Lars Hanschke, Jan Heitmann, and Christian Renner. Challenges of WiFi-Enabled and Solar-Powered Sensors for Smart Ports. In Proceedings of the 4th International Workshop on Energy Harvesting & Energy Neutral Sensing Systems, ENSsys '16, Stanford, CA, USA, November 2016

Challenges of WiFi-Enabled and Solar-Powered Sensors for Smart Ports

Lars Hanschke
Research Group smartPORT
Hamburg University of Technology
lars.hanschke@tuhh.de

Jan Heitmann
University of Lübeck

Christian Renner
Research Group smartPORT
Hamburg University of Technology
christian.renner@tuhh.de

ABSTRACT

Increasing shipment volumes in the space-limited Hamburg port area demand a more efficient way of port organization. To achieve this goal, a key aspect of Hamburg's smart port initiative is to deploy wireless sensors and actuators to realize a cyber-physical port. However, the required large number of sensor and actuator devices raises two essential challenges. Firstly, external power provision infrastructure would be non-economic, inflexible, and therefore infeasible. Secondly, connecting common cyber-physical system devices to the Internet requires additional infrastructure and thus complexity due to the need for gateways. Our research tackles these challenges by investigating the applicability of novel, low-power IEEE 802.11 WiFi-based devices to enable communication with existing or planned WiFi access points while being reliably and autarkicly powered by solar power. As a first step, we analyzed the power consumption of a low-power IEEE 802.11 WiFi platform. Our measurements show promising results that allow us to propose methods for energy conservation and dimensioning of a miniature solar power supply.

CCS Concepts

•**Computer systems organization** → **Sensor networks;** •**Hardware** → **Renewable energy;** *Platform power issues;*

Keywords

WiFi, sensor node, solar harvesting, supercap, smart port, measurement

1. MOTIVATION

Current forecasts of the Hamburg Port Authority (HPA) [7] show an increase of container shipment volumes of 70% until 2025. The port of Hamburg as an example for other cities, such as Rotterdam or Le Havre, is a crucial factor for the economy in neighboring regions, employing more than 260 000 people. Since in most port areas spatial growth is restricted, an efficient use of the available space is inevitable. Since the volume of truck-transferred containers is expected to rise by 140%, observing environmental factors, e.g. carbon-hydroxide concentration, is vital. Another example is monitoring fine particles of diesel engines to ensure pollution-free working conditions. To solve these problems, a cost effective, environmentally friendly and flexible solution are WSNs.

However, most WSN protocols are based on the physical layer of the IEEE 802.15.4 standard, which is designed to cover only small distances and requires dedicated gateways [13] to connect the sensing environment to the Internet. Since several larger cities including Hamburg plan to offer public access to city-covering WiFi networks, utilizing small and low-power WiFi hardware is promising. This allows operation and connectivity with already deployed infrastructure without additional gateways, which reduces costs and enables direct access from the equipment of the maintenance staff to the sensor nodes; e.g. changes in configuration can be applied . Even concepts of augmented reality visualizing information are imaginable, e.g. displaying data flow or cable routing directly on WiFi-enabled tablet devices of workers.

Progress in the development of small WiFi nodes with increased energy efficiency makes IEEE 802.11 modules an alternative for sensing environments. Their power consumption is less than half compared to ten years ago, their footprint is reduced and their data rate is increased. Nevertheless, literature lacks of experience deploying them in sensing environments with limited energy resources. Measurements are therefore highly relevant for dimensioning proper resources. Here, the goal is to maintain the small footprint of sensing devices offering high flexibility; thus, using large batteries is no option. Moreover, the maintenance effort for large-scale, battery-powered sensor networks is unmanageable, so that an independent power supply is desirable. Especially in port areas, not all sectors are accessible at any time without interrupting ongoing processes; thus, changing batteries would mean a serious intrusion of work flow. A regenerative and environmentally friendly option are solar panels in combination with supercapacitors. Their benefit in outdoor scenarios has been investigated in [12] and [6]; thus, performance and achievable power resources are well understood. However, it remains unclear if WiFi nodes can be supplied by existing devices or if modifications are needed.

