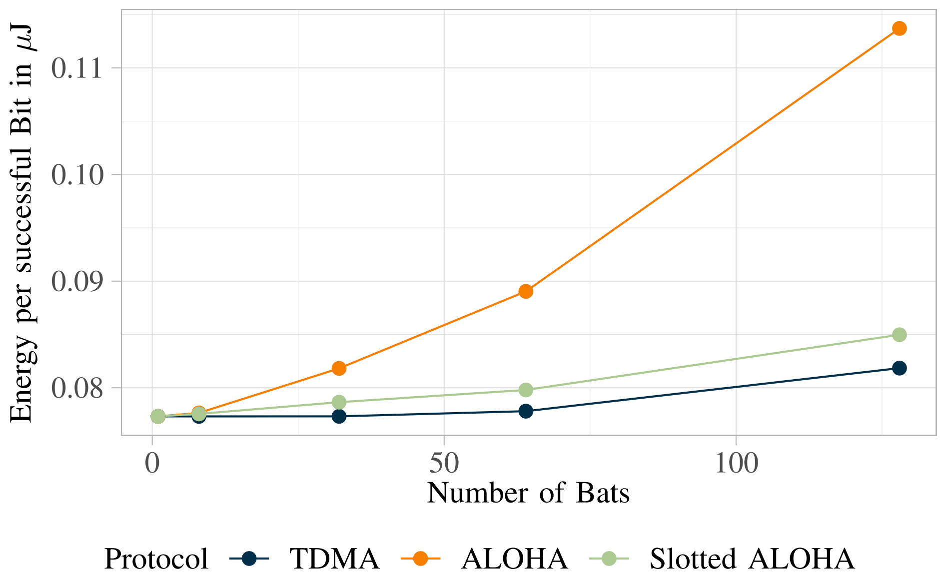


FIGURE 3.6 – Required energy to successfully send a bit for an increasing number of bats. The lower the value, the more efficiently a protocol uses the available energy.

a common synchronization after which each node tries to transmit a packet. That implies that in all variants each node consumes the same amount of energy per superframe. The energy efficiency (the energy required to successfully transmit a Bit) therefore only depends on the amount of successfully transmitted Bytes. Figure 3.6 shows the energy efficiency for the simulation run with a packet size of 12 B and 32 time slots which are representative parameters for our application scenario. The graph shows that the pure ALOHA variant is less efficient than the slotted options because it wastes a lot of energy on failed transmissions. For each point in the graph the amount of transmitted bits is the same. The difference in efficiency therefore is only induced by failed transmissions.

3.6 Conclusion

Modern Wireless Sensor Networks enable researchers to monitor environments – in our case even small animals like bats – with sensor nodes that can exchange data over a wireless link. One of the main requirements for this to work is energy efficiency, because usually such sensor nodes are powered by a limited battery. The duty cycling protocols that have been developed in the past two decades already greatly reduced energy waste caused by idle listening and overhearing but at the cost of reduced throughput and increased delays. Also they require some sort of synchronization among the nodes which adds additional overhead in terms of complexity and energy consumption. The wake-up receiver approach can mitigate these performance impairments by enabling nodes to be woken up remotely just before a transmission in an energy efficient way. This technology has a great potential to further reduce the size of wireless sensor nodes and prolonging the lifetime of such decentralized networks. For the BATS project, which has very tight energy constraints and a challenging, fast changing network topology, we combined these two approaches to provide a reliable and energy efficient downlink for the mobile nodes. We use wake-up receivers to provide a centralized synchronization, which then allows us to use simple MAC protocols to coordinate the transmissions of the mobile node. This way we can achieve a reliable transfer of the collected data to the ground nodes, despite the short contact times and the tight energy constraints. The protocol options TDMA and slotted ALOHA are well suited for our application scenario, given that sufficient slots are allocated per superframe.

Chapter 4

Wake-Up based Wireless LAN

While there has been much interest in Wake-Up Receivers (WURs) in the realm of Wireless Sensor Networks (WSNs), there are also other fields that can benefit from the low-power and asynchronous power saving that WURs provide. Concepts like the Internet Of Things (IoT) differ from WSNs in their network topology, application scenarios, self-organization and underlying communication technology, but they share the need for energy efficient communication protocols. In a typical IoT scenario many devices with sensors/actors are connected to a wireless network (i.e., Wireless LAN (WLAN)) [54]. These devices can be operated by battery (like temperature sensors in a smart home) and therefore need a way to communicate with very little energy in order to provide a long lifetime. The requirements for Medium Access Control (MAC) protocols in such a scenario are different from MACs in WSNs because the devices do not have to form a decentralized ad-hoc network but can connect to a common infrastructure provided by access points or base stations (star topology). There are solutions for IoT available that provide low data rates and low power consumption. For long range communication (Low Power Wide Area Networks (LPWAN)) technologies like LoRa or SigFox offer long communication ranges of many kilometers but with very low data rates [55]. For short range communication popular technologies are the IEEE 802.15.4 based Zigbee or Bluetooth Low Energy (BLE). The drawback of these technologies is that they support only low data rates and are therefore not suited for applications with higher demands on throughput. WLAN on the other hand offers high data rates and is already available in most households, offices, factories and even public places. However, it is too power hungry for battery operated IoT devices as it relies on a duty cycling approach that requires the Stations (STAs) to wake up periodically. Also, the high performance Orthogonal Frequency-Division

Multiplexing (OFDM) based radio consumes more power than radios with a simpler modulation scheme.

Wake-Up Receivers have shown a great potential in WSNs. Therefore, it seems reasonable to investigate whether they can, mutatis mutandis, also be used in conjunction with WLAN to achieve similar improvements in power consumption and delay. While the basic principal is the same (devices are equipped with an additional WUR and can be woken up before exchanging data over the main radio), such systems differ from WSNs in the following aspects:

1. **Network Topology:** WLAN is usually deployed in a centralized topology with one Access Point (AP) serving multiple STAs (Basic Service Set (BSS) mode),

2. **Homogeneous Network:** The network is used by low power, low data rate IoT devices as well as by multimedia devices like smart TVs or smartphones with high data rate demands,

3. **Energy:** We assume that the AP is not energy constrained, the energy saving is therefore only done at the STA side,

4. **Compatibility:** Any changes to the MAC or Physical Layer (PHY) must be compatible to existing systems.

In the following chapter I will explain how we included wake-up receivers into a WLAN system and show how such a system behaves through measurements of a real world hardware prototype and through an extensive simulation study with many STAs and varying traffic loads.

4.1 Energy Saving in WLAN

Before looking at the new power saving approach using wake-up receivers we will first revisit the current state-of the art and describe how the normal power save mode in WLAN works. I will also give an overview on other research that has been published in this field.

4.1.1 Power Saving in IEEE 802.11

The main focus of power saving techniques for WLAN is mainly on saving energy at the mobile STAs that are usually powered by a battery with limited energy. In contrast, we assume that APs are connected to a mains supply and are therefore not energy constrained (it is, however, still possible to also save energy on the AP side, but that is not the focus of this work). Even though energy consumption was not the main optimization criterion during the design of IEEE 801.11, the initial standard already includes a native power saving scheme [56]. This initial approach has then been extended (especially in the IEEE 802.11e amendment) to improve the performance of real-time and delay sensitive applications like Voice Over IP (VoIP) or streaming by integrating Quality Of Service (QoS) techniques.

The normal power save mode defined in the IEEE 802.11 standard [56] is based on a duty-cycling technique. We focus here on the infrastructure mode (BSS), where multiple STAs are connected to a central AP. In this mode, the AP will always be powered on and power saving only happens on the mobile STAs.

With each IEEE 802.11 MAC frame a header containing a frame control field is transmitted, which is used to provide general information about the frame such as version, type, and subtype of the frame. For the power saving technique the `power management` flag and the `more data` flag are crucial. The `power management` flag is used by a STA to inform the AP that it will enter or leave the power save mode.

The standard defines two power modes for a STA in a BSS network. In the *Awake* state the STA is fully powered and has its main radio activated. In the *Doze* state it consumes little power but is also not able to transmit or receive any data. To keep the STA in sync with the network, the STA has to inform the AP that it will enter sleep mode, by sending a so called *Null frame*. A Null frame is a data frame without any data payload. Only the Power Save (PS) flag in the frame control field of the MAC header is of interest here. After this null frame has been acknowledged by the AP, the STA is allowed to enter doze mode. The AP will now start to buffer all packets that are addressed to the sleeping STA. Figure 4.1 shows the frames exchanged between STA and AP. The solid line indicates that the STA is awake, the dashed line indicates that it is in power save mode.

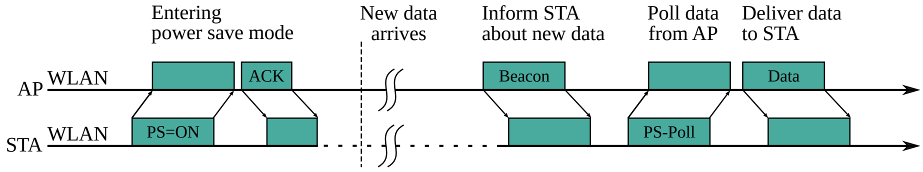


FIGURE 4.1 – Packet exchange using normal power save mode with polling. The dashed line indicates that the WLAN radio is in sleep mode.

The AP regularly sends out so called beacons that are also used to inform sleeping STAs about new buffered unicast packets. Beacons can include a so called Traffic Indication Map (TIM), which is a partial virtual bit map that encodes which STAs have new packets waiting at the AP. The STA has to wake up regularly and receive the beacons in order to learn about possible buffered data. Therefore the beacon interval significantly influences the behavior and the performance of the system. Following the standard, a STA that was informed about new data can then request each of the buffered packets successively from the AP by sending PS-Poll frames. In each packet send to the STA the AP will set the *more data* flag to 1 if there are more packets buffered. With the last packet this field will be set to 0 and the STA can go back to sleep.

Pending multi- and broadcast traffic is announced in the Delivery Traffic Indication Map (DTIM) that is included in a subset of the beacons. The next beacon including a DTIM is determined by the DTIM count and interval that is transmitted with every beacon. To receive buffered multicast/broadcast traffic all sleeping STAs should at least wake up for those DTIMs beacons

As the polling technique using PS-Poll frames is inefficient and has a very negative impact on the throughput [57], the Linux software MAC framework `mac80211` implements a different protocol. The signaling via Null frames and the TIM remains the same but the way pending packets are fetched from the AP is different: Instead of polling each frame, the STA leaves the power save mode if the AP has indicated that data is buffered via the TIM. It does so by sending a Null frame with the power mode flag disabled. The AP then sends all buffered frames immediately. This consumes more power if the STA does not go to sleep again immediately but prevents the degradation in throughput of the polling approach.

The native power save mode, however, is not well suited for applications with QoS requirements. Therefore in IEEE 802.11e additional Power Save Modes (PSMs) where introduced to reduce the latency for such applications. These additional modes are called Automatic Power Save Delivery (APSD). The Unscheduled Automatic Power Save Delivery (UAPSD) mode is well suited for delay-sensitive, highly regular, bidirectional traffic like VoIP where data is sent and received in a periodic pattern. In this mode the AP does not have to wait until a STA polls buffered data using a PS-poll packet. Instead it directly transmits buffered packets if it receives a data packet from the STA, assuming the STA must be currently in active mode. In scenarios, where the inter-arrival time of packets is known in advance (e.g. a stream with Constant Bit Rate (CBR)), the Scheduled Automatic Power Save Delivery (SAPSD) mode can schedule the wake-up times accordingly without the need of signaling via beacons.

In comparison to the native PSM, these APSD modes can significantly reduce the delay and are therefore very important for timing- and delay sensitive (real time) applications [58]. However, they can only be used if the communication is bidirectional (UAPSD) or if the inter-arrival time of packets is known in advance (SAPSD). For other traffic that is more sporadic and irregular none of the existing PS modes provide a solution with low power consumption *and* low delay.

The wake-up based system described in the following sections can overcome the shortcomings of existing solutions and provide low power consumption and low delay for a wide range of applications. It does not depend on the choice of parameters or prior knowledge of the traffic patterns and is therefore suitable for irregular, low data traffic of IoT devices, regular, delay sensitive traffic like VoIP as well as high throughput applications like video streaming.

4.1.2 Related Research on Power Saving in WLAN

The performance of power saving techniques, the influence of different parameters and the source of energy consumption has been studied by many researchers. In [59] the authors give an overview of different aspects of energy consumption and sources of wasted energy in WLAN and of improvements that can decrease energy consumption. Possible improvements aim at the reduction of energy consumption during the contention period and for transmission and re-transmission of packets, and on improvements to the native PSM. The influence of the PSM on QoS mechanisms is studied in [57]. An important observation in this work is that the polling mechanism as described in the previous section has a negative impact on the achievable throughput. Also the *beacon interval* has a significant impact on the required energy and the downlink delay, as we will confirm in our evaluation in Section 4.3. This *inherent tradeoff between power saving and performance* is well known and has already been described in 1998 when WLAN was rather new: For ad-hoc networks, for example, the influence of the beacon interval and the Announcement Traffic Indication Message (ATIM) window size has been analyzed in [60]. In such ad-hoc WLAN networks higher beacon rates (meaning that STAs have to wake up more often) can actually reduce the energy consumption when the network load increases because the probability of collisions is reduced.

Even though the tradeoff between performance and energy consumption cannot be circumvented with current duty-cycling based systems, there is certainly room for improvement. For example, the static nature of the duty-cycle can be made more dynamic to adapt to varying traffic loads. In [61], the authors propose a way of improving the PSM in ad-hoc WLANs networks. In such scenarios there is no central AP which is why every node could receive data from all other nodes. Instead of using beacons to signal new data, in the ad-hoc mode each node stays active for an ATIM window. During this window each STA can announce pending packets. The size of this window has a significant impact on energy consumption and performance which is why the authors of [61] suggest to estimate the current network load and adapt the ATIM window size accordingly. This can greatly reduce the time during which nodes have to stay awake and, therefore, reduce overall power consumption. A similar approach is used in [62], where the authors present an algorithm to dynamically adjust the power saving mechanisms depending on the QoS requirements. With such adaptive approaches, the energy saving vs. performance tradeoff can be adjusted.

In WLAN the distributed channel access scheme (Distributed Coordination Function (DCF)) is rather inefficient in terms of energy consumption as it relies on channel sensing which consumes a lot of energy. Therefore in [63], an optimization of the channel access scheme is proposed. If a STA tries to get access to the channel it first listens for other ongoing transmissions and goes into backoff if it finds the channel busy. The authors suggest to keep the STA's radio in sleep mode during this backoff period to save energy. Depending on the higher layer protocols and the load condition in the network this can reduce the energy consumption by 28–80 % . Another technique to reduce the energy consumption caused by sensing the channel, called energy-efficient multi-polling scheme, is proposed in [64]. In order to reduce the energy required for channel access the AP can announce a fixed wake-up schedule for the STAs which allows contention free access, thus reducing the need for idle listening and overhearing.

In most scenarios only the communication between the AP and a STA in PSM is considered. However, the authors in [65] show that background traffic to other non-sleeping STAs can significantly increase the energy consumption for other nodes in the network. This is caused by the prolonged time that a STA has to wait from leaving the PSM mode until the delivery of the buffered packets. Also, if many sleeping STAs are informed about buffered packets with the same beacon, they will compete for the channel at the same time which also increases energy consumption (this problem of causing a coordinated collision can also be seen in Section 5.3.2). To solve these problems, the authors propose to:

- prioritize packets for STAs that just woke up from sleep mode in order to reduce the time the STA has to stay awake. Fairness towards non-sleeping STAs is achieved by postponing the notification via beacons accordingly.

- Use virtual APs for each STA to spread the delivery of beacons, thus avoiding the coordinated collisions.

A similar idea for for a multi-AP scenario is discussed in [66], where the authors propose to spread the beacons of different APs evenly in order to avoid contention between the STAs. These techniques can significantly reduce the energy consumption that is caused by concurrent transmissions and the resulting costs induced by the distributed channel access of WLAN by distributing the traffic more evenly in time.

The concept of reducing the active times of the main radio can also be applied at a smaller time scale. If STAs start to receive data that is not intended for them they can deactivate their radios to save energy as proposed in [67] and [68]. In a naïve approach a STA would receive and decode a packet, check if the destination address matches its own and drop the packet if this is not the case. It is, however, sufficient to only decode the header of a packet that contains the destination address and stop receiving if it does not match. In this case the STA can just update its Network Allocation Vector (NAV) that indicates how long the channel will be busy and immediately go to sleep. In [69], the authors propose a power saving technique that is similar to the one presented here. Instead of adding an additional wake-up receiver, the authors propose to reduce the sampling clock rate of the main radio. As with wake-up based operation the normal WLAN frames then have to be prepended with a special preamble that can be detected by the downclocked radio. This preamble can also contain an address. The advantage of the system in [69] is that no additional hardware is required.

One of the earliest works that considered a secondary, low-power channel for 802.11b WLAN was already published in 2002 by Shih et al. [70]. In this paper the authors propose to physically separate the control channel from the data channel. The prototype presented was based on Personal Digital Assistants (PDAs), the successors of today's smartphones, which where equipped with an external, low-power Amplitude-Shift Keying (ASK) based transceiver. While the main idea of this paper (a secondary, low-power radio) is similar to the system based on wake-up receivers, there are some major differences in technology, system architecture and depth of analysis. The transceiver used in this paper still had to be duty-cyled, because it draws 7 mW in receive mode and 2 mW in sleep mode. Since then, the technological advances in the design of Integrated Circuits (ICs) made it possible to create wake-up receivers that only need a couple of µW (see Section 2.1). Also, the system architecture differs from the architecture presented here. The system in [70] handles WLAN and wake-up as two different networks. A central server can send the wake-up command via additional proxies if it receives a communication request for a sleeping PDA, which takes 5–10 s. The wake-up based WLAN presented here does not need a central server. Instead, each WLAN AP can send out the wake-up signal which only takes ms.

4.2 System Architecture

Existing PSMs that rely on duty cycling have some inherent drawbacks that can be solved by using wake-up receivers. In this section I will describe how we included wake-up receivers into a WLAN system, introduce the new power save mode and the prototype we build to test our concept.

4.2.1 Wake-Up Enabled WLAN

To include the wake-up concept into WLAN we equipped the AP with a Wake-Up Transmitter (WUT) and the STAs with Wake-Up Receivers. When a station now enters power save mode, it will follow a communication protocol as shown in Figure 4.2, that is slightly different from the normal PSM. As in the normal PSM the AP has to be informed by the STA about the mode change using a Null frame. The STA can then turn off its main radio to save energy, the WUR, however, stays active all the time and consumes only little power. The AP will now also buffer incoming packets for the sleeping STA, but now it does not have to wait for the next beacon to signal pending packets to the STA. It can immediately (after gaining access to the channel using the DCF first) send a wake-up signal to the STA. The WUR will receive the signal, wake up the STA which can then retrieve the new data either by using the polling approach or by changing to the active mode. It is therefore not necessary anymore to enable the radio periodically to receive the beacons which can save a lot of energy as we will show in Section 4.3.3.

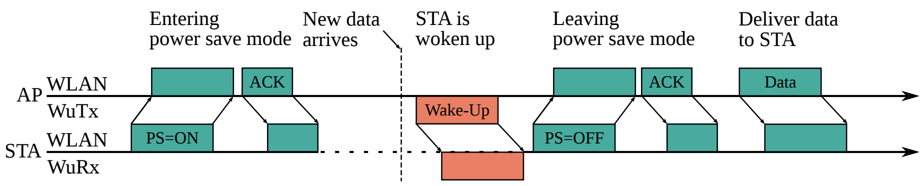


FIGURE 4.2 – Packet exchange using the WUR system. Instead of waiting for the next Beacon, the STA is woken up immediately. The dashed line indicates that the main transceiver is in sleep mode.

For this protocol to work we must assume that the sensitivity (ergo the communication range) of the Wake-Up Receiver is equal to the one of the main radio. This is a strong assumption and as of today WURs with such a high sensitivity are not yet available on the market. However, researchers have made substantial progress in the last years (i.e. in [71]) which shows that this assumption can actually be met in the future. For our prototype we used a WUR implementation from our previous work in [9], that uses different frequencies for the wake-up signal (868 MHz) and for the main radio (2.4 GHz). In our simulation model used in Section 4.4 we also investigated a single band solution to show that the results are also valid in such a scenario.

4.2.2 Hardware Prototype

To show the feasibility of our approach and to gather crucial parameters for our simulation model we built a prototype using off the shelf hardware. We equipped normal WLAN hardware with an additional WUT and WUR and adapted the firmware and drivers accordingly. The general architecture of the prototype is depicted in Figure 4.3.

As WLAN hardware we used a USB WLAN dongle (Netgear WNDA3200) with an Atheros ath9k-based chipset. The software based driver in the Linux kernel (*mac80211*) allows us to use and modify the power save mode of this WLAN chip. The access point uses the standard hostapd software to provide the AP functionality. The Wake-Up Transmitter consists of an ESP32 microcontroller that gets the wake-up address of the receiver via a Universal Asynchronous Receiver Transmitter (UART) connection from the WLAN dongle. It then sends out the wake-up signal that is compatible to the format required by the AS3933 chip (see Figure 2.7) using a commodity 868 MHz transmitter (HPD8407F–868S).

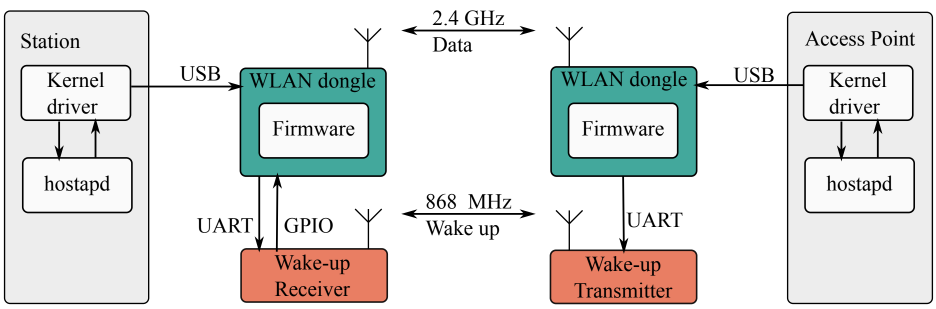


FIGURE 4.3 – Block diagram of our prototype with wake-up extension

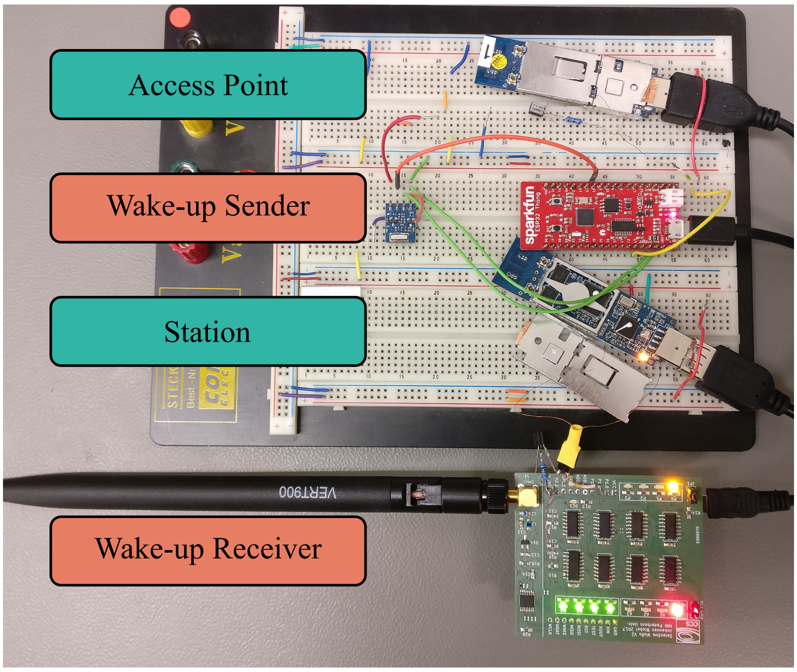


FIGURE 4.4 – Hardware prototype with two WLAN dongles and a wake-up sender and receiver.

The STA uses the same USB dongle that is connected to a computer running Linux. One of the General Purpose Input/Output (GPIO) pins of the WLAN chip is connected to the wake-up receiver from Section 2.3 which consists of an AS3933 wake-up receiver and a microcontroller (MSP430–G2452IN20). Using a UART connection the STA can configure the local address of the WUR. Since we have to rely here on the AS3933 based prototype the wake-up signal only supports a very low data rate of 4 kbit/s which is much lower than the data rate suggested in the current IEEE 802.11ba standard (up to 250 kbit/s [72]). Figure 4.4 shows the complete prototype including the WLAN dongles and the WUT and WUR.

The driver and firmware where adapted to implement the new Power Save Mode. After the STA has successfully connected to the AP it is assigned an Association ID (AID). The AID is then used as the address pattern for the WUR (see Section 2.2). The STA can then go into sleep mode using the protocol described in Section 4.1.1. The AP still sends out the beacons for legacy STAs but the wake-up enabled STA does not wake up periodically to receive them. If new data for the sleeping STA is now received at the AP it will be buffered and a wake-up signal is send out. For this the AID of the intended receiver is send to the WUT via the UART connection which then generates the On-Off-Keying

(OOK) signal. This signal is received and decoded by the WUR which generates an interrupt. This interrupt is detected by the firmware of the USB dongle which then forwards this information to the driver. The driver then sends out a Null frame to the AP to leave the sleep mode as shown in Figure 4.2 and can then receive the pending packets.

4.2.3 Prototype Validation

Figure 4.5 shows the measurements at four measurement points of the prototype during sending and receiving of a wake-up signal. Before the measurement the STA entered sleep mode using the WUR protocol. We then send one unicast User Datagram Protocol (UDP) packet from the AP to the STA.

Figure 4.5 shows how the system reacts after the UDP packet was received:

1. The AP sends the AID of the STA to the Wake-Up Transmitter (labeled 'UART in V').

2. The ESP32 generates the wake-up signal (including preceding burst and pattern), which is transmitted by the OOK transmitter (labeled 'Wake-up Signal in V').

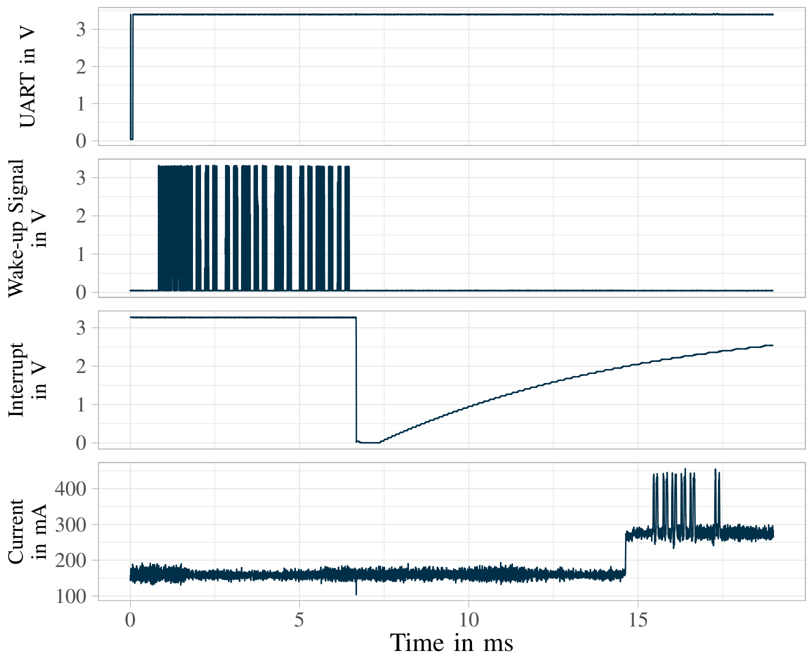


FIGURE 4.5 – System validation measurements. After receiving the wake-up signal, an interrupt is generated and the power consumption goes up.

3. The WUR receives the signal, decodes the address and performs the address matching. It then generates an interrupt at $t = 6.5\,\text{ms}$ (labeled 'Interrupt in V'). The ramp afterwards is caused by an RC-Timer that resets the WUR circuit.

4. The interrupt is handled by the driver which then enables the main radio. This causes an increase in the power consumption (labeled 'Current in mA').

We used these measurements to get the basic parameters of the system required to properly configure the simulation and the analytical model. The parameters like power consumption and timing information can be found in Table 4.1. The USB WLAN dongle needs more power than an internal WLAN chip because it also includes a USB-PCI bridge that is turned on all the time. Since this additional energy is specific to the type of hardware it is not of concern here. In order to get the power consumption of the WLAN chip only we first measured the base consumption when the dongle is connected to a PC but without enabling the driver. We then subtracted this base power consumption of 766.2 mW from the actual measurements to get the values from Table 4.1.

4.3 Evaluation

Our evaluation is focused on the key metrics energy consumption, delay, and packet delivery ratio. We compared three WLAN power saving modes:

- No power save (always on),
- Native power save (duty-cycling), and
- WUR power save (wake-up receiver).

Table 4.1 – Parameters taken from the hardware prototype

Description	Parameter	Value	Unit
Idle power	P_{idle}	28.55	mW
Active power	P_{active}	593.1	mW
WUR power	P_{wurx}	7.59	µW
Wake-up delay	t_{wu}	15	ms
Active time per beacon	t_b	10	ms

4.3.1 Analytical Model

To model the energy consumption of a real world system one have to take many factors into account that can have an influence. Such factors include the specific hardware used, the selected Modulation And Coding Scheme (MCS), the application and other STA in the same network. A detailed energy model based on empirical results that also takes the processing costs throughout the network stack into account can be found in [73]. In the following evaluation all of these factors are equal for all Power Save Modes which is why we can use a much simpler model to analyze the influence of the PSM.

We designed an analytical model that can predict the power consumption and delay of the three PSMs under consideration. Without any power saving the main radio is always powered on and has a constant power consumption of 593.1 mW in idle mode. If we use a wake-up receiver the STA does not have to wake up periodically to receive the beacons. It therefore also just draws a constant power of 28.55 mW + 7.59 µW. The native power save mode is based on duty cycling, its power consumption therefore depends on how often the STA wakes up which in turn depends on the beacon interval t_i. We measured how long the STA is active to receive the beacon which is $t_b = 10\,\text{ms}$. The actual beacon takes only 1.6 ms but due to clock drift, switching times and possible delay of the beacons due to ongoing traffic, a STA has to stay up longer for each beacon. The power consumption in native PSM for a given beacon interval t_i can then be calculated using Equation (4.1):

$$P_{native} = P_{active} \cdot \frac{t_b}{t_i} + P_{idle} \cdot (1 - \frac{t_b}{t_i}) \tag{4.1}$$

We also investigate the delay between the time when a packet addressed to the sleeping STA arrives at the AP and the time it is received by the STA. We assume that a new packet will arrive at the AP at a random point in time. The delay depends on the time between arrival of the packet and the time that the next beacon is sent out. Additional time that is required for medium access and internal processing is ignored in the following analysis as they are assumed to be equal for all PSM in a low load scenario. If the STA is always active the packet can be delivered immediately, hence the delay is $d_{none} = 0\,\text{ms}$. When using our wake-up based prototype it takes $d_{wur} = 15\,\text{ms}$ to send out the wake-up signal and for switching to active mode. This delay would be shorter if we used a WUR that supports a higher data rate as proposed in the IEEE 802.11ba standard. In the native PSM the delay is determined by the beacon interval. With a high beacon interval the delay will be high because it takes a longer time until the STA is informed about the pending packet.

We assume that the time of arrival of new data is independent of the time that the next beacon is sent out and that it is uniformly distributed between time $t = 0$ and the next beacon at time t_i. Hence the density function in the interval $[0, t_i]$ is $f(t) = 1/t_i$.
The average delay in native power-save mode therefore is:

$$\overline{d}_{native} = \int_{-\infty}^{+\infty} t \cdot f(t) dt = \frac{t_i}{2} \tag{4.2}$$

with a variance of:

$$\begin{aligned} \sigma^2_{native} &= Var(X) = E(X^2) - (E(X))^2 \\ &= \frac{1}{t_i} \int_{t=0}^{t_i} t^2 \cdot 1 dt - \left(\frac{t_i}{2}\right)^2 = \frac{{t_i}^2}{12} \end{aligned} \tag{4.3}$$

4.3.2 Simulation Model

In order to validate our results and to perform more sophisticated experiments with our new protocol we created a simulation model of our system. The model is based on the OMNeT++ network simulator [32] and the INET framework in version 4.2 which includes a detailed model of the IEEE 802.11 stack. The provided model did not yet include the power saving functionality as described in Section 4.1.1 which we first had to implement. This includes the buffering

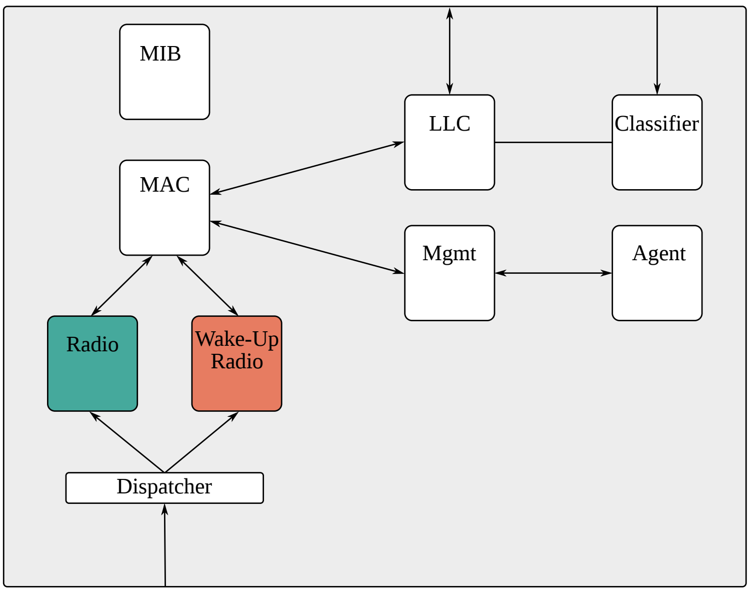


FIGURE 4.6 – Structure of the OMNeT++ simulation model of the Network Interface Card (NIC) with two radios.

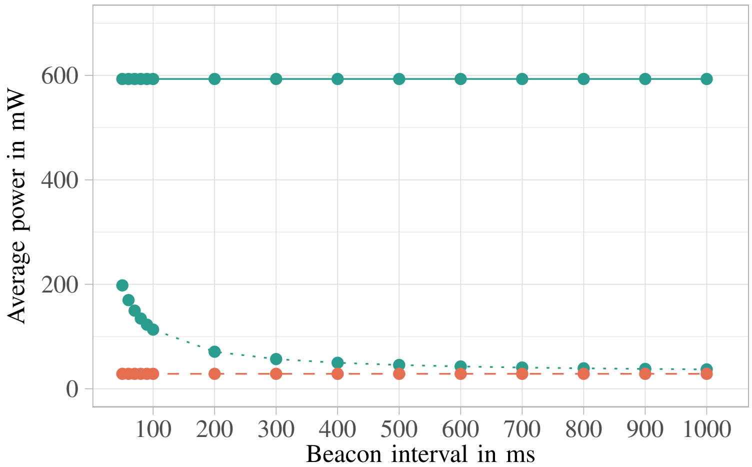
of data at the AP if the destination STA is currently in sleep mode, the signaling of new data via the beacons, as well as the power-mode flag in the IEEE 802.11 MAC header for signaling the current sleep mode to the AP. In addition we extended the model of the Network Interface Card (NIC) to include a second, low power wake-up radio (see Figure 4.6). We parametrized the simulation models with values for energy consumption in different modes and timings taken from measurements of the prototype (see Table 4.1).

4.3.3 Energy Consumption

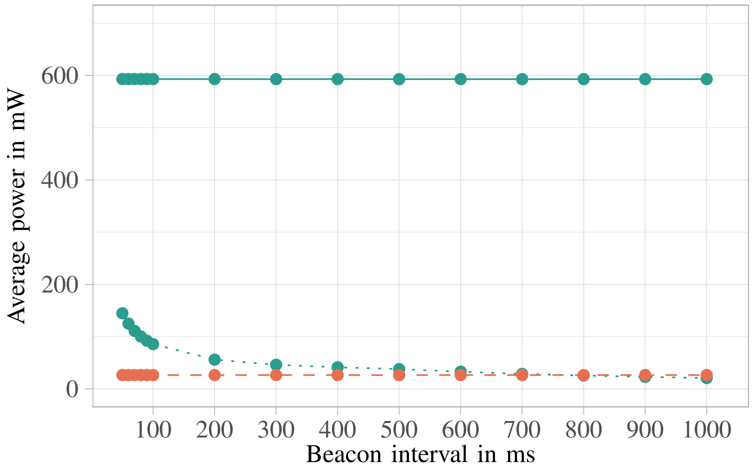
The main incentive for using Wake-Up Receivers is the potential in saving energy as shown in Section 3.2. We conducted extensive experiments with our prototype and our simulation model to evaluate the energy consumption of the different PSMs. The most important parameter in this set of experiments is the beacon interval because it has a significant impact as discussed before in the analytical model (see Section 4.3.1). We evaluated the three PSMs under different beacon intervals ranging from 50–1000 ms. The experiment setup consisted of one AP and one STA, each experiment lasted one minute. Before each of the experiments we configured the AP and STA to use a specific beacon interval and PSM and then measured the energy drawn by the STA when the network was idle. The energy consumption during idle times is especially important for the lifetime of IoT devices because this makes up most of the time in low-traffic scenarios.

The results of these experiments are shown in Figure 4.7. The plots show the mean power drawn by the STA for ten repetitions and under varying beacon intervals. As expected, without using a PSM (solid line) the energy consumption is very high and is not affected by the beacon interval since the WLAN radio is powered on all the time. The native PSM (dotted line), however, draws significantly more power if the beacon interval is decreased. This is caused by the more frequent wake ups from the STA to receive the beacons. Our new wake-up based system (dashed line) shows the lowest energy consumption of all modes because it does not depend on the reception of the beacons which saves a lot of energy. For high beacon intervals the native and WUR power save mode consume nearly the same amount of energy during the experiments.

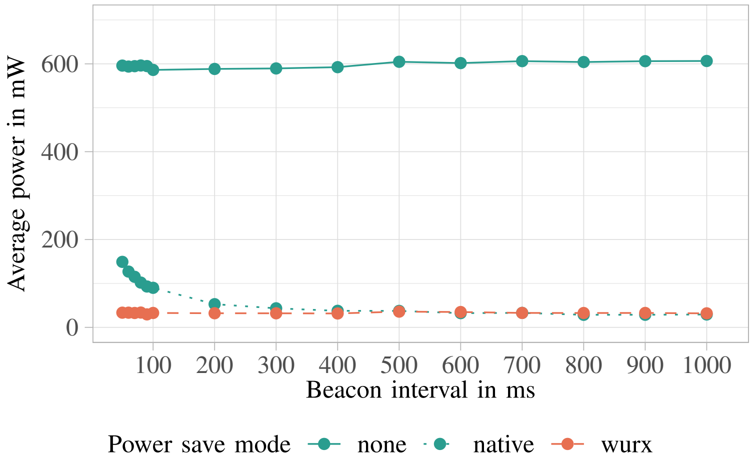
With a beacon interval of 100 ms (which is the default value in most WLAN systems), the average power drawn in normal power save mode is 85.5 mW. This is a reduction by 85 % compared to the always on mode (593 mW). The WUR mode only needed 26.3 mW, which is only 30 % of the energy required



(a) Analytical results



(b) Simulation results



(c) Prototype measurements

FIGURE 4.7 – Energy performance results: In native power save, the energy consumption depends on the beacon rate; the WUR operation has a low energy consumption regardless of the beacon interval.

in normal power save mode and 4 % of the energy using no power save mode, respectively.