This paper examines the challenges of equipping WiFi nodes with an existing energy harvester based on a solar

ENSsys'16, November 16, 2016, Stanford, CA, USA

ISBN 978-1-4503-4532-3/16/11...$15.00

DOI: http://dx.doi.org/10.1145/2996884.2996887

Table 1: Datasheet Power Consumption

State	ATWINC1500 [3] (IEEE 802.11n)	AT86RF230 [2] (IEEE 802.15.4)
TRANSMIT	880 mW@17.8 dBm	51 mW@3 dBm
RECEIVE	297 mW	48 mW
DOZE	1 mW	4.5 mW
SLEEP	2.2 µW	0.06 µW

panel and supercapacitor to enable smart and seamless monitoring of larger areas including ports but also production plants or refineries. First, we inspect the given low-power possibilities of WiFi and compare them to IEEE 802.15.4 hardware. We point out possible benefits and fields of deployment. Second, we present results of power measurements for a WiFi node and discuss challenges tackling the improvement of power consumption. Third, we verify the power-compatibility of an existing energy harvester with an off-the-shelf WiFi node.

2. CHALLENGES AND TOOLS OF IEEE 802.11

Usually, WiFi systems are declared not suitable for wireless sensing purposes due to their high power consumption and focus on providing high data rates. To tackle these issues the research group for WiFi systems is developing the IEEE 802.11ah standard. It uses sub-1-GHz frequency bands to enlarge the coverage area and offers several MAC layer enhancements to reduce power consumption. At this point it is still unclear if future IEEE 802.11ah access points also support transmission in the 2.4 GHz band to avoid additional gateways. Moreover, the adoption of the new standard and its time frame are uncertain. Thus, we argue that relying on the established IEEE 802.11n standard is future-proof while solving the gateway problem.

2.1 Hardware Power Consumption

A brief glance at raw data sheet numbers of WiFi hardware and comparing them with IEEE 802.15.4 compatible hardware reveals their heavy power consumption, cf. Table 1. A typical sensor node radio chip, such as the Atmel 68RF230 radio chip on the IRIS mote, consumes only 50 mW during both transmission and reception of data packets. In contrast, low-power WiFi transceivers like the Atmel WINC1500A working on the Arduino/Genuino MKR1000 run at 880 mW transmit power and almost 300 mW receive power. These findings are even aggravated when considering that IEEE 802.15.4 radio chips require intelligent sleep mechanisms to achieve run times exceeding weeks (when powered by batteries) or to meet the power output of miniature energy harvesters (see Section 2.4). For IEEE 802.15.4, there are several MAC layer protocols, such as X-MAC [4], designed to provide low duty cycles and consumption. For IEEE 802.11, MAC layer adoptions with the same focus are rare, but we will point out in Section 2.2 that WiFi already allows for energy conservation by design.

However, comparing raw power consumption might be misleading. Taking the maximum data rates into account leads to a slightly different image. The depicted WiFi module delivers a 54 Mbit/s PHY layer data rate, yielding a raw energy efficiency of 61 Mbit/J. This is, in theory, more than ten times higher than the transceiver of the IRIS mote providing 4.9 Mbit/J (assuming a raw PHY rate of 250 kbit/s). Nevertheless, physical transmission of the data packet itself is not the only power consuming operation: Each transmission requires different phases of preparation, e.g. settling of the radio oscillator after leaving the sleep mode, with different power demands. Thus, the whole process of transmission has to be measured accurately to obtain valid predictions.

Additionally, typical transmission power of WiFi hardware is significantly higher (17.8 dBm compared to 3 dBm), which leads to higher transmission ranges. This makes multihop network structures potentially unnecessary, consequently decreasing radio usage, since no message forwarding is needed.

2.2 IEEE 802.11 MAC Layer and Consumption

A node's energy bill stems from its power consumption in all states (mainly sending, receiving, and sleeping) and the time spent in these. Therefore, we explore the MAC capabilities concerning downlink (communication flow from access point (AP) to wireless station (STA)) and uplink to identify mechanisms to save baseline energy. Before communication in the network is possible, each STA has to connect and authenticate to the AP by exchanging several data frames; thus, loosing the connection comes at the energy cost of transmitting and receiving multiple control messages.