4.3.4 Delay

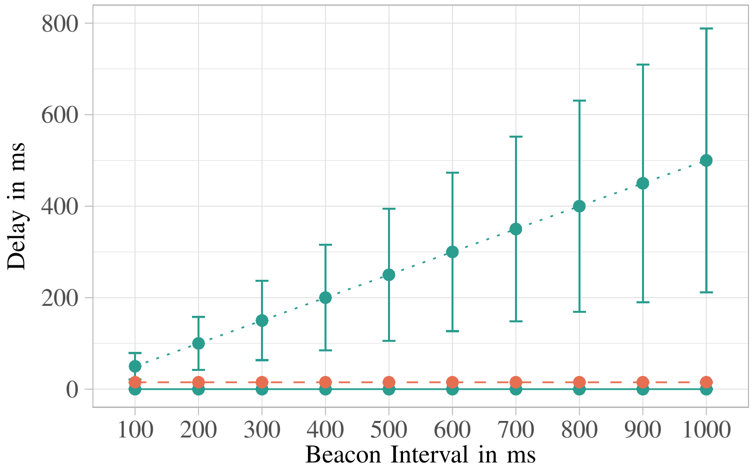
The next important metric that we wanted to evaluate is the delay for delivering an incoming packet to the STA. To measure the delay we sent UDP packets with a timestamp from the AP to the STA. In our experiment setup both entities where running on the same computer which made a separate time synchronization of them unnecessary and allowed us to directly measure the one-way delay. We ensured that all traffic was actually sent over the WLAN dongles by using the Linux network namespacing functionality.

From Equations (4.2) and (4.3) of the analytical model in Section 4.3.1 we get the expected delays shown in Figure 4.8a. With a high beacon interval it takes much longer to deliver incoming packets and also the standard deviation of the delay increases. These observations agree with the results from [74], where a more sophisticated queuing model was used. It shows the same interdependence between the beacon interval and delay/energy consumption.

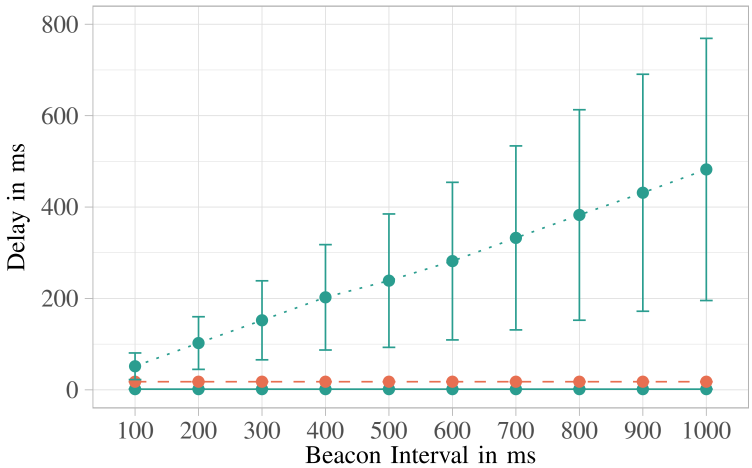
We can see the same behavior in our simulation model and in the hardware experiments. The results from theses experiments are shown in Figures 4.8b and 4.8c. Without using a PSM we measured the smallest delay because there is no need to send a beacon or wake-up signal. The only delay introduced is caused by internal processing time and the time to gain access to the channel. The delay of our new WUR approach is also very low but it includes an additional 15 ms required for sending the wake-up signal. As expected from the analytical model, the native PSM significantly increases the delay because the AP first has to wait until the next beacon time to inform the sleeping STA about the pending packet. The results clearly show the inherent tradeoff between energy consumption and delay that is characteristic for duty cycling based protocols. A higher beacon interval reduces the energy consumption (cf. Figure 4.7b) but increases the delay. With the native PSM we observed not only a higher average delay but also a higher variability (standard deviation). If a packet arrives just after a beacon has been sent it has to be buffered much longer than a packet that arrives just before the next beacon is send. This explains the high standard deviation seen in Figure 4.8.

The hardware prototype has similar results that show the same trend but lower absolute numbers. Especially for a higher beacon interval we can see that the average delay of our prototype is significantly lower than in the simulation. We recorded a detailed trace of the exchanged packets using tshark[23] running

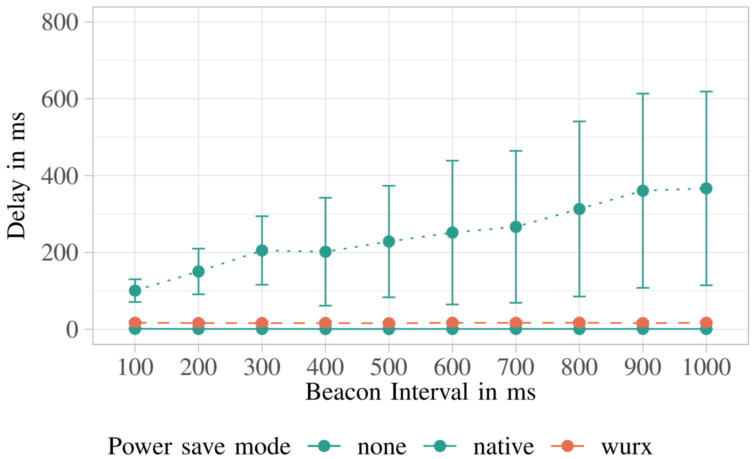
[23] `https://www.wireshark.org`



(a) Analytical results



(b) Simulation results



(c) Prototype measurements

FIGURE 4.8 – Delay performance results: A higher beacon interval increases the delay in native power save mode; without power save or in WUR mode, the delay is considerably lower. Error bars show the standard deviation.

on a separate machine. From this trace we could see that the ath9k based WLAN dongle does not always follow the pattern described in Section 4.2.1. The expected pattern that we observed most of the time (Beacon, Null frame, UDP packet(s), Null frame) was sometimes interrupted when the STA did not go to sleep mode or left sleep mode by itself. This is probably caused by some system processes or internal optimization approaches and is the reason for the lower average delay of the prototype. Such non-standard behavior of commodity hardware is not uncommon as shown in [75].

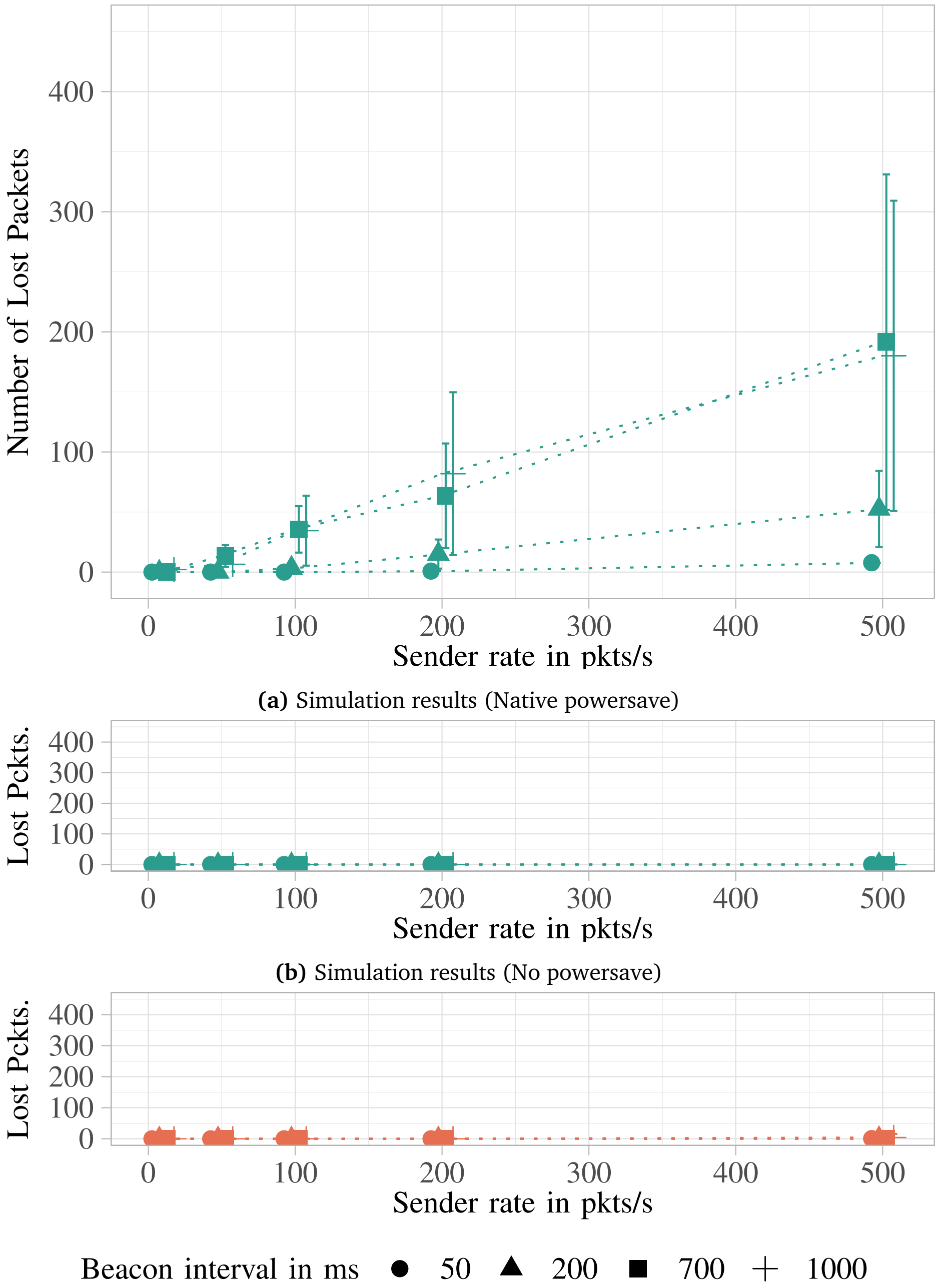
4.3.5 Throughput and Bursty Traffic

The third metric to evaluate the performance of the Power Save Modes is the application layer throughput. For the prototype we used the iperf[24] program to measure the throughput. The results are not shown here because we observed the same throughput for all PSM and beacon intervals. This is caused by the way a STA enters sleep mode: With every data packet a timer is reset that will cause the STA to enter a PSM if it expires (100 ms in our prototype and simulations). If we now run the experiment the steady stream of incoming data will prevent the STAs from entering sleep mode again, which is why the Power Save Mode has no negative impact on the throughput.

The limited buffer size at the AP can, however, lead to a loss of packets if the traffic shows a very bursty pattern. To assess the severity of this problem we conducted the following experiments: Before each experiment run we waited for 10 seconds to make sure the STA has entered sleep mode. We then sent UDP packets for two seconds and with varying packet rates from the AP to the STA. Each packet had a size of 8 kB and was sent with a rate of 10 packet/s – 500 packet/s. At the receiving side we then recorded how many packets where received to see if any packet loss occurred.

Figures 4.9 and 4.10 show the number of lost packets for different sending rates and beacon intervals. If no PSM was used there where nearly no lost packets in all experiments because all incoming packets could immediately be forwarded to the STA (cf. Figures 4.9b and 4.10b). Only in the hardware experiments we recorded some lost packets for high packet rates which could have been caused by interference on the wireless channel or other processes running at the test machines. Also with the new wake-up based system no packets where dropped (cf. Figures 4.9c and 4.10c). The short time from the arrival of a new packet until the Station was woken up of 15 ms was not long enough to fill the buffer of the AP.

[24] `https://iperf.fr`

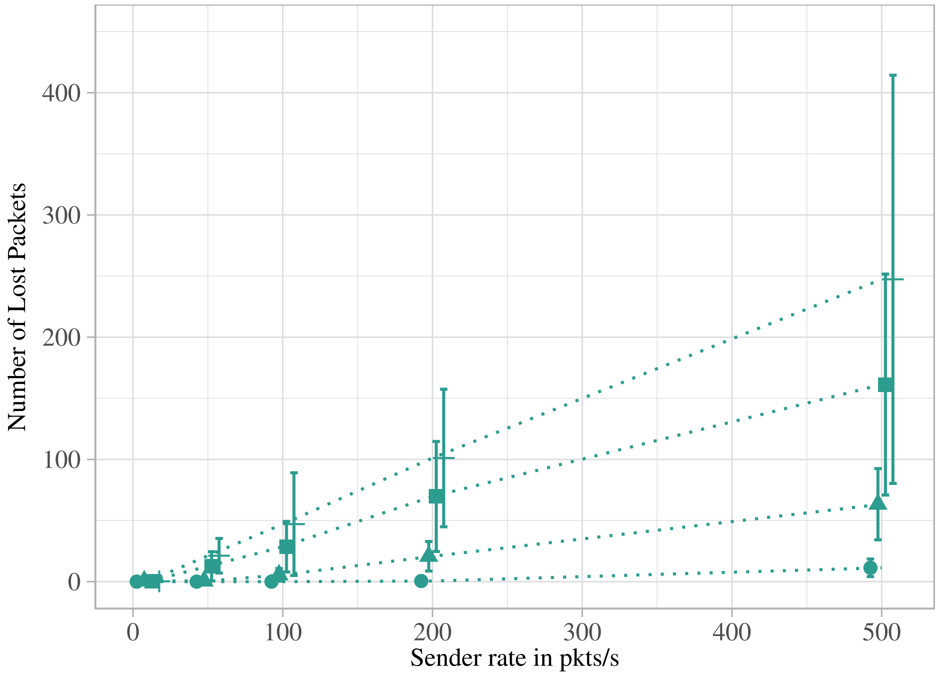


(a) Simulation results (Native powersave)

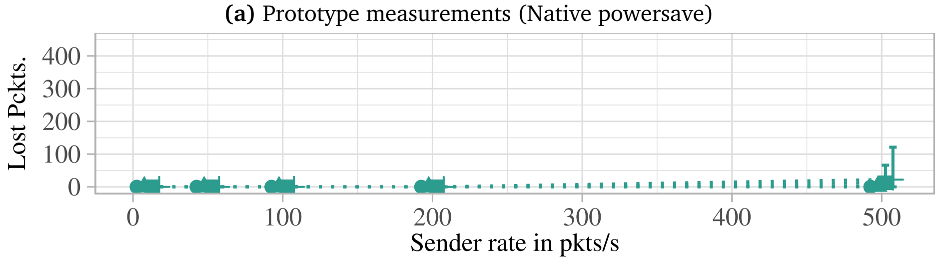
(b) Simulation results (No powersave)

(c) Simulation results (WUR powersave)

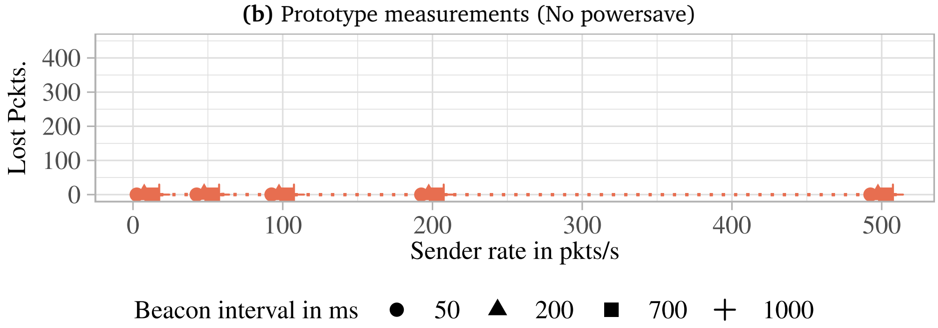
FIGURE **4.9** – Effects of bursty traffic (simulation): In native power mode packets can be lost if they arrive at a high rate and can not all be buffered.



(a) Prototype measurements (Native powersave)



(b) Prototype measurements (No powersave)



(c) Prototype measurements (WUR powersave)

FIGURE 4.10 – Effects of bursty traffic (prototype): In native power mode packets can be lost if they arrive at a high rate and can not all be buffered.

The results for the native PSM, however, show a very different picture (see Figures 4.9a and 4.10a): As soon as we send with a packet rate of 50 packet/s or higher we can see that an increasing amount of packets is lost, especially for higher beacon intervals. Our experiments show this behavior when using the native Power Save Mode and under bursty traffic. It just takes too long until the next beacon is send out (especially with a larger beacon interval) so that the buffer at the AP can fill up and packets have to be dropped. This problem could be mitigated with a larger buffer but the wake-up approach is superior as it does not rely on the choice of yet another parameter.

4.4 Experiments with many stations

The experiments of the previous section showed the basic performance of the different Power Save Modes for a single STA. To see the influence of these modes on a more complex scenario with multiple STAs we extended our simulation model and ran multiple experiments under varying load scenarios. In the extended simulation scenario we placed a single AP in the middle and randomly distributed n STAs around it. We also changed the model of the wake-up transmitter and receiver to use the same frequency band as the WLAN signal and a higher data rate of 250 kbit/s. Before sending the wake-up signal the AP first has to gain access to the channel using the same DCF that is used for the WLAN packets. Each experiment lasted 60 s (simulation time) during which the STAs send UDP packets of 32 B to random destinations.

The simulation parameters that we used are shown in Table 4.1. To ensure that the results are comparable to the previous section we will show here the results based on the simulation runs with a buffer size of 50 packet and a data timeout of 100 ms.

Table 4.2 – Simulation parameters for the multi node scenario.

Description	Values	Unit
Number of hosts	2, 4, 8, 16, 32	nodes
Buffer size	10, 30, 50, 100	UDP packets
Power Save Mode	none, native, wurx	
Beacon Interval	100, 200, 500, 700, 1000	ms
Data Timeout	20, 50, 100	ms
Packet Rate	1, 2, 7, 10	packets per second

4.4.1 Energy Consumption

The results from Section 4.3.3 show that the wake-up based approach requires less energy than the native PSM even for high beacon intervals. This, however, is only the case for scenarios with very little traffic as we would expect in an IoT environment. If a STA receives many packets (higher packet rate) the wake-up based approach consumes more energy than the native one as shown in Figure 4.11. While the buffering of packets until the next beacon is send out in the native PSM increases the delay it also keeps the energy consumption low because a STA can receive a larger amount of packets once it wakes up. The wake-up based STA are woken up for every packet, which keeps the delay low (see Section 4.3.4) but can also increase the energy consumption. This could be prevented if the AP would not wake up a STA immediately but wait a bit to see if there are more packets arriving, which could be implemented in an adaptive way to keep the delays low. The amount of wake-up packets that need to be send should also be kept low for another reason: Since the data rate of the wake-up channel is considerably lower than that of the WLAN channel these signals block the channel for a significant time which has a negative impact on other ongoing transmissions (see Section 5.3.3).

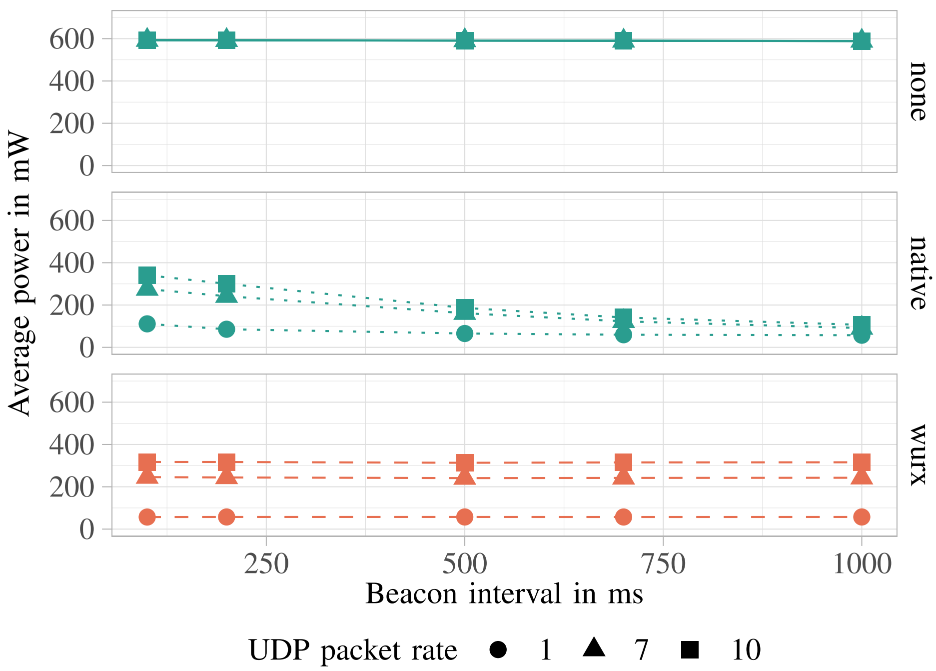


FIGURE 4.11 – Simulation results with 32 nodes. The WUR-approach particularly saves energy for low data rates.

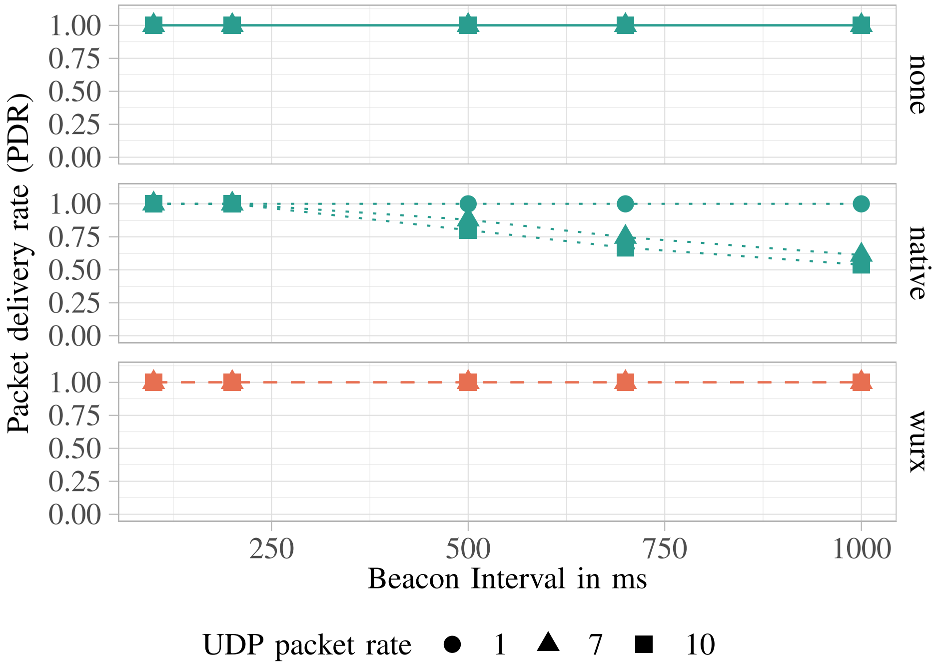


Figure 4.12 – Packet Delivery Ratio (PDR) for scenario with 32 nodes and a buffer size of 50 packets. The AP has to drop packets in native power save mode.

4.4.2 Packet Loss and Buffering Behavior

In Section 4.3.5 we observed that the limited buffer size can cause packet loss for very bursty traffic scenarios. This problem is even more severe in a network with many STAs and with higher traffic. Figure 4.12 shows the Packet Delivery Ratio (PDR) for the multi node scenario with 32 nodes under varying loads (packet rate) and beacon intervals. Without power save mode or with the wake-up approach all packets where transmitted. But the native PSM experiences heavy packet loss if the beacon interval is high. The reason for this can be seen in Figure 4.13. With many stations sending packets, the buffer at the AP quickly fills up for the native PSM, which leads to a significant loss of packets. The WUR approach quickly delivers all incoming packets, the buffer fill level is kept low.

4.5 Conclusion

As we demonstrated in this chapter, it is feasible and beneficial to adapt the wake-up approach also for different use cases and technologies than the ones originating in the field of WSNs. The benefits of low power operation

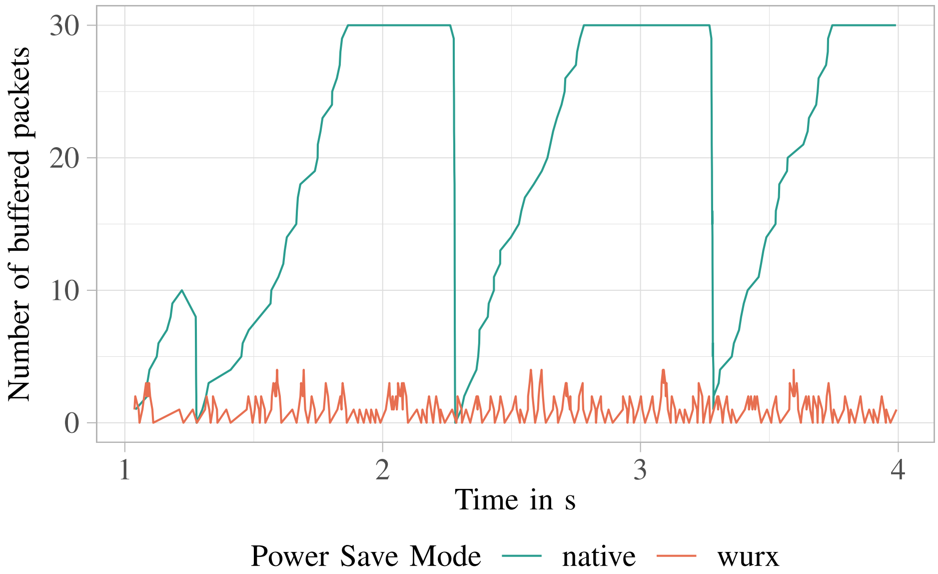


FIGURE 4.13 – Buffer fill level at the AP. In native power save mode packets have to be dropped because the buffer runs full before the next beacon interval. (Simulation run with buffer size 30 packet and 32 STA sending with a rate of 2 packet/s)

without the drawback of high latency as it is common to duty cycling protocols can also be applied to the well known IEEE 802.11 WLAN standard. We demonstrated that a wake-up enabled approach performs significantly better than the native power save mode in terms of energy consumption, delay and stable behavior under varying load conditions. The performance of the protocols was evaluated using an analytical model, a real world prototype and a system-oriented simulation model as well.

Our work showed that WLAN can benefit from wake-up receivers but it also revealed a fundamental limitation of such systems: the low data rate of the OOK wake-up channel blocks the channel for a long time compared to higher modulation schemes which can cause a bottleneck if many STAs have to be woken up. The simple modulation of the wake-up signals also means a poor spectral efficiency, thus using valuable resources from the existing networks. The more dynamic scheduling of the spectrum that is possible with the IEEE 802.11ax WLAN can mitigate this problem. Instead of blocking the complete frequency spectrum of the WLAN channel (20 MHz or more), the AP could simply block the small bandwidth of around 4 MHz required by the wake-up signal for WLAN transmissions, but still use the remaining radio Resource Units (RUs).

Also, too frequent wake ups of a single STA increases the energy consumption and waking up many nodes consecutively increases the delay. Further research has to be done in this field to carefully schedule the wake-up signals and to make use of the dynamic addressing scheme presented in Section 2.2. If many nodes are woken up at the same time and then have to compete for the channel the following resolving of collisions would again use a significant amount of energy. For such a scenario the new 802.11ax WLAN standard can help by coordinating the channel access centrally from the AP as suggested in [76].

The wake-up concept can bridge the gap between high throughput OFDM based radios like WLAN and low power and low data rate protocols for IoT devices. This enables IoT devices with a very tight energy budget to connect to the already existing high performance WLAN networks, making additional hardware bridges and routers unnecessary. While this technology requires additional hardware on the STA side, the transmission of wake-up signals could be achieved with the existing OFDM based radios as suggested in [72]. This upcoming amendment to the WLAN standard (IEEE 802.11ba) defines how the existing hardware can be used to generate an OOK modulated signal, thus enabling the new PSM by simply updating the firmware.

Chapter 5

Low Power Downlink using Wake-Up Receivers

IN the past chapters I demonstrated how the WUR concept can be applied for different applications ranging from WSNs to modern WLAN systems. The main idea behind this concept is having an additional low-power and low-datarate Radio Frequency (RF) receiver that can be used to wake up a node from its sleep state remotely. In this final technical chapter I will explain how this concept can be generalized by using the additional receiver not just for waking up nodes, but for receiving arbitrary control and application data in an energy efficient way.

In [12] we introduce a second radio link between a central WLAN AP and the IoT devices called Low Power Downlink (LPD). While the technical change required for this is rather small, the benefits from this extension are significant and open up a whole new class of protocols. We propose two extensions to the traditional WUR mechanism to realize energy efficient communication:

- *Low-power downlink communication:* We show how we can use the capability of the WUR to receive data not just for addressing information, but for arbitrary application data that can be received by a node without having to enable the main transceiver. As the data rate of the WUR is much lower than that of the main radio we only use this technique for transmissions that require only small amounts of data to be transmitted. This could be some controlling information to change the configuration of the node or an Acknowledgement (ACK) packet to acknowledge the reception of prior transmissions.

- *Efficient uplink scheduling*: If we want to collect data from multiple devices we need to implement a mechanism to access the uplink channel and for contention resolution, which adds additional energy overhead. We show how we can use the WLAN AP to perform the uplink channel access scheduling and keep the channel free for the low power IoT devices with a so called Centrally Controlled Channel Access (CCCA). This mechanism also ensures backward compatibility with legacy WLAN devices. The OOK modulated wake-up/LPD signal can be send by a secondary radio or by using Cross Technology Communication (CTC) techniques [77], [78].

This new functionality can be implemented in an energy efficient way by using the Low-Power Universal Asynchronous Receiver/Transmitter (LPUART) module that is available in modern microcontrollers [79]. This module allows the reception and decoding of the LPD packets without having to wake up the microcontroller from its deep sleep state, thus reducing the energy required for reception. We also introduce a flexible frame format that supports a wide range of possible applications, allows for flexible addressing of nodes using unicast, broadcast or multicast while keeping the length of the signal as small as possible. Such frames can be used to transmit arbitrary application data, controlling information and uplink scheduling information which can be used by the nodes to compute their *contention-free* uplink transmission slot. The well known functionality of waking up nodes with a remote signal is still possible with our extension and can be seen as a special case of a more general communication protocol.

To demonstrate the feasibility of our LPD approach we created a hardware prototype using commodity hardware. In addition we evaluated the performance of our system and the impact on existing WLAN systems using a detailed simulation model. We also provide an application example where we improve the energy harvesting efficiency of a low power, wearable device using our novel LPD protocol.

Our main contributions are:

- *WUR-based Low Power Downlink:* We generalize the wake-up concept to support a more flexible and energy efficient communication with IoT devices while keeping compatibility with existing 802.11 WLAN systems. The flexible LPD frame format (Section 5.1.4) supports different addressing schemes and a wide range of possible applications while minimizing the signal length.

- *Reliable and Efficient Uplink:* The proposed CCCA mechanism can also be used to schedule the uplink data transfer for many concurrent IoT nodes in an energy-efficient way (Section 5.1.1). The reservation of the channel is performed centrally by the AP which prevents other WLAN devices from interfering and thus removes the need to perform a Clear Channel Assessment (CCA) by the IoT nodes before transmission. This in conjunction with a Time Division Multiple Access (TDMA) based multiplexing significantly reduces the energy required for sending data from the IoT devices to the AP.
- *Energy efficient LPD Message Processing*: The increased complexity of the LPD frames in comparison to a simple wake-up message makes it necessary to use a microcontroller for receiving and decoding the message. We demonstrate how this can be done while the microcontroller is in a low power sleep mode by using the LPUART peripheral module that is commonly available. We developed a hardware prototype (Section 5.3.1) that can receive an LPD message with just 22.3 µJ energy overhead, which is 46% less than with a naive microcontroller based solution.
- *Superiority over Competitive Baselines:* We evaluate the performance of the LPD approach using extensive simulation studies and compare it with native WLAN power save and the wake-up based 802.11ba protocol. Our system can provide an $\sim$ 87% – 186% increase in throughput for the uplink channel and a $\sim$ 50% – 98% decrease in energy consumption. We also show in an application example how the LPD approach can be used to optimize the beamforming – and hence the harvested energy – in a WLAN based RF energy harvesting system.

Our generalized communication protocol provides IoT devices with an energy efficient and flexible way to exchange data with central AP while maintaining compatibility with existing WLAN systems.

5.1 System Architecture

In the last chapter we evaluated how Wake-Up Receivers can be used in networks with a star topology and heterogeneous devices.[25] This combination can already provide a system that is very energy conserving and thus can be used for IoT devices and, at the same time, provides high data rates for more powerful multimedia devices using the main radio. From this work we take the concept of having two radios, one powerful and energy hungry main

[25] Centralized AP that has unlimited energy and battery powered mobile devices.

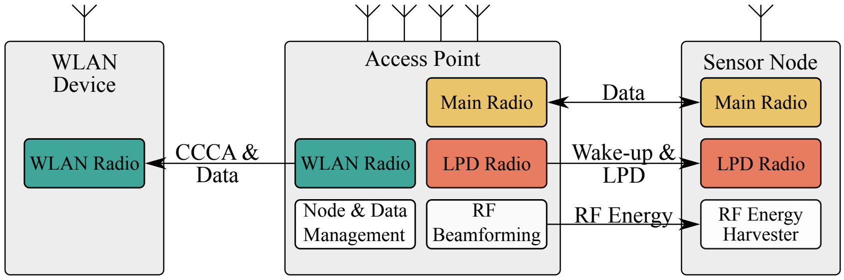


FIGURE 5.1 – System Architecture: A central AP can communicate to both standard IEEE 802.11 and LPD IoT devices using a high-performance and a low-power radio, respectively. The AP can also use RF beamforming to support RF energy harvesting.

radio (IEEE 802.11 WLAN in our case) and one ultra-low power radio with a low data rate. In contrast to the previous chapter, however, we use the secondary radio in a more general way to transmit arbitrary data from the AP to IoT devices, hence the name Low Power Downlink (LPD). In our system the IoT nodes still use a high-performance WLAN radio for data communication, but we reduce the energy consumption by a) using the LPD to send data requests and scheduling information and b) improve the efficiency of the uplink channel with our Centrally Controlled Channel Access (CCCA). The secondary receiver, of course, can still be used in the traditional way to just wake up nodes remotely.

Figure 5.1 illustrates the general architecture of our system: The AP is equipped with at least two radios, a legacy IEEE 802.11 WLAN radio and a simple OOK radio.[26] The main radio of the IoT devices can be based on WLAN, or, as in our prototype, use a different technology. In addition to providing a wireless connection for normal WLAN devices the AP also performs additional tasks like detecting new LPD nodes, collecting data from these nodes and coordinating the channel access in a WLAN compatible way. We assume that the AP is equipped with multiple antennas and supports beamforming that we use twofold: a) To increase the signal strength for the WUR to mitigate the problem of low sensitivity and low communication range, and b) to transfer RF energy to the mobile nodes (see Section 5.3.4). The low-power sensor nodes have two radios: A low-power, low data rate wake-up LPD radio that is monitoring the channel for incoming OOK signals constantly and a high-performance, high data rate radio.

[26] The transmission of OOK modulated signals can also be done with the OFDM based WLAN chip using Cross Technology Communication (CTC) as in [78].

5.1.1 Centrally Controlled Channel Access (CCCA)

The channel access of WLAN is based on the Distributed Coordination Function (DCF) which includes sensing the channel before transmission and possible backoff times. This access scheme can consume a considerable amount of energy as shown in [76]. We therefore propose a channel access similar to the one proposed in the 802.11ba standard: Before an LPD packet is send the AP first gains access to the channel using the normal 802.11 DCF. It then transmit a normal WLAN header to inform legacy STAs about the ongoing transmission and sets the `duration` field of this header to a sufficiently large value. Normal WLAN devices that receive this header will update their Network Allocation Vector (NAV) and refrain from sending during this time. In contrast to the 802.11ba operation we do not only use this technique to guard the OOK signal from interference but also the following uplink phase, where the low power LPD nodes transmit data back to the AP. This way we can ensure compatibility with legacy WLAN devices and simplify the uplink channel access for IoT devices, thus reducing the energy consumption in comparison to a simple wake-up operation.

5.1.2 Wake-up and Data transmission

Based on the CCCA we can now define our protocol for low-power data transmission. We use the ability of a wake-up receiver to decode OOK modulated data as a general purpose Low Power Downlink (LPD) from the AP to the nodes. Based on the results from Section 2.2 we support different addressing schemes and different operation modes (wake-up only, data transfer or both) that are described in more detail in Section 5.1.4. We can receive these LPD frames in an energy efficient way without waking up the node's main radio or the microcontroller (see Section 5.3.1).

As explained before, the nodes do not have to perform channel sensing before uplink transmission because the CCCA scheme ensures that no other WLAN devices transmit during the uplink phase. However, as we also support broadcast LPD packets, we do have to consider concurrent uplink transmissions between multiple LPD nodes. We propose to use a simplified channel access that can also be controlled by the AP and additional scheduling data that is encoded in an LPD frame. As an example we used a slotted TDMA scheme where the number of slots is selected by the AP. We assume that each node has a unique ID[27] which determines the uplink transmission slot.

[27] The AP could assign a unique ID to the nodes during the discovery phase.

5.1.3 RF Power Delivery

In addition to the communication capabilities of our system we also show how we can combine low power communication with RF energy harvesting. While these two aspects can be studied independently they complement each other very well as it shows how the LPD link can be used in a very challenging IoT scenario. Also we will demonstrate how the energy transmission can be optimized with the LPD technology. The AP uses its antenna array to first derive the angular position of the nodes from the incoming signals. It then steers a beam of RF energy towards the nodes, which can harvest this energy to run without a battery. We use our low power communication scheme to periodically update the node's position and to redirect the RF beam if nodes are mobile.

5.1.4 Frame Format

The low power consumption of wake-up receivers is achieved by using a very simple modulation, such as On-Off Keying (OOK), which in turn implies very low data rates. For example, the chip we used for our studies (AS3933) supports data rates up to 4 kbit/s, and the IEEE 802.11ba standard draft supports data rates up to 250 kbit/s for WLAN wake-up. Accordingly, the data length of a wake-up frame should be fairly small, to both reduce the wireless receiver energy consumption and to minimize the channel *holding time* (which would adversely affect overall WLAN throughput as we will show in Section 5.3.3).

The structure of an LPD frame is shown in Figure 5.2 and Table 5.1. It is designed to be small and at the same time flexible enough to support a variety of use cases. Before the transmission of an LPD frame the AP gains access to the channel and then sends a WLAN header as explained in Section 5.1.1. The beginning of a frame is determined with a fixed preamble that is prepended

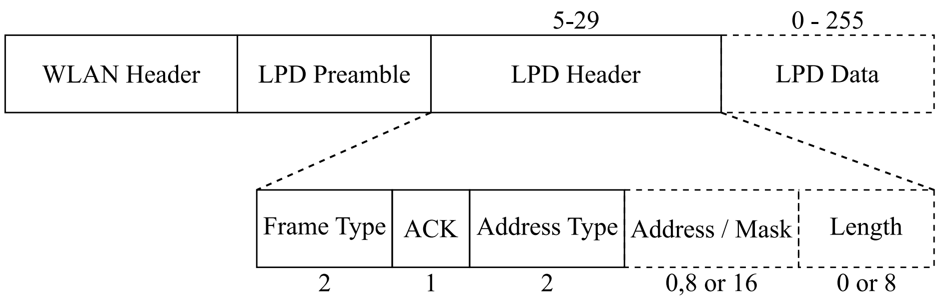


FIGURE 5.2 – Frame format for low-power downlink. The numbers denote the length of the fields in bits. Dashed fields are optional

to each frame. In our system we follow the format from the AS3933 chip as depicted in Figure 2.7, with other hardware this might be different. The first data field of the LPD frame is the *frame type* which denotes whether it is a wake-up frame, a wake-up frame with additional data or a data frame. If a node receives a wake-up frame (with or without data) it will enable the main radio to exchange data with the AP. The second field is the *ACK* field and can be used by the AP to acknowledge the reception of a previously transmitted uplink packet. This way the main radio only has to be enabled to send an uplink packet without waiting for an acknowledgment. The third field denotes the *address type* which enables our system to use a flexible and variable-length addressing to communicate with either a single or with many nodes. The address type denotes whether the LPD frame is addressed to all nodes (broadcast), a single node (unicast) or to a subset of nodes (multicast). Depending on the address type, the frame then contains 0 to 16 bits of addressing information[28]: When using a broadcast we do not have to send any additional information. For a unicast frame we have to specify the 8 bit *address* field. If the address type denotes a multicast frame we have to send an *address* and in addition an 8 bit mask. The shortest possible LPD frame therefore is a broadcast wake-up frame which only contains 5 bit of data plus the preamble. If the frame carries additional data the last fields of the LPD header contains an 8 bit *length field* and finally the application *data field* (0–255 bit).