2.2.1 Downlink

The standard description of IEEE 802.11 [9] mainly defines two power modes, the Constant Awake Mode (CAM) and the Power Saving Mode (PSM), whilst CAM is obviously unsuitable for the described usage scenario. Once a STA, e.g. a sensor node, decides to enter PSM, it informs the AP about its sleep duration and enters the doze state. During PSM, the STA has to wake up periodically to receive beacons containing the traffic indication map (TIM). In the TIM field, the AP announces unicast traffic, while the DTIM is used for broad- and multicast traffic. If the STA leaves PSM and recognizes buffered frames in the TIM, it sends a PS-POLL message to the AP to indicate its willingness to receive packets. As depicted in Fig. 1, a station can skip consequent beacons or TIM fields; in the shown case, the sleep interval is twice the beacon period. This allows a STA to save energy, without the need for an adaption of the whole network, e.g. increasing the beacon interval. To avoid disconnection during the sleep period, the connected STA must not stay in sleep mode longer than 65 s as defined in [9]. Before this maximum idle period runs out, a specific frame should be sent to avoid reconnection. Thus, the maximum sleep time is upper bounded by the maximum idle period. Obviously, the maximum sleep time has to be balanced against the delay criteria of the served application.

One advantage of beacons in WiFi is the synchronization of the network. Although the beacon interval is fixed, STAs can skip beacons in between according to their power save goals as long as they stay connected. However, CSMA/CA affects also beacons, which leads to possible longer waiting times. Compared to an unsynchronized protocol as X-MAC, the idle listening period can be reduced. In X-MAC, wakeup intervals can not be adjusted according to the needs of the nodes, since neighbored stations may rely on message forwarding. Thus, changing the own sleep interval has an unpredictable effect on the whole network communication flow.

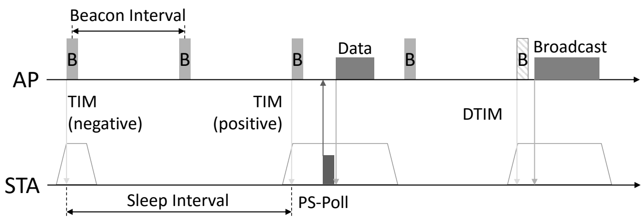

Figure 1: Sleep intervals in PSM span several beacon (B) intervals; stored unicast data is indicated in the TIM field, the station answers with PS-Poll message to receive data; the DTIM field indicates broadcast traffic transmitted directly afterwards.

2.2.2 *Uplink*

The popular carrier sense multiple access with collision avoidance (CSMA/CA) assures fair usage of the wireless medium. CSMA/CA demands senders to listen to the channel first before transmitting. In case of a busy channel, an exponentially increasing random backoff scheme is applied. The effect of an increasing delay is even amplified when the background traffic in the same network or in networks transmitting on the same channel increases [8]. Since the nodes stay awake during the backoff time, the power consumption of uplink is mainly determined by the number of transmission attempts. Thus, a large contention window of the backoff scheme leads to long waiting times leading to higher power consumption. Additionally, beacons are also affected by the CSMA/CA algorithm, thus reception of beacons might be delayed and consume more energy than originally intended. Our measurements cover this aspect by pointing out the influence on the overall power consumption.

In usage scenarios mainly dominated by uplink traffic, focus should lie on the adjustment of the contention window to reduce idle listening phases due to overhearing. This problem is treated as one of the major problems of uplink power consumption of energy harvesting devices in [11]. They propose to enable devices, running out of energy, to reduce the contention window size after a longer sleeping period, to prioritize their traffic decreasing the time spent in backoff period. Additionally, they introduce a controlled access for their deep sleep mechanism, letting devices enter the CSMA/CA phase only with a certain probability. Since the AP determines this probability, it allows to force several nodes to longer sleeping periods to avoid congestion in large networks. Since the main effect of possible contentions in the network are increasing delays, it is suitable for sensing applications with relaxed delay criteria.

2.3 Operating Modes

Popular WiFi modules for Internet of Things applications like the Arduino/Genuino MKR1000 offer access to simple sensing applications. The underlying WiFi hardware is configurable and offers several operating modes. Without hardware modifications, the devices can be configured to act as a simple WiFi station, as an access point spanning an own network or as part of a direct peer-to-peer connection without the need for existing infrastructure. The gained flexibility offers several usage scenarios. Dependent on the deployment area, multi-hop networks covering large distances can be established only by changing a few devices' operating modes. This might even happen autonomously dependent on signal quality, e.g. due to changing weather conditions.