5.1.5 Use of the LPUART Module

A fundamental requirement of the LPD protocol is the reception of an OOK modulated signal in an energy efficient way. While we have shown in Chapter 2 how the analog frontend can be built with nearly no active components and the chip we used in our experiments so far (AS3933) has proven to be very energy efficient, we still have to consider the costs to decode an incoming packet. If we would simply enable the microcontroller to read and process the incoming bits we would use a significant amount of energy, because the data rate of the LPD channel is low. In our prototype, for example, where we send an LPD frame with up to 64 bit at a rate of 4 kbit/s, the microcontroller would have to stay active for 16 ms to receive the complete frame. Using this naive approach would consume too much power, rendering the LPD approach useless. We therefore propose to make use of the Inter-Processor Communication (IPC)

[28] With an address/mask length of 8 bit we can address 256 different nodes which should be sufficient for most application scenarios as the addresses only have to be unique for all nodes in communication range of an AP. Of course this can be adapted if necessary. For example in a scenario with WLAN as main radio one could use the 16 bit Association ID (AID).

functionality included in modern ARM-based microcontrollers that enables us to receive the incoming LPD frames while keeping the microcontroller in its low power *sleep* mode. The STM32L053 microcontroller that we used in our prototype has a so called LPUART module that runs independently from the main processor and can receive serial data, while the rest of the microcontroller is in sleep mode. We combine this module with the functionality of the AS3933 wake-up receiver to output any additional data it receives on one of its pins.[29] The signal transmitted by the AP was then chosen in a way that it follows the format that is expected by the LPUART module (including start and stop bit). In Section 5.3.1 we will show how this approach reduces the energy required for receiving LPD frames significantly.

5.2 Application Example

In this chapter we will now illustrate how the LPD concept can be applied in an IoT scenario with mobile, wearable devices. We will describe the functionality of our application and how it can be achieved using the protocol described in the previous chapter. Our application scenario contains mobile, wearable devices that measure acceleration data and send the collected data to an AP. The nodes do not have a battery and receive their energy solely by harvesting RF energy that is radiated by the AP. As main radio we use a 2.4 GHz based low power radio (nRF24L01). Even though this radio consumes less power

[29] We already used this to build the Selective Wake-Up Receiver from Section 2.2.

Field Name	Length	Value	Description
Frame Type	2 bit	00	Wake-up only
		01	Wake-up and Data
		10	Data only
ACK	1 bit	0	No ACK
		1	Acknowledgment
Address Type	2 bit	00	Broadcast
		01	Unicast
		10	Multicast
Address	8 bit	0 – 255	Address of receiver
Mask	8 bit	0 – 255	Multicast mask
Length	8 bit	0 – 255	Length of Data field
Data	0–255 bit	*	Arbitrary application data

Table 5.1 – Fields of an LPD frame and its values

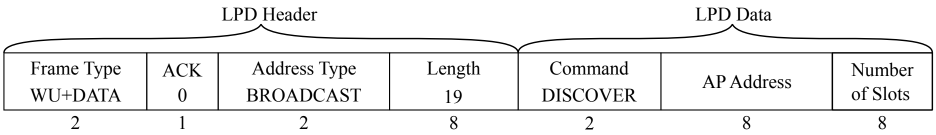


FIGURE 5.3 – Frame format for the discovery of new IoT nodes by the AP.

(57 mW RX, 39 mW TX) than a typical WLAN chip, we cannot us duty cycling or Carrier Sense Multiple Access (CSMA) because the RF harvesting does not provide enough energy. Instead, we equipped the nodes with the AS3933 wake-up receiver, that we use as a general ultra-low power OOK receiver as described before. The system supports the following functionality:

1. Initial discovery of new nodes,
2. Requesting sensor data from the nodes,
3. Adaptive pinging to optimize the beamforming and hence the harvested RF energy at the nodes.

Based on the frame format from Section 5.1.4 we created an application specific communication protocol that is visualized in Figures 5.4 and 5.5. We use the data field of the LPD frames to define different types of packets like *DISCOVER*, *REQUEST*, or *PING*.

Initial Node Discovery

The first step in our application example is the discovery of new nodes. As we cannot know the node's positions in advance, the AP will first make sure that the devices can harvest a minimal amount of energy by sweeping the area before (changing the angle of the RF beam from 0–360°) for a certain period. We also do not know how many nodes are in communication range and can therefore not schedule the uplink communication using a TDMA approach in this step. The AP, however, can use the data field to send scheduling data like the number of available slots (see Figure 5.3) or a transmission probability per slot to the nodes.[30] We can also leverage the multicast addressing of the frame format to split the new devices into smaller groups. Such algorithms are known in the realm of Radio Frequency Identification (RFID) where an unknown set of tags needs to be identified [81], [82]. For simplicity, in our application we assume that each node has a unique ID which allows us to use a slotted TDMA based uplink.

[30] The selection of these values will depend on the network density, similar to the ALOHA based protocol used in Long Range Wide-Area Network (LoRaWAN) [80].

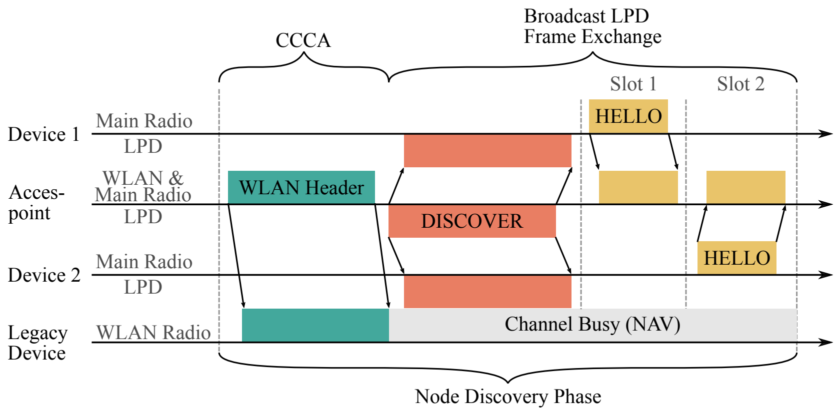


FIGURE 5.4 – Overview of the communication scheme. The AP first performs a CCCA and then sends a broadcast discovery packet via the LPD link to the nodes. The nodes answer with a HELLO packet using a slotted TDMA protocol.

The discovery phase is initiated by the AP that sends a broadcast LPD packet with a structure as shown in Figure 5.3. The frame type is set to 01 (wake-up and data) and contains a command (*DISCOVER*), the address of the AP and the number of slots used for the following uplink phase. All nodes that receive this packet will wake up, enable their main radio and answer with a *HELLO* packet in the slot determined by their own unique ID (see Figure 5.4). The AP receives the *HELLO* packets and thus learns the IDs of the mobile nodes. In addition, it can determine the nodes' position by calculating the Angle Of Arrival (AoA) from the Channel State Information (CSI). Later on, this information can be used to control the direction of the RF beam towards the nodes. The *HELLO* packet can also include additional information such as supported sensors. As we assume the nodes to be mobile, they can enter and leave the communication area. Therefore the AP will perform the discovery phase repeatedly.

Requesting Sensor Data

After the AP has discovered all nodes, it can request the collected data from the nodes as illustrated in Figure 5.5. To do so it sends an LPD packet either to a single node (address type: 01) or to many nodes (address type: 00 or 10) with the command field set to *REQUEST*. The mobile nodes then answer with a packet containing the sensor data over the main radio. Channel sensing before transmission is not necessary as the CCCA mechanism ensures that no other WLAN device transmits during the uplink phase. Upon the reception of the data packets, the AP can also update the positions of the nodes.

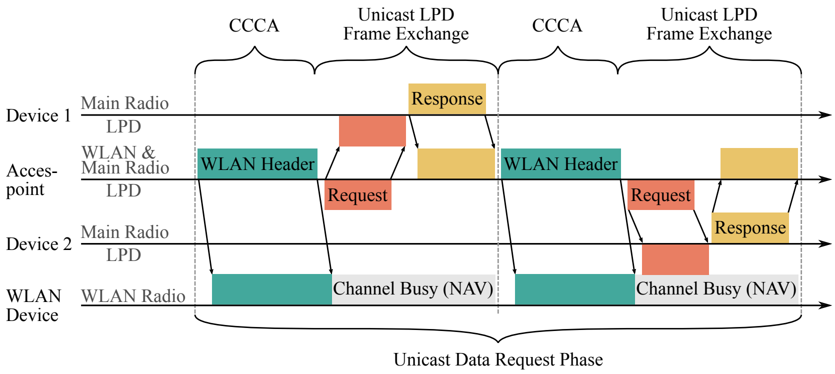


FIGURE 5.5 – Overview of the communication scheme. The AP first performs a CCCA and then sends unicast data request packets via the LPD link to the nodes. The receiving node then answers with a data response packet.

Adaptive Pinging

The wearable devices we consider are not equipped with a battery, but harvest their energy from the electromagnetic waves emitted by the AP. In order to harvest enough energy this way, the AP has to send the energy in a narrow beam towards the devices using beamforming. As described before, this is possible because the AP can estimate the AoA from the signals of the mobile nodes. If a node changes its position the AP has to know about the new position, so that it can update the direction of the beam. The rate at which data packets from the nodes are received might not be high enough (depending on the mobility of the nodes) to follow the nodes position accurately. Therefore the AP can send a *PING* request packet to individual nodes over the LPD link. The receiving node then answers by sending a short response packet over the main radio that can be used to determine the new position of the node. The AP can adapt the rate of such packets for each node individually based on its mobility. In addition, the nodes can transmit information about the charging state in the response packet that can be used to further schedule the energy transmission from the AP to the nodes.

5.3 Evaluation

To show how the LPD approach performs in terms of energy consumption and latency we conducted experiments using a hardware prototype and extensive simulations. The experiments are based on the application example from the previous chapter with battery-less wearable devices, that harvest RF energy and send collected sensor data to a central Access Point. We also investigated the influence of the LPD on legacy WLAN networks.

5.3.1 Hardware Measurements

The hardware prototype that is shown in Figure 5.6 consists of the following four main components:

1. An ARM-based microcontroller (STM32L053), that has a very low power consumption (sub-µW) in the deep-sleep mode,
2. An ultra-low power 3-axis accelerometer (LIS3DH),
3. A main radio (nRF24L01+), that uses a proprietary protocol on the 2.4 GHz band with up to 2 Mbit/s, and
4. A wake-up receiver board based on the AS3933 with an envelope detector tuned to the same band as the main radio.

The AP is based on the Wireless Open-Access Research Platform (WARP) version 3 [83] in conjunction with the open-source WARPLab experiment platform. This setup was extended by a Field-Programmable Gate Array (FPGA)-based packet detector to receive the nRF24L01+ packets. To measure the power consumption of the hardware prototype we used a high precision 10 Ω shunt resistor in series with the power pin of the device and measured the resulting voltage drop using a NI DAQ 6003 data acquisition device at a rate of 100 kS/s.

The first experiments show how we can receive LPD packets in an energy efficient way and without waking up the microcontroller as described in Section 5.1.5. To do so, we implemented the reception in two different ways:

Microcontroller: The microcontroller is woken up by the WUR and actively reads in the voltage level of the GPIO connected to the DAT pin of the AS3933. This takes 6 ms because the data rate of the LPD signal is very low (at most 8 kbit/s without Manchester coding).

LPUART: The microcontroller ARM-core is kept in a deep-sleep state. The reception of the data is done by the LPUART module that can be active independently of the rest of the chip. While this was actually introduced

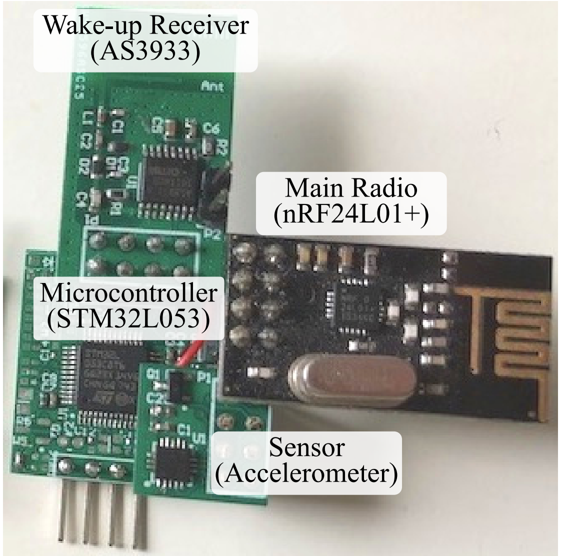


FIGURE 5.6 – Picture of the hardware prototype with microcontroller board, sensor board, main radio and Wake-Up Receiver.

to support multiprocessor communication, we repurpose the module to receive the data coming from the AS3933 in an energy efficient way.

We then measured the current consumption of the device while receiving an LPD packet sent from the WARP board. The packet includes the preamble required by the AS3933 chip and two bytes of data. The LPUART module expects the data in a serial format where each byte is wrapped by a start bit and three stop bits. Additionally we had to prepend 10 consecutive 1 bits in front of the data to make sure that the LPUART is in IDLE state. After the reception of the frame the device wakes up and sends back a packet using the main radio.

The results of the hardware experiments are shown in Figure 5.7. The reception of the LPD signal starts at 4 ms with the preamble for the AS3933 chip (see Figure 2.7). At 12 ms the actual data reception starts, which causes the power consumption to rise to 3 mW when using the naive microcontroller based approach (blue line). When using the LPUART to receive the data (orange line) the power consumption is significantly smaller because the microcontroller can stay in a sleep state. Only at the begin of the data reception (12 ms) and when one byte is received (16 ms), it has to wake up briefly to configure the LPUART and to save the content of the receive buffer. In total, the reception of the LPD packet used 27.7 µJ of energy when using the naive approach, and only 8.6 µJ for the LPUART approach which is a reduction of 69 %. At 19 ms

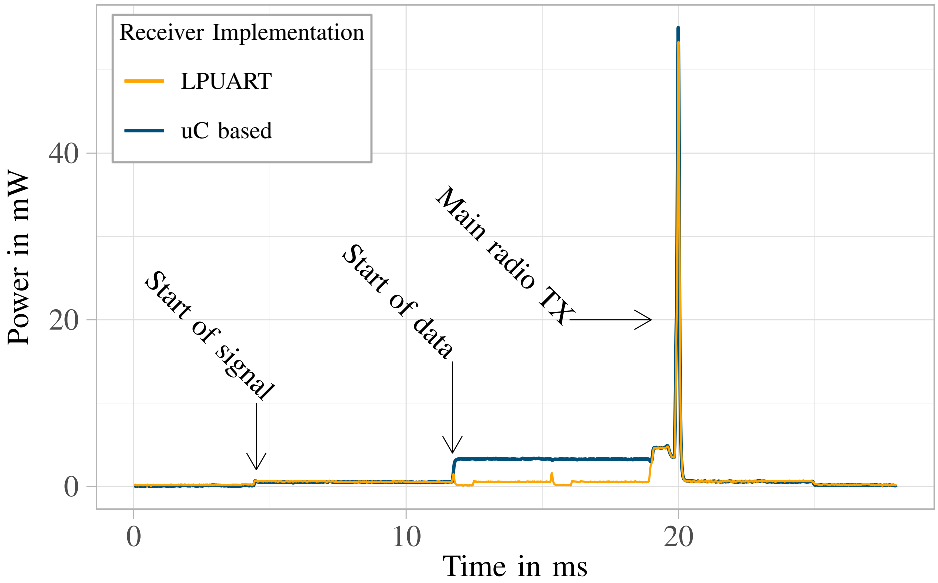


FIGURE 5.7 – Power consumption of prototype (with and without the LPUART) during reception of LPD frame and subsequent uplink transmission of a packet over the main radio.

the microcontroller enables the main radio and sends a packet back to the AP which causes the spike in energy consumption at 20 ms.

5.3.2 Comparison with 802.11 Energy Saving and 802.11ba

We compared our LPD approach with the normal duty-cycling based WLAN Power Save Mode (PSM) and the wake-up based IEEE 802.11ba protocol as described in Section 4.1 ff. The evaluation scenario for the simulation is based on the application example from Section 5.2, where a central AP discovers new nodes and then queries sensor data from the nodes. The simulation model was built using the OMNeT++ simulation environment and the INET framework to model the wireless transmissions. It includes an antenna model that supports beamforming, an energy harvesting model that can receive RF energy and charge a capacitor, and a comprehensive model of the three communication protocols using multiple radios. In this chapter we focus on the communication protocols.

The WLAN power save protocol is based on the duty cycling approach, explained in Section 4.1, where a Station first notifies the AP about going to sleep mode and then periodically wakes up to receive the beacons send by the AP. If there is new data available for the STA (as indicated by the Traffic Indication Map (TIM) in the beacons), it will notify the AP about leaving sleep mode and receive the data afterwards. With the wake-up based approach from the 802.11ba standard [72], a STA does not have to receive the beacons as the AP can directly notify it if new data arrives using a wake-up signal. The transmission of the wake-up signal is guarded against interfering WLAN transmissions by prepending it with a WLAN header that indicates the duration of the transmission.

The simulations use the three protocols to implement the following experiments:

1. In the initial phase the central AP discovers new nodes in its vicinity as explained in Section 5.2. The WLAN based protocols first have to associate with the network.
2. The AP then initiates the data collection (request phase) by sending a *Request* packet to either
   (a) each node consecutively using unicast packets, or
   (b) to all nodes simultaneously using a broadcast packet.[31]

The relevant simulation parameters are shown in Table 5.2. We evaluate the performance with varying request intervals, number of nodes in the network and with unicast and broadcast requests. In these scenarios, there are no other WLAN Stations that would interfere with the communication. Each combination of parameters was simulated 10 times with random placements of the nodes around the AP.

The first metric we analyzed was the time it takes for the AP to query the data from all nodes (time of one request phase). Figure 5.8 shows that the

[31] The following uplink transmissions are coordinated using the 802.11 DCF for the WLAN based protocols, and using a simple slotted TDMA based approach when using the LPD protocol.

Table 5.2 – Simulation parameters for the protocol scenario.

Description	Values	Unit
Simulation time	20	s
Number of sensor nodes	4, 8, 16, 32	nodes
Request Mode	Broadcast, Unicast	
Protocol	WLAN, 8021.ba, LPD	
Main radio	nRF24L01+, CC3200	
Request Interval	33, 100, 1000	ms
Slot length	1.2	ms

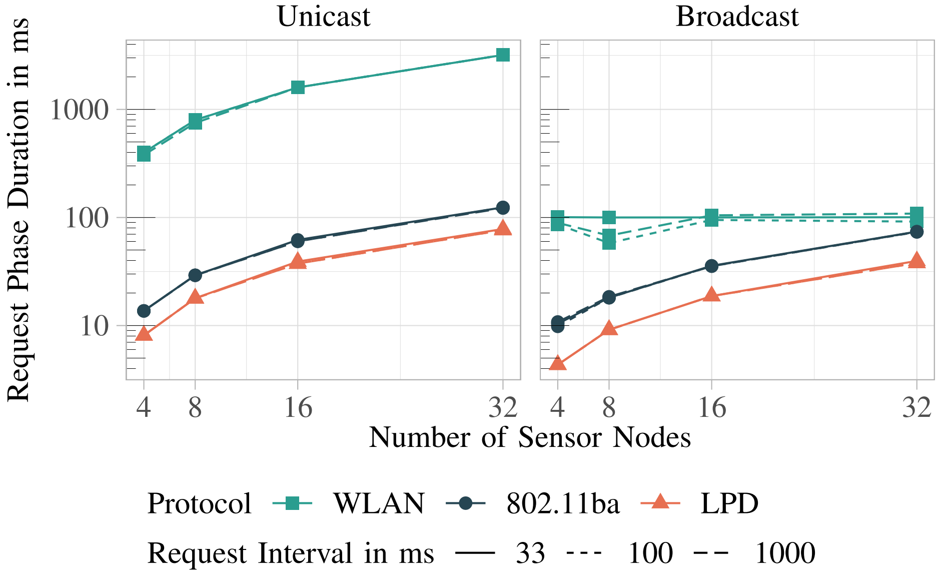


FIGURE 5.8 – Duration of time to get sensor data from all nodes(request phase)

LPD has the lowest mean duration per request phase among all protocols. When using the unicast approach, where the AP polls the data from each node consecutively, it takes 8 ms to poll the data from 4 nodes and 78 ms in the scenario with 32 nodes. Each request takes the same amount of time (no interfering networks in this scenario), hence the duration of the request phase grows linearly with the number of nodes that have to be queried per phase.[32] The 802.11ba based approach takes about 60 % more time (13–123 ms) because each node first has to be woken up for each request, receive the request packet from the AP and then send back the sensor data. Even though the data rate on the physical layer from WLAN is much higher than the one used in the LPD approach, the distributed channel access using the DCF causes a lot of additional waiting time and an overall increased request phase duration. As expected from the results in Section 4.3.4, the duration of the request phase is the highest when using the native Power Save Mode from WLAN. This is caused by the waiting times introduced by duty cycling which requires the AP to wait until the next beacon is send out to inform a node about the pending request packet.

[32] Note the logarithmic scale!

For the unicast approach this results in a request phase duration of 378–3187 ms[33].

The broadcast approach significantly reduces the time it takes to query all nodes. The AP will send the request packet to all nodes at the same time using a broadcast packet. With the LPD approach this packet will be send over the LPD link and the uplink transmissions are coordinated using slotted TDMA with a slot length of 1.2 ms. The two WLAN options send the request packet via the normal WLAN link and uplink channel access is coordinated using the DCF. For the native WLAN PSM the major factor that determines the duration of the request phase is the beacon interval. The duration to transmit the actual data back to the AP is negligible in comparison. Within the 20 s of the simulation we where able to query the data from 32 nodes 435 times using the LPD approach, which is an improvement of 186 % over the native WLAN PSM (152 request phases), and an improvement of 87 % over the 802.11ba based approach with 232 requests.

A fast completion of a request phase ensures that nodes can be queried often. If the duration of one request phase is larger than the request interval, the beginning of the next phase is delayed which means that only a fraction of the planned requests can be fulfilled. This behavior can bee seen in Figure 5.9 where the ratio of successful requests during the simulation time and the number of scheduled requests (simulation time/request interval) is shown. All protocols show a better ratio when using the broadcast approach than with unicast requests, because it can complete a request faster than with the unicast approach.

The LPD approach can fulfill nearly all request when using the broadcast approach. Only for the lowest request interval (33 ms) and highest number of nodes (32 nodes) not all requests can be fulfilled. The 802.11ba protocol has a slightly higher request duration which translates into a reduced number of successful requests, especially if many nodes have to queried. WLAN has the worst performance in this metric because of its high delays and can thus only satisfy a small fraction of all planned requests.

Finally, we evaluated the energy efficiency of the three protocols. We parametrized the simulation models of the wireless interfaces with the values for energy consumption during sleep, idle, transmission and reception taken

[33] Here the average delay is not $t_i/2$ as in Section 4.3.1 (t_i being the beacon interval) because the arrival of new packets and the time when a beacon is send are no longer independent from each other. Instead, right after a beacon has been sent and the request of one node is finished, the AP will try to query the next node but now again has to wait 100 ms until it can send the next beacon. Only the first query will take $t_i/2$ on average

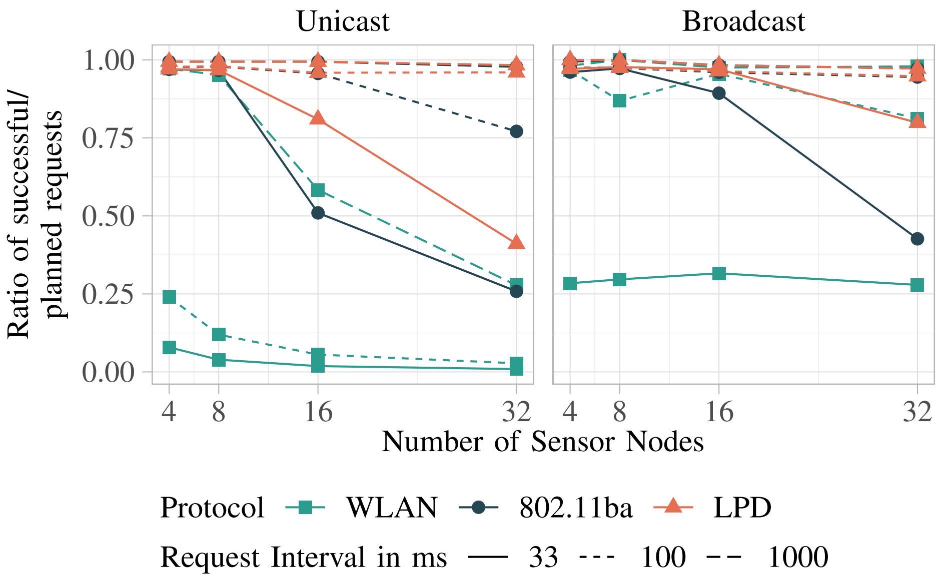


FIGURE 5.9 – Ratio of successful request phases to planned number of requests. If one request phase takes longer than the request interval, not all requests can be fulfilled.

from the data sheets[34] [4]. The mean energy per transferred uplink byte (application layer) is shown in Figure 5.10. The total energy that a node consumes includes the energy to transmit the application data, the communication overhead (including the reception of beacons in the WLAN scenario) and the energy consumed in idle/sleep states (in the wake-up based scenarios the WUR is always on). Because of that the energy per byte (*relative*) is higher for scenarios with a low request rate (high request interval), even though the *absolute* energy is lower. The overhead of the duty cycling based native PSM of WLAN leads to the highest energy consumption (relative and absolute) in comparison to the wake-up based protocols. The 802.11ba scenarios have a higher power consumption than the LPD based ones. Especially with many nodes in the broadcast scenario it can be seen that the energy consumption increases due to the increased time it takes the nodes to gain access to the channel when using the DCF of WLAN. If many nodes are woken up at the same time by the AP, they try to access the channel at the same time which causes collisions, backoff waiting times and retransmissions. The CCCA approach

[34] As we want to quantify the difference in energy consumption caused by the protocols, we used the same values for all three protocols. This ensures a fair comparison between the protocols and ignores the differences caused by the different main radios. If we would simulate the LPD approach using the energy consumption of the nRF24L01 chip, the results for LPD would be even better.

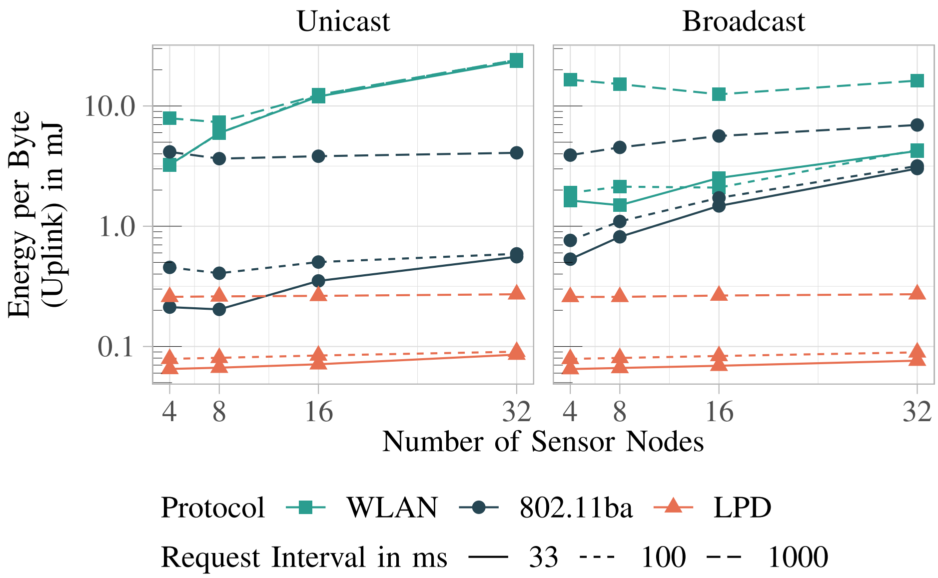


FIGURE 5.10 – Energy required to transmit one byte from the nodes to the AP. This includes overhead caused by the underlying communication protocol.

of the LPD protocol in conjunction with the collision free TDMA scheduling reduces the complexity of the uplink channel access which leads to a reduced energy consumption. Each node has to enable the main radio *only* during their timeslot, independent of the number of nodes or the request mode. The problem of an inefficient channel access has also been investigated in [76] where a solution based on 802.11ax Multi User Orthogonal Frequency Division Multiple Access (MUOFDMA) is proposed.

In the scenario with a request interval of 33 ms and 32 nodes the LPD protocol has an energy efficiency of 0.07 mJ/B which is only 1.6% of the energy required by the WLAN protocol (4.22 mJ/B) and 2.3% of the energy required by 802.11ba (3.01 mJ/B).

5.3.3 Coexistence with other WLAN Devices

The channel access described in Section 5.1.1 is designed in a way that is compatible to legacy WLAN devices. The AP first gains access to the channel using the DCF and then sends a WLAN header with the *duration* field set to a sufficiently large value. This ensures that other WLAN devices do not interfere with the following LPD packet and the uplink phase. However, this also means that during that time no other WLAN device can transmit data, hence reducing

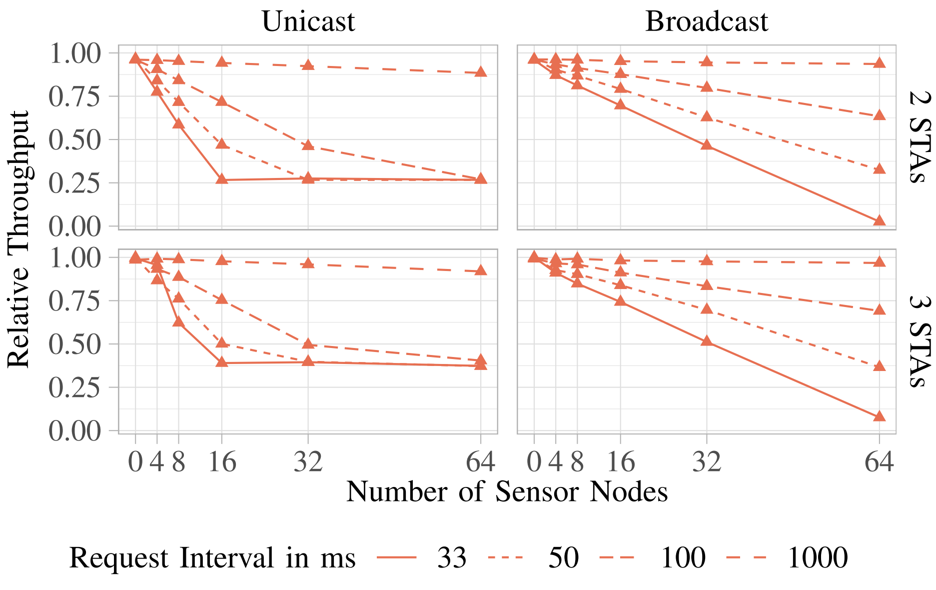


FIGURE 5.11 – Relative throughput of application: If no LPD devices are present the throughput is 1 (absolute value: 26.5 Mbit/s).

the throughput of these devices. In this section we will investigate the severity of the influence of our LPD protocol on legacy devices.

In the simulation scenario we created for this, we placed an AP in the middle, surrounded by 0–64 LPD devices and 2–3 normal WLAN devices[35]. The simulation parameters are shown in Table 5.3. One or two WLAN devices (transmitter) send UDP packets to the receiving WLAN device as fast as possible, thus saturating the wireless channel. At the same time and using the same WLAN channel, the AP performs a discovery of nodes and collects data from

[35] The WLAN devices are independent of the AP. They are not associated with it but instead form an own network in the Ad-Hoc/IBSS mode)

Table 5.3 – Simulation parameters for the coexistence scenario.

Description	Values	Unit
Number of LPD nodes	0,4,8,16,32,64	nodes
Number of WLAN nodes	2,3	nodes
Request Mode	Broadcast, Unicast	
Request Interval	33, 50, 100, 1000	ms
Slot Length	500	µs
WLAN Phy Rate	65	Mbit/s

the nodes with different intervals (33, 50, 100, 1000 ms) using unicast and broadcast requests.

The relative throughput[36] is shown in Figure 5.11. The more requests are performed by the AP the more the throughput of the WLAN devices is reduced. The comparison between unicast and broadcast request shows three effects.

First, for scenarios with few nodes (≤ 16) the unicast approach causes a higher reduction in WLAN throughput than the broadcast approach. This is caused by the increased overhead of unicast requests where for each node an LPD packet has to be transmitted.

Second, for scenarios with more nodes (> 16) the unicast approach limits the throughput to nearly 25% (2 WLAN STAs: 1 sender, 1 receiver) or 30% (3 WLAN STAs: 2 sender, 1 receiver). The reason for that is that the AP has to gain access to the channel for each of the unicast requests using the DCF which gives other legacy WLAN devices a fair chance to access the channel as well. In the scenario with 2 WLAN devices, both, the AP and the transmitting WLAN STA compete for the channel, but the AP gets a larger fraction of the channel. This is caused by the low data rate of the LPD channel which results in a longer blocking of the channel and the well-known WLAN rate anomaly problem [84]. To exchange a 1500 B UDP packet it only takes 274 µs, while the AP reserves 500 µs for one LPD packet and the following uplink slot.

Third, when using the broadcast approach, the throughput of the WLAN devices drop close to zero with many nodes and a high request frequency. This is caused by the CCCA that has to reserve the channel for a very long period of time ($slotLength * numberOfNodes$). With 64 nodes in the scenario, the AP will block the channel for 32 ms, once it won the contention for the channel, leaving almost no chance to transmit packets for the WLAN devices. This shows an important limit of the LPD approach: Too frequent polling and blocking the channel for an excessive amount of time significantly reduces the throughput of legacy WLAN devices.

5.3.4 Application of LPD in WiFi-based Energy Harvesting

So far we only evaluated the performance of the communication protocols to transmit sensor data from the devices to the AP. In this chapter we will focus on the RF energy transfer from the AP to the devices and how we can increase the amount of harvested energy by using the LPD protocol. We extended the hardware prototype from Section 5.3.1 with an RF energy harvesting circuit that includes a PCB antenna, a diode based rectifier circuit, a voltage regulator

[36] The throughput is divided by the maximum achievable throughput when there is no interfering LPD device (26.5 Mbit/s).

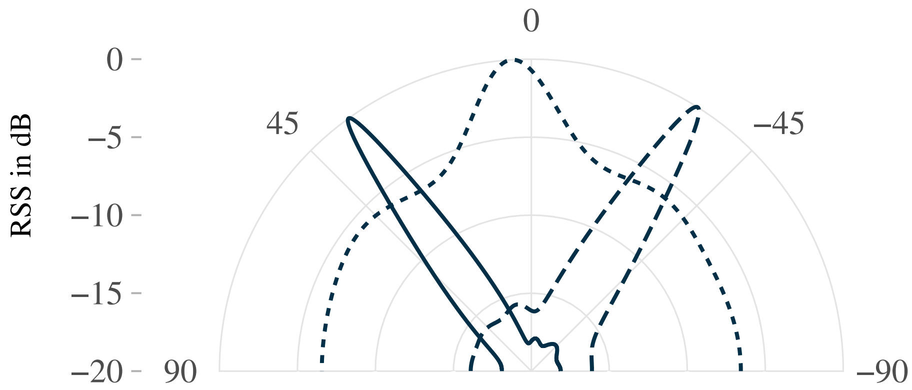


FIGURE 5.12 – Received Signal Strength (RSS) received by the antenna array of the AP for different node positions. The Ping signals from the nodes are used to perform Angle Of Arrival (AoA) measurements to update the beam direction when nodes move

and a capacitor, that stores the received energy. We use an antenna array to first estimate the Angle Of Arrival (AoA) of the signals emitted by the mobile nodes using the MUSIC algorithm [85] and then to use beamforming for an optimized transmission of energy as studied in [86]. In this previous work, the authors use a physical motion trigger to wake up the microcontroller. The device then sends out a *Ping* packet to the AP that can then re-calibrate the angle. This approach is static and does not adapt well to scenarios where the mobility of nodes varies.

We show how this static approach can be made more dynamic with the help of the LPD protocol. Once the AP has discovered a new node it will send a narrow beam of RF energy towards the nodes. To keep the angle in sync with the node's position, it will measure the AoA whenever it receives a packet from the node. If it detects a movement of the node it will update the angle. This, however, only works if the node sends out data often enough (i.e. if it is queried often) or does not move a lot. If the AP notices that a node moves too much between two angle updates, it can initiate an update of the position as often as required. It will send out an LPD packet to the node with a *Ping Request*. The node will then answer over the main radio with a *Ping Response* which can be used by the AP to update the node's angle. The pattern of the received signals at the AP for different node positions can be seen in Figure 5.12. The frequency of these Ping requests can be adapted for each node individually, depending on the mobility.

To show how this protocol increases the amount of energy that can be harvested at the nodes, we conducted the following experiment: We mounted

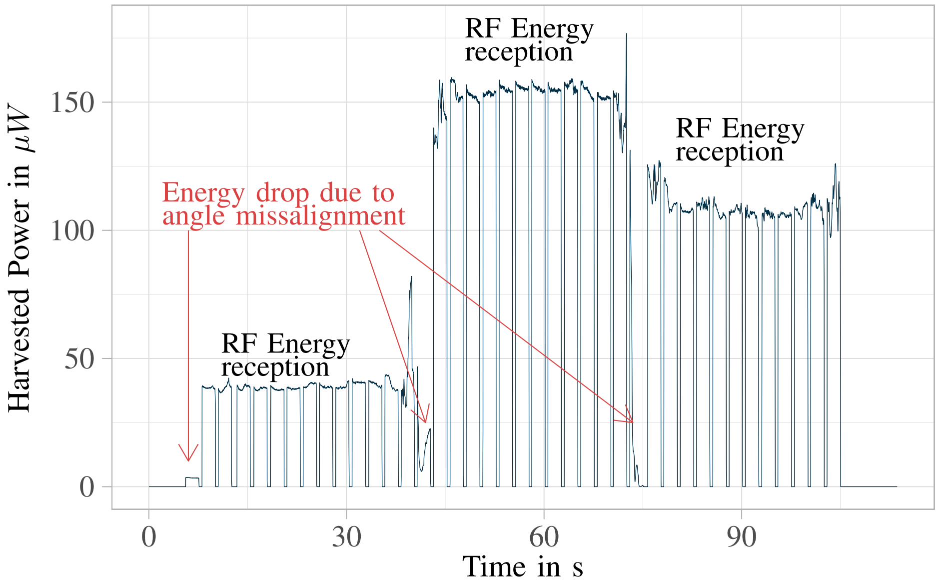


FIGURE 5.13 – If a node changes it's position, the AP can update the angle of the RF beam by sending out *PING* requests and receive the *PING* responses.

the mobile device with the harvesting circuit on a linear slider with a distance of around 90 cm from the antenna array. We then moved the device at three different positions and kept them there for 30 s. The AP will notice the change in position due to the Ping procedure described before and change the phase compensation of the antennas to steer the beam in the new direction. Figure 5.13 shows the power generated by the harvesting circuit over time. At the start of the experiment and after a change in position (at 7 s, 40 s and 75 s), the received energy is low because the beam emitted by the AP is not aligned with the node. After the AP learns the new position, the energy received increases significantly. At the beginning, the received power rises from 4 µW to 40 µW, after the first change of position it even rises from 20 µW to 150 µW.

5.4 Conclusion

In this chapter, we introduced a generalized concept for low power communication based on the WUR technology. The Low Power Downlink protocol enables a wide range of IoT applications that is compatible to existing WLAN systems. By extending the compatibility concept of first sending a WLAN header before transmitting the OOK signal, we cannot only prevent other STAs from interfering with the LPD packet, but can also ease the following uplink phase to be more energy efficient than existing solutions. The frame format introduced in Section 5.1.4 is flexible and efficient and can support a wide range of applications with little effort. We demonstrated the feasibility of our approach by using the LPUART module of modern microcontrollers to receive the LPD packets in an energy efficient way. We also showed how the concept can be combined with an RF energy harvesting approach, where conventional protocols are too power hungry, to optimize the wireless transmission of energy.