2.4 Solar-Power Supply

To remove the energy bottleneck of wireless sensor networks, renewable power supplies (so-called energy harvesters) and algorithms to ensure perpetual operation have been proposed and investigated recently. In outdoor environments, such as ports, solar power is a good choice. A common miniature solar harvester with a supply voltage of 2.7 V and a 25 F to 200 F supercapacitor as energy buffer has been presented in [12]. It has been designed with particular focus to maintain the small dimensions the sensor node. This, however, comes at the cost of a relatively low power output; the harvester produces up to 140 mW in perfect conditions with an average of only about 5 mW in a partly shaded position throughout the year. Since maintaining the small footprint of the harvester is a main design goal to increase flexibility, power-saving techniques are required to ensure perpetual operation. A main challenge hence is to reduce the consumption of a WiFi-enabled sensor node without violating the specification of the WiFi standard, so that it can be powered from an solar-harvesting power supply similar to that in [12] while achieving the requested quality of service (e.g., data rate, latency). Therefore, our goal is to investigate whether and how this can be realized without modification of the harvester or, if this is not possible, to derive a node's power consumption and propose changes to the harvester.

3. CONSUMPTION MODEL

Most enhancements on power saving in WiFi aim at using adaptive PSM: applying large TIM and DTIM periods while having low traffic and decreasing when experiencing high loads [10]. Since most of the wireless sensor network usage scenarios have deterministic and known traffic, an adaption on changing load is not the major concern in this paper. The priority of power-saving techniques should lie on the downlink tweaking of the MAC layer parameters. Beacon and DTIM periods have a large influence; thus, they have to be investigated carefully assuring low duty cycles to enable solar-powered sensors. Since most examinations in literature focus on typical WiFi traffic like HTTP, which is fundamentally different to traffic occurring in sensor networks, the need for measurements of the influencing parameters rises.

Without adjustments, the required power P of a device, which enters PSM, wakes up for the beacons each beacon interval T_{BCN}, and transmit a data packet with duration T_{Tx} each transmit interval I_{Tx}, calculates as follows when assuming constant power over time:

$$P = P_{\mathrm{doze}} + \frac{T_{\mathrm{Rx}}}{T_{\mathrm{doze}}}\left(P_{\mathrm{Rx}} - P_{\mathrm{doze}}\right) + \frac{T_{\mathrm{Tx}}}{I_{\mathrm{Tx}}}\left(P_{\mathrm{Tx}} - P_{\mathrm{doze}}\right) . \quad (1)$$

Here, P_{doze}, P_{Rx}, and P_{Tx} denote the power consumption during doze/sleep, receive, and transmit state respectively. Furthermore, the time to receive one beacon is defined as

$$T_{\mathrm{Rx}} = \frac{L_{\mathrm{BCN}}}{R_{\mathrm{BCN}}} , \quad (2)$$

with L_{BCN} denoting the length of a beacon directly before transmission (including MAC and PHY header, and beacon frame body) and R_{BCN} the transmission data rate of a beacon. For the time to transmit a data packet,

$$T_{\mathrm{Tx}} = \frac{L_{\mathrm{Data}}}{R_{\mathrm{Data}}} \tag{3}$$

holds. Ensuring correct reception, usually $R_{\mathrm{BCN}} \leq R_{\mathrm{Data}}$ holds. As [9] defines many different PHY layers, the data packet length L_{Data} and even the data rate R_{Data} strongly varies dependent on preamble length, guard interval, modulation technique and others. Thus, power consumption presented in Section 4.3 takes only the resulting T_{Tx} into account.

Assuming a simple example without data transmission and typical values ($L_{\mathrm{BCN}} = 250\,\mathrm{B}$, $R_{\mathrm{BCN}} = 1\,\mathrm{Mbit/s}$) combined with the power consumption values of Table 1 yields $P = 7\,\mathrm{mW}$ for the common beacon interval $T_{\mathrm{BCN}} = 100\,\mathrm{ms}$. Doubling T_{BCN} decreases the average power consumption to $P = 4\,\mathrm{mW}$. This underlines the great influence of the beacon interval on power consumption. In contrast, adding data transmission of one packet with 1 kB each minute at 1 Mbit/s, only adds 0.1 mW at most. Nevertheless, increasing the beacon interval has to be balanced against the delay criteria of the network. While actuators require direct transmission of their packets, networks mainly consisting of sensors accept larger delays. The authors in [5] showed that LPL implementations with sleep intervals up to 500 ms work well in certain WSN applications; thus, the aimed beacon interval is in an appropriate range.

4. PRELIMINARY EVALUATION

In the following, we report on our first experiences with a WiFi-enabled sensor node, namely the Arduino/Genuino MKR1000 [1], evaluate its power consumption, and discuss the required dimensioning of the solar power supply.