Chapter 6

Conclusion

ENERGY efficient communication is a building block for many applications ranging from environmental monitoring with a WSN to medical applications and smart homes. Current technologies require a trade-off between power consumption and performance. WLAN can offer high data rates but is not well suited for low power IoT devices. Protocols for IoT devices like the IEEE 802.15.4 based Zigbee have a much lower power consumption but only offer a low data rate. This trade-off is inherent to all duty-cycling based protocols.

Wake-Up Receivers can bridge this gap and provide low-power communication with low delay and high performance when needed. The main idea of these receivers is to enable devices to monitor the wireless channel continuously in order to be informed about an incoming transmission without using much energy. This way the nodes of a network can stay in a low power mode much longer and do not require any further synchronization. If a node wants to communicate with another node it can just send a wake-up signal, which will wake up the other node, and then exchange data over the normal radio. This way the communication delay is kept low and the protocol overhead is reduced.

In this thesis I demonstrated how a wake-up receiver can be built and used in different application scenarios. The concept of a Selective Wake-Up Receiver adds a flexible, sender-based addressing scheme which can be used to select the nodes that should be woken up in a fine granular way. This enhancement over a simple pattern based approach can be implemented without the help of a microcontroller and with little energy overhead. The better selectivity can save energy because unnecessary wake-ups can be prevented. In contrast to other MAC protocols, WUR based solutions can follow a simpler protocol which eases communication. In a highly dynamic network like in the BATS project where the network topology changes constantly, normal protocols, that rely on the agreement of a common schedule, cannot provide a robust communication between the nodes. Using a WUR based beaconing system in conjunction with a simple multiplexing approach is much better suited for such a challenging scenario. This way we where able to reliably download the collected data of the mobile sensors to gain more insight into the behavior of the bats that we tracked.

Also, in centralized networks like a standard IEEE 802.11 WLAN network WURs can improve the energy efficiency of the mobile, energy constrained devices. In contrast to the normal, duty cycling based Power Save Mode of WLAN, WUR based operation has a low energy consumption and a low delay at the same time. This is especially beneficial for IoT devices that currently have to use other technologies. With an ultra-low power WLAN option such devices could be integrated into existing environments much easier, because WLAN is already widely available. By using CTC to send the wake-up signals it would be even possible to use existing APs by just installing a new firmware. This way, battery operated sensors and high performance devices like a smartphone or laptop could use the same infrastructure which is a huge advantage to today's fragmented market. Since STAs that are in a sleep mode can be woken up by the AP is new data arrives, the buffer at the AP will not quickly fill up if the traffic shows a bursty pattern.

Finally, we developed an extension to the wake-up system that uses the secondary receiver in a more flexible way. Instead of using it just to wake up a node remotely, we demonstrated how we can transmit arbitrary data over this Low Power Downlink (LPD) and receive it at the mobile nodes in an energy efficient way. This enables many new application scenarios like battery-less, wearable devices that can communicate with a central AP in an energy efficient way. By extending the approach from the IEEE 802.11ba standard, we can achieve compatibility with existing WLAN systems and simplify the channel access for the energy constrained nodes.

While the concept of WURs has already proven to decrease energy consumption of wireless devices significantly, there are some open issues that are not yet fully understood and resolved. The most important one is that most WUR implementations have a much lower sensitivity than normal radios which are based on the superheterodyne receiver architecture. This limits the wake-up range and therefore the communication range. This problem, however, is already tackled in current research. While WUR chips with high sensitivity (< -70 dBm) and low power consumption are not yet available on the market, there exists some prototypes and papers that show that this is feasible. Another problem is the low data rate of the wake-up signal. If data is sent with a low rate, the resulting signal will have a longer duration than high data rate signals. This has a negative influence on other nodes in the network because the wake-up signals block the channel for other communication. It is therefore advisable to limit the amount of data sent over such a slow link as much as possible. With modern IEEE 802.11ax WLAN the radio resources can be scheduled in a more flexible way. This would make it possible to just use a fraction of the available bandwidth for the wake-up signal and use the rest for normal, high speed WLAN communication.

The vision of a closely connected world requires efficient wireless communication in order to connect energy constrained devices with each other. A secondary, low power receiver, used as a Wake-Up Receiver or general purpose Low Power Downlink, has a huge potential to make this vision come true. It implements the otherwise energy hungry idle listening in an energy efficient way, thus saving energy on the mobile devices while having a better performance than duty cycling based protocols. Since WURs are an addition to normal radio communication, they can be integrated into existing systems in a way compatible to legacy devices.

List of Figures

List of Tables

Bibliography

[1] K. Hafner and M. Lyon, Arpa Kadabra oder die Geschichte des Internet. Heidelberg, Germany: dpunkt.verlag, 2000.

[2] N. Abramson, "The Alohanet - surfing for wireless data [History of Communications]," *IEEE Communications Magazine (COMMAG)*, vol. 47, no. 12, pp. 21–25, Dec. 2009. DOI: 10.1109/mcom.2009.5350363.

[3] E.-S. Jung and N. H. Vaidya, "A Power Control MAC Protocol for Ad Hoc Networks," *ACM/Kluwer Wireless Networks (WINET)*, vol. 11, no. 1-2, pp. 55–66, 2005.

[4] CC3200 SimpleLink™Wi-Fi® and Internet-of-Things Solution, a Single-Chip Wireless MCU, CC3200, Texas Instruments, Feb. 2015.

[5] A. Bensky, Short-range Wireless Communication, 3rd ed. Elsevier, 2019, p. 462. DOI: 10.1016/c2017-0-02356-x.

[6] P. Huang, L. Xiao, S. Soltani, M. W. Mutka, and N. Xi, "The Evolution of MAC Protocols in Wireless Sensor Networks: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 101–120, Feb. 2013. DOI: 10.1109/SURV.2012.040412.00105.

[7] M. Mutschlechner, B. Li, R. Kapitza, and F. Dressler, "Using Erasure Codes to Overcome Reliability Issues in Energy-Constrained Sensor Networks," in *11th IEEE/IFIP Conference on Wireless On demand Network Systems and Services (WONS 2014)*, Obergurgl, Austria: IEEE, Apr. 2014, pp. 41–48. DOI: 10.1109/WONS.2014.6814720.

[8] J. Blobel, C. Sommer, and F. Dressler, "Protocol Options for Low Power Sensor Network MAC using Wake-up Receivers with Duty Cycling," in *IEEE International Conference on Communications (ICC 2016)*, Kuala Lumpur, Malaysia: IEEE, May 2016, pp. 3925–3930. DOI: 10.1109/ICC.2016.7511318.

[9] J. Blobel, J. Krasemann, and F. Dressler, "An Architecture for Sender-based Addressing for Selective Sensor Network Wake-Up Receivers," in *17th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2016)*, Coimbra, Portugal: IEEE, Jun. 2016. DOI: `10.1109/WoWMoM.2016.7523516`.

[10] J. Blobel and F. Dressler, "Sender-Triggered Selective Wake-Up Receiver for Low-Power Sensor Networks," in *36th IEEE Conference on Computer Communications (INFOCOM 2017), Demo Session*, Atlanta, GA: IEEE, May 2017. DOI: `10.1109/INFCOMW.2017.8116522`.

[11] J. Blobel, F. Menne, D. Yu, X. Cheng, and F. Dressler, "Low-power and Low-delay WLAN using Wake-up Receivers," *IEEE Transactions on Mobile Computing (TMC)*, Oct. 2020. DOI: `10.1109/TMC.2020.3030313`.

[12] J. Blobel, V. H. Tran, A. Misra, and F. Dressler, "Low-Power Downlink for the Internet of Things using IEEE 802.11-compliant Wake-Up Receivers," in *40th IEEE International Conference on Computer Communications (INFOCOM 2021)*, Virtual Conference: IEEE, May 2021.

[13] Q.-H. Nguyen, J. Blobel, and F. Dressler, "Energy Consumption Measurements as a Basis for Computational Offloading for Android Smartphones," in *14th IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (EUC 2016)*, Paris, France: IEEE, Aug. 2016. DOI: `10.1109/CSE-EUC-DCABES.2016.157`.

[14] F. Klingler, J. Blobel, and F. Dressler, "Agriculture meets IEEE 802.11p: A Feasibility Study," in *15th IEEE International Symposium on Wireless Communication Systems (ISWCS 2018)*, Lisbon, Portugal: IEEE, Aug. 2018. DOI: `10.1109/ISWCS.2018.8491239`.

[15] J. Blobel, "Energy Efficient Communication using Wake-up Receivers," in *International Conference on Networked Systems (NetSys 2019), PhD Forum*, Munich, Germany, Mar. 2019.

[16] F. Dressler, M. Mutschlechner, M. Nabeel, and J. Blobel, "Ultra Low-Power Sensor Networks for Next Generation Wildlife Monitoring," in *11th IEEE International Conference on Communication Systems and Networks (COMSNETS 2019)*, Bengaluru, India: IEEE, Jan. 2019. DOI: `10.1109/COMSNETS.2019.8711475`.

[17] M. S. Amjad, G. S. Pannu, A. Memedi, M. Nabeel, J. Blobel, F. Missbrenner, and F. Dressler, "A Flexible Real-Time Software-based Multi-Band Channel Sounder," in *31st IEEE Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2020)*, Virtual

Conference: IEEE, Aug. 2020. DOI: 10.1109/PIMRC48278.2020.9217369.

[18] C. F. Chiasserini and R. R. Rao, "A distributed power management policy for wireless ad hoc networks," in *IEEE Wireless Communications and Networking Conference (WCNC 2000)*, Chicago, IL: IEEE, Sep. 2000, pp. 1209–1213. DOI: 10.1109/WCNC.2000.904803.

[19] M. Huber, S. von der Mark, N. Angwafo, and G. Boeck, "Ultra low power wakeup circuits for pico cell networks, a conceptional view," in *European Workshop on Wireless Sensor Networks (EWSN 2004)*, Berlin, Germany: Springer, Jan. 2004.

[20] J. Ansari, D. Pankin, and P. Mähönen, "Radio-Triggered Wake-ups with Addressing Capabilities for extremely low power sensor network applications," in *19th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2008)*, Cannes, France: IEEE, Sep. 2008. DOI: 10.1109/PIMRC.2008.4699501.

[21] L. Gu and J. A. Stankovic, "Radio-triggered wake-up for wireless sensor networks," *Real-time Systems*, vol. 29, pp. 157–182, 2005.

[22] S. von der Mark, R. Kamp, M. Huber, and G. Boeck, "Three stage wakeup scheme for sensor networks," in *SBMO/IEEE MTT-S International Conference on Microwave and Optoelectronics*, Brasília, Brazil: IEEE, Jul. 2005, pp. 205–208. DOI: 10.1109/IMOC.2005.1579978.

[23] J. Oller, I. Demirkol, J. Casademont, J. Paradells, G. Gamm, and L. Reindl, "Performance Evaluation and Comparative Analysis of SubCarrier Modulation Wake-up Radio Systems for Energy-Efficient Wireless Sensor Networks," *MDPI Sensors*, vol. 14, no. 1, pp. 22–51, Dec. 2013. DOI: 10.3390/s140100022.

[24] I. Demirkol, C. Ersoy, and E. Onur, "Wake-up receivers for wireless sensor networks: benefits and challenges," *IEEE Wireless Communications*, vol. 16, no. 4, pp. 88–96, Aug. 2009. DOI: 10.1109/MWC.2009.5281260.

[25] J. Oller, I. Demirkol, J. Paradells, J. Casademont, and W. B. Heinzelman, "Time-Knocking: A novel addressing mechanism for wake-up receivers," in *8th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2012)*, Barcelona, Spain: IEEE, Oct. 2012, pp. 268–275. DOI: 10.1109/WiMOB.2012.6379086.

[26] S. Ishida, T. Takiguchi, S. Saruwatari, M. Minami, and H. Morikawa, "Evaluation of a wake-up wireless module with Bloom-filter-based ID matching," in *8th Asia-Pacific Symposium on Information and Telecommunication Technologies (APSITT 2010)*, Kuching, Malaysia: IEEE, Jun. 2010, pp. 1–6.

[27] H. Milosiu, F. Oehler, M. Eppel, D. Fruhsorger, S. Lensing, G. Popken, and T. Thones, "A 3-μW 868-MHz wake-up receiver with -83 dBm sensitivity and scalable data rate," in *2013 European Conference on Solid-State Circuits (ESSCIRC)*, Bucharest, Romania, Sep. 2013. DOI: 10.1109/esscirc.2013.6649154.

[28] C. Petrioli, D. Spenza, P. Tommasino, and A. Trifiletti, "A Novel Wake-Up Receiver with Addressing Capability for Wireless Sensor Nodes," in *2014 IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS 2014)*, Marina del Rey, CA: IEEE, May 2014. DOI: 10.1109/dcoss.2014.9.

[29] F. Dressler, S. P. Ripperger, M. Hierold, T. Nowak, C. Eibel, B. Cassens, F. Mayer, K. Meyer-Wegener, and A. Koelpin, "From Radio Telemetry to Ultra-Low-Power Sensor Networks: Tracking Bats in the Wild," *IEEE Communications Magazine (COMMAG)*, vol. 54, no. 1, pp. 129–135, Jan. 2016. DOI: 10.1109/MCOM.2016.7378438.

[30] F. Dressler, M. Mutschlechner, B. Li, R. Kapitza, S. P. Ripperger, C. Eibel, B. Herzog, T. Hönig, and W. Schröder-Preikschat, "Monitoring Bats in the Wild: On Using Erasure Codes for Energy-Efficient Wireless Sensor Networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 12, no. 1, Feb. 2016. DOI: 10.1145/2875426.

[31] D. MacKay, "Fountain codes," *IEE Proceedings - Communications*, vol. 152, no. 6, 1062–1068(6), Dec. 2005. DOI: 10.1049/ip-com:20050237.

[32] A. Varga, "The OMNeT++ Discrete Event Simulation System," in *European Simulation Multiconference (ESM 2001)*, Prague, Czechia, Jun. 2001.

[33] A. Köpke, M. Swigulski, K. Wessel, D. Willkomm, P. K. Haneveld, T. Parker, O. Visser, H. S. Lichte, and S. Valentin, "Simulating Wireless and Mobile Networks in OMNeT++ – The MiXiM Vision," in *1st ACM/ICST International Conference on Simulation Tools and Techniques for Communications, Networks and Systems (SIMUTools 2008), 1st ACM/ICST International Workshop on OMNeT++ (OMNeT++ 2008)*, Marseille, France: ACM, Mar. 2008.

[34] F. Dressler, B. Bloessl, M. Hierold, C.-Y. Hsieh, T. Nowak, R. Weigel, and A. Koelpin, "Protocol Design for Ultra-Low Power Wake-Up Systems for Tracking Bats in the Wild," in *IEEE International Conference on Communications (ICC 2015)*, London, United Kingdom: IEEE, Jun. 2015, pp. 6345–6350. DOI: 10.1109/ICC.2015.7249335.

[35] I. Demirkol, C. Ersoy, and F. Alagoz, "MAC Protocols for Wireless Sensor Networks: a Survey," *IEEE Communications Magazine (COMMAG)*, vol. 44, no. 4, pp. 115–121, Apr. 2006. DOI: 10.1109/MCOM.2006.1632658.

[36] I. Dietrich and F. Dressler, "On the Lifetime of Wireless Sensor Networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 5, no. 1, pp. 1–39, Feb. 2009. DOI: 10.1145/1464420.1464425.

[37] W. Ye, J. Heidemann, and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," in *21st IEEE Conference on Computer Communications (INFOCOM 2002)*, New York City, NY: IEEE, Jun. 2002, pp. 1567–1576. DOI: 10.1109/INFCOM.2002.1019408.

[38] W. Ye, F. Silva, and J. Heidemann, "Ultra-low duty cycle MAC with scheduled channel polling," in *4th ACM Conference on Embedded Networked Sensor Systems (SenSys 2006)*, Boulder, CO: ACM, Nov. 2006, pp. 321–334. DOI: 10.1145/1182807.1182839.

[39] W. Ye, J. Heidemann, and D. Estrin, "Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 12, no. 3, pp. 493–506, Jun. 2004. DOI: 10.1109/TNET.2004.828953.

[40] T. van Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *1st ACM Conference on Embedded Networked Sensor Systems (SenSys 2003)*, Los Angeles, CA: ACM, Nov. 2003, pp. 171–180. DOI: 10.1145/958491.958512.

[41] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, "Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks," *ACM/Springer Wireless Networks (WINET)*, vol. 12, no. 1, pp. 63–78, Feb. 2006. DOI: 10.1007/s11276-006-6151-z.

[42] J. Polastre, J. Hill, and D. Culler, "Versatile Low Power Media Access for Wireless Sensor Networks," in *2nd ACM Conference on Embedded Networked Sensor Systems (SenSys 2004)*, Baltimore, MD: ACM, Nov. 2004, pp. 95–107. DOI: 10.1145/1031495.1031508.

[43] C. C. Enz, A. El-Hoiydi, J.-D. Decotignie, and V. Peiris, “WiseNET: An Ultralow-Power Wireless Sensor Network Solution,” *Computer*, vol. 37, no. 8, pp. 62–70, Aug. 2004. DOI: 10.1109/MC.2004.109.

[44] L. Tang, Y. Sun, O. Gurewitz, and D. Johnson, “PW-MAC: An energy-efficient predictive-wakeup MAC protocol for wireless sensor networks,” in *30th IEEE Conference on Computer Communications (INFOCOM 2011)*, Shanghai, China: IEEE, Apr. 2011, pp. 1305–1313. DOI: 10.1109/INFCOM.2011.5934913.

[45] F. Z. Djiroun and D. Djenouri, “MAC Protocols With Wake-Up Radio for Wireless Sensor Networks: A Review,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 587–618, Feb. 2017. DOI: 10.1109/COMST.2016.2612644.

[46] M. J. Miller and N. H. Vaidya, “A MAC Protocol to Reduce Sensor Network Energy Consumption Using a Wakeup Radio,” *IEEE Transactions on Mobile Computing (TMC)*, vol. 4, no. 3, pp. 228–242, May 2005. DOI: 10.1109/TMC.2005.31.

[47] R. Piyare, A. L. Murphy, C. Kiraly, P. Tosato, and D. Brunelli, “Ultra Low Power Wake-Up Radios: A Hardware and Networking Survey,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2117–2157, Oct. 2017. DOI: 10.1109/COMST.2017.2728092.

[48] J. Oller, I. Demirkol, J. Casademont, J. Paradells, G. U. Gamm, and L. Reindl, “Has Time Come to Switch From Duty-Cycled MAC Protocols to Wake-Up Radio for Wireless Sensor Networks?” *IEEE/ACM Transactions on Networking (TON)*, vol. 24, no. 2, pp. 674–687, Apr. 2016. DOI: 10.1109/TNET.2014.2387314.

[49] K. Chowdhury, N. Nandiraju, D. Cavalcanti, and D. Agrawal, “CMAC - A multi-channel energy efficient MAC for wireless sensor networks,” in *IEEE Wireless Communications and Networking Conference (WCNC 2006)*, vol. 2, Las Vegas, NV: IEEE, Apr. 2006, pp. 1172–1177. DOI: 10.1109/WCNC.2006.1683636.

[50] C. Guo, L. C. Zhong, and J. Rabaey, “Low power distributed MAC for ad hoc sensor radio networks,” in *IEEE Global Telecommunications Conference (GLOBECOM 2001)*, vol. 5, San Antonio, TX: IEEE, Nov. 2001, pp. 2944–2948. DOI: 10.1109/GLOCOM.2001.965967.

[51] F. Sutton, R. D. Forno, J. Beutel, and L. Thiele, “BLITZ: Low Latency and Energy-Efficient Communication for Event-Triggered Wireless Sensing Systems,” *ACM Transactions on Sensor Networks (TOSN)*, vol. 15, no. 2, pp. 1–38, Mar. 2019. DOI: 10.1145/3309702.

[52] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," *IEEE Communications Magazine (COMMAG)*, vol. 40, no. 8, pp. 102–114, Aug. 2002. DOI: 10.1109/MCOM.2002.1024422.

[53] P. Lin, C. Qiao, and X. Wang, "Medium access control with a dynamic duty cycle for sensor networks," in *IEEE Wireless Communications and Networking Conference (WCNC 2004)*, Atlanta, GA: IEEE, Mar. 2004. DOI: 10.1109/wcnc.2004.1311671.

[54] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, Oct. 2015. DOI: 10.1109/COMST.2015.2444095.

[55] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low Power Wide Area Networks: An Overview," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 855–873, 2017. DOI: 10.1109/comst.2017.2652320.

[56] IEEE, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," IEEE, Std 802.11-2016, Dec. 2016.

[57] X. Costa-Pérez, D. Camps-Mur, and T. Sashihara, "Analysis of the integration of IEEE 802.11e capabilities in battery limited mobile devices," *IEEE Wireless Communications*, vol. 12, no. 6, pp. 26–32, Dec. 2005. DOI: 10.1109/MWC.2005.1561942.

[58] X. Costa-Pérez and D. Camps-Mur, "IEEE 802.11E QoS and power saving features overview and analysis of combined performance," *IEEE Wireless Communications*, vol. 17, no. 4, pp. 88–96, Aug. 2010. DOI: 10.1109/MWC.2010.5547926.

[59] S.-L. Tsao and C.-H. Huang, "A survey of energy efficient MAC protocols for IEEE 802.11 WLAN," *Elsevier Computer Communications*, vol. 34, no. 1, pp. 54–67, 2011.

[60] H. Woesner, J.-P. Ebert, M. Schläger, and A. Wolisz, "Power-saving mechanisms in emerging standards for wireless LANs: the MAC level perspective," *IEEE Personal Communications, Special Edition on Power Saving*, vol. 5, no. 3, pp. 40–48, Jun. 1998. DOI: 10.1109/98.683738.

[61] E.-S. Jung and N. H. Vaidya, "Improving IEEE 802.11 power saving mechanism," *ACM/Springer Wireless Networks (WINET)*, vol. 14, no. 3, pp. 375–391, Jun. 2008. DOI: 10.1007/s11276-006-0726-6.

[62] D. C. Mur, X. Costa-Pérez, and S. S. Ribes, "An adaptive solution for Wireless LAN distributed power saving modes," *Elsevier Computer Networks (COMNET)*, vol. 53, no. 18, pp. 3011–3030, 2009. DOI: `10.1016/j.comnet.2009.07.010`.

[63] V. Baiamonte and C. F. Chiasserini, "Saving Energy during Channel Contention in 802.11 WLANs," *ACM/Springer Mobile Networks and Applications (MONET)*, vol. 11, no. 2, pp. 287–296, Apr. 2006.

[64] J.-R. Hsieh, T.-H. Lee, and Y.-W. Kuo, "Energy-efficient multi-polling scheme for wireless LANs," *IEEE Transactions on Wireless Communications (TWC)*, vol. 8, no. 3, pp. 1532–1541, Mar. 2009. DOI: `10.1109/TWC.2008.080554`.

[65] E. Rozner, V. Navda, R. Ramjee, and S. Rayanchu, "NAPman: Network-Assisted Power Management for Wifi Devices," in *8th ACM International Conference on Mobile Systems, Applications and Services (MobiSys 2010)*, San Francisco, CA: ACM, Jun. 2010, pp. 91–106. DOI: `10.1145/1814433.1814445`.

[66] J. Manweiler and R. R. Choudhury, "Avoiding the Rush Hours: WiFi Energy Management via Traffic Isolation," in *9th International Conference on Mobile Systems, Applications, and Services (MobiSys 2011)*, Bethesda, MD: ACM, Jun. 2011, pp. 253–266. DOI: `10.1145/1999995.2000020`.

[67] A. Azcorra, I. Ucar, F. Gringoli, A. Banchs, and P. Serrano, "μNap: Practical micro-sleeps for 802.11 WLANs," *Elsevier Computer Communications*, vol. 110, pp. 175–186, 2017. DOI: `10.1016/j.comcom.2017.06.008`.

[68] B. Balaji, B. R. Tamma, and B. S. Manoj, "A Novel Power Saving Strategy for Greening IEEE 802.11 Based Wireless Networks," in *IEEE Global Communications Conference (GLOBECOM 2010)*, Miami, FL: IEEE, Dec. 2010. DOI: `10.1109/GLOCOM.2010.5684071`.

[69] X. Zhang and K. G. Shin, "E-MiLi: Energy-Minimizing Idle Listening in Wireless Networks," *IEEE Transactions on Mobile Computing (TMC)*, vol. 11, no. 9, pp. 1441–1454, Sep. 2012. DOI: `10.1109/tmc.2012.112`.

[70] E. Shih, P. Bahl, and M. J. Sinclair, "Wake on Wireless: An Event Driven Energy Saving Strategy for Battery Operated Devices," in *8th Annual International Conference on Mobile Computing and Networking (MobiCom 2002)*, Atlanta, GA: ACM, Sep. 2002, pp. 160–171. DOI: `10.1145/570645.570666`.

[71] E. Alpman, A. Khairi, R. Dorrance, M. Park, V. S. Somayazulu, J. R. Foerster, A. Ravi, J. Paramesh, and S. Pellerano, “802.11g/n Compliant Fully Integrated Wake-Up Receiver With -72-dBm Sensitivity in 14-nm FinFET CMOS,” *IEEE Journal of Solid-State Circuits*, vol. 53, no. 5, pp. 1411–1422, May 2018. DOI: 10.1109/JSSC.2018.2817603.

[72] IEEE, “P802.11ba/D2.0 - IEEE Draft Standard for Information Technology–Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks–Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment: Wake-up radio operation,” IEEE, Std 802.11ba-2019, Jan. 2019.

[73] P. Serrano, A. Garcia-Saavedra, G. Bianchi, A. Banchs, and A. Azcorra, “Per-Frame Energy Consumption in 802.11 Devices and Its Implication on Modeling and Design,” *IEEE/ACM Transactions on Networking (TON)*, vol. 23, no. 4, pp. 1243–1256, Aug. 2015. DOI: 10.1109/TNET.2014.2322262.

[74] H. Lei and A. A. Nilsson, “Queuing Analysis of Power Management in the IEEE 802.11 Based Wireless LANs,” *IEEE Transactions on Wireless Communications (TWC)*, vol. 6, no. 4, pp. 1286–1294, Apr. 2007. DOI: 10.1109/TWC.2007.348325.

[75] G. Bianchi, A. Di Stefano, C. Giaconia, L. Scalia, G. Terrazzino, and I. Tinnirello, “Experimental assessment of the backoff behavior of commercial IEEE 802.11b network cards,” in *26th IEEE Conference on Computer Communications (INFOCOM 2007)*, Anchorage, AK: IEEE, May 2007, pp. 1181–1189. DOI: 10.1109/INFCOM.2007.141.

[76] H. Hong, Y. Y. Kim, R. Y. Kim, S.-H. Hwang, and Seungkeun Park, “A novel low power WLAN operation scheme for multiple wake-up receivers,” in *2017 IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, NV: IEEE, Jan. 2017, pp. 334–335. DOI: 10.1109/ICCE.2017.7889343.

[77] Z. Li and T. He, “WEBee: Physical-Layer Cross-Technology Communication via Emulation,” in *23rd ACM International Conference on Mobile Computing and Networking (MobiCom 2017)*, Snowbird, UT: ACM, Oct. 2017, pp. 2–14. DOI: 10.1145/3117811.3117816.

[78] M. Schulz, J. Link, F. Gringoli, and M. Hollick, “Shadow Wi-Fi: Teaching Smartphones to Transmit Raw Signals and to Extract Channel State Information to Implement Practical Covert Channels over Wi-Fi,” in *16th ACM International Conference on Mobile Systems, Applications, and*

Services (MobiSys 2018), Munich, Germany: ACM, Jun. 2018. DOI: 10.1145/3210240.3210333.

[79] Ultra-low-power ARM Cortex-M0+ MCU with 64 Kbytes Flash, 32 MHz CPU, USB, LCD, STM32L053, Rev. 9, STMicroelectronics, Jun. 2019.

[80] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, "Long-range communications in unlicensed bands: the rising stars in the IoT and smart city scenarios," *IEEE Wireless Communications*, vol. 23, no. 5, pp. 60–67, Oct. 2016. DOI: 10.1109/mwc.2016.7721743.

[81] C. Law, K. Lee, and K.-Y. Siu, "Efficient memoryless protocol for tag identification (extended abstract)," in *4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM 2000)*, Boston, MA: ACM, Aug. 2000. DOI: 10.1145/345848.345865.

[82] J. Ryu, H. Lee, Y. Seok, T. T. Kwon, and Y. Choi, "A Hybrid Query Tree Protocol for Tag Collision Arbitration in RFID systems," in *IEEE International Conference on Communications (ICC 2007)*, Glasgow, Scotland: IEEE, Jun. 2007. DOI: 10.1109/icc.2007.991.

[83] WARP Project, http://warpproject.org, Accessed: 2020-08-14.

[84] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, "Performance Anomaly of 802.11b," in *22nd IEEE Conference on Computer Communications (INFOCOM 2003)*, vol. 2, San Francisco, CA: IEEE, Mar. 2003, pp. 836–843. DOI: 10.1109/INFCOM.2003.1208921.

[85] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, Mar. 1986. DOI: 10.1109/tap.1986.1143830.

[86] V. H. Tran, A. Misra, J. Xiong, and R. K. Balan, "WiWear: Wearable Sensing via Directional WiFi Energy Harvesting," in *17th IEEE International Conference on Pervasive Computing and Communications (PerCom 2019)*, Kyoto, Japan: IEEE, Mar. 2019. DOI: 10.1109/percom.2019.8767406.

Appendices

Appendix A

SWUR Circuit Design

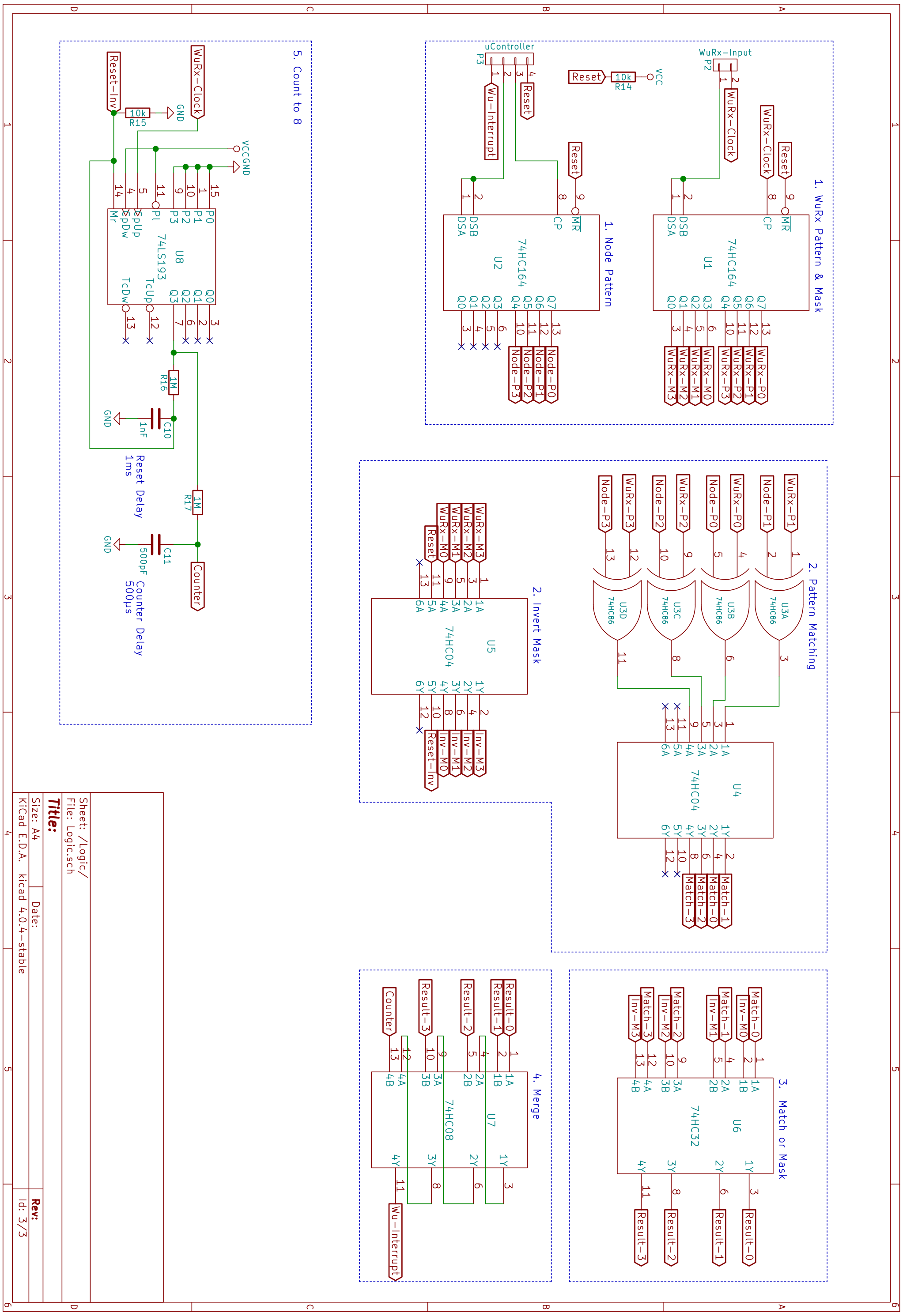


FIGURE A.1 – Circuit design of the matching logic of the SWUR

Appendix B

SWUR GNU Radio Transmitter

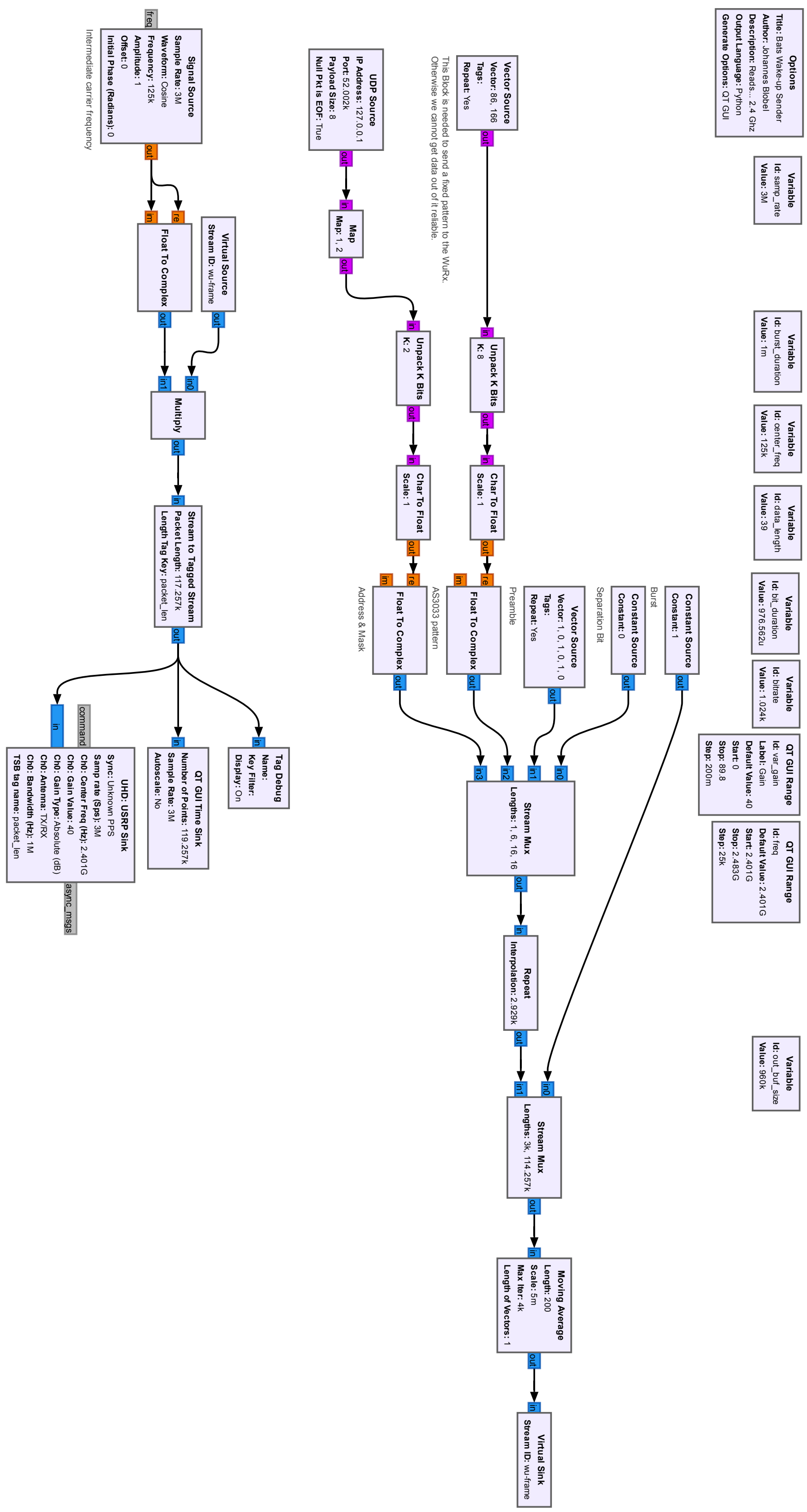


FIGURE B.1 – Flowgraph of the GNU Radio transmitter for wake-up signals

Acknowledgments

This dissertation is the result of my research over the past six years. During this time I have received a lot of support from family, friends and colleagues to whom I would like to express my gratitude!

I would like to thank my PhD supervisor Falko Dressler, from whom I was able to learn a lot. He always gave me enough freedom to follow my own ideas and supported me when I got stuck. The more I talked to other PhD students from other groups, the more I came to appreciate this style of leadership. Thank you, Falko, for your open manner and your help at eye level!

Furthermore, I experienced a lot of support from my colleagues. In the last few years, I have been able to work on many great projects with other PhD students, post-docs and students from the CCS/TKN team. We mounted antennas and measuring equipment on anything we could get our hands on, like cars, tractors, parking robots, bicycles, toy cars or bats. Many thanks for the wonderful experiences!

Finally, I would like to thank my family, especially Nele and my parents, and my friends. Since I started studying I have had many good times, but sometimes also difficult ones. You were there for me and built me up when I had doubts, and you have rejoiced with me when I have successfully completed another chapter in my live. Thank you for always being with me!

Danksagung

Diese Dissertation ist das Ergebnis meiner Forschung aus den vergangenen sechs Jahren. In dieser Zeit habe ich viel Unterstützung von Familie, Freunden und Kollegen erhalten, bei denen ich mich hiermit herzlich bedanken möchte!

Ich danke meinem Doktorvater Falko Dressler, von dem ich viel lernen durfte. Er gab mir stets genug Freiraum, um mich zu entfalten, und unterstützte mich, wenn ich mal nicht weiter wusste. Je mehr ich mich mit anderen Doktoranden aus anderen Gebieten austauschte, desto mehr lernte ich diesen Führungsstil zu schätzen. Vielen Dank, Falko, für deine offene Art und deine Hilfe auf Augenhöhe!

Des weiteren habe ich viel Unterstützung durch meine Fachgruppe erfahren. Ich durfte in den letzten Jahren an vielen tollen Projekten mit anderen Doktoranden, Post-Docs und SHKs aus dem CCS/TKN-Team arbeiten. Wir haben Antennen und Messtechnik auf alles montiert, was uns in die Hände fiel, zum Beispiel auf Autos, Traktoren, Park-Roboter, Fahrräder, Spielzeugautos oder Fledermäuse. Vielen Dank für die vielen schönen Erfahrungen!

Zuletzt möchte ich mich noch bei meiner Familie, insbesondere bei Nele und meinen Eltern, und meinen Freunden bedanken. Seitdem ich 2006 mit meinem Informatikstudium begonnen habe, hatte ich viele gute, manchmal aber auch schwierige Zeiten. Ihr wart für mich da und habt mich aufgebaut, wenn ich gezweifelt habe, und ihr habt euch mit mir gefreut, wenn ich wieder einen Abschnitt erfolgreich abgeschlossen hatte. Vielen Dank, dass ihr mich stets begleitet!