4.1 Hardware

The Arduino/Genuino MKR1000 was initially released to offer a cost effective solution for Internet of Things applications. The dimensions of the package allow us to maintain the small footprint of the solar-powered sensor node, to be open for a large variety of application fields beyond smart ports. Combining the 32 bit low-power Cortex M0+ and Atmel WINC1500 WiFi chip supporting IEEE 802.11n, it builds a solid basis for our investigations. Furthermore, the USB connector provides the access to the serial programming port of the underlying WiFi chip. We removed the linear voltage regulator allowing us to access the supply voltage of the platform directly, which is needed to support the energy harvester. The supplied battery charger was not used during all measurements.

4.2 Measurement Setup

The MKR1000 is powered by a laboratory power supply. Current measurement is enabled with an INA139 measurement amplifier and a $3\,\Omega$, 1% precision (series) measuring shunt in the high-side path. This enables us to investigate the current drawn in the different states of both the Arduino hardware and the WiFi chip across their current range. We record the current and the supply voltage of the Arduino with an Agilent DSO-X 3014A. The output voltage of the power supply is a constant 3.3 V, so that the input voltage of the MKR1000 stays above 3 V while receiving and above 2.5 V when transmitting. We verified a proper function of the device in these conditions. We would like to point out that current consumption is barely affected by the supply voltage, but leave out the corresponding figure due to space constraints. However, in this preliminary evaluation we only discuss consumption of the sleep, idle, and receive states. We used the library *Wifi101* to configure the WiFi chip using the deep automatic power save mode (`M2M_PS_DEEP_AUTOMATIC`). We adjusted the listen interval for beacon reception to 102.4 ms and 1024 ms.

4.3 Power Consumption

First, we assessed the power consumption in different hardware states at 2.7 V, c.f. Section 4.4. When the Arduino sleeps and the WiFi module is powered off, we noted a power consumption of 1.16 mW. Switching the Arduino to idle mode without powering the WiFi module, leads to 23.4 mW. When the WiFi chip receives data and the Arduino is in sleep mode, the system consumes 233 mW. When the Arduino is in sleep mode and the WiFi chip is connected but idle, we noted 29.2 mW.

Section 3 shows that the time spent in the receive state to obtain beacons is a crucial factor for the overall power consumption. Since CSMA/CA is also applied for beacon transmission, the time T_{awake} to receive beacons varies. Figure 2 shows the distribution of awake times for a beacon interval of $T_{\mathrm{BCN}} = 102.4\,\mathrm{ms}$ in two traffic scenarios in an empty laboratory at our university. We created high traffic by downloading large files on the university WiFi network on the same channel as our setup to simulate interference from co-located networks. The figure shows that awake times are similarly distributed with a distinct peak in the bin from 10 ms to 15 ms. In the high traffic scenario, however, there is a notable amount of longer times needed to receive the beacons, leading to a higher power consumption. In particular, awake times of more than 100 ms—i.e., a missed beacon—increased from less than 1% to almost 2%.

When assessing power consumption, we faced several issues discussed in Section 4.5. Thus, we calculated power consumption based on the consumption per state and Eq. (1) (we will analyze the bug in our ongoing research). With a beacon period of 102.4 ms, we calculated a mean power consumption of $P_{\mathrm{highTraffic}} = 33.0\,\mathrm{mW}$ and $P_{\mathrm{lowTraffic}} = 30.0\,\mathrm{mW}$ in 10 experiment runs of 5 s each. This shows that with adjustments of the beacon period, e.g. $T_{\mathrm{BCN}} = 1024\,\mathrm{ms}$, the desired range in power consumption is achievable.

4.4 Solar Power Supply

We verified proper function of the MKR1000 slightly below its specification down to a supply voltage of 2.5 V with a simple sense-and-send application. This paves the ground for powering the MKR1000 with our harvester from [12] with a little safety margin regarding supply voltage. The datasheet of the TPS61220 DC/DC converter of our harvester states a minimum input voltage of 0.7 V (the cut-off voltage), so that 93% of the energy stored in the supercapacitor (rated at 2.7 V) could be used. Unfortunately, the power consumption during receive and send states of WiFi chips, such as the ATWINC1500, is up to 260 mW[1]. The critical issue here is that the supercapacitor voltage is decreasing during periods of no (solar) harvest, while the maximum output power of the DC/DC converter is also decreasing

[1]We measured a current of 96 mA at 2.7 V in 802.11g mode, higher figures in the datasheet correspond to 802.11b networks.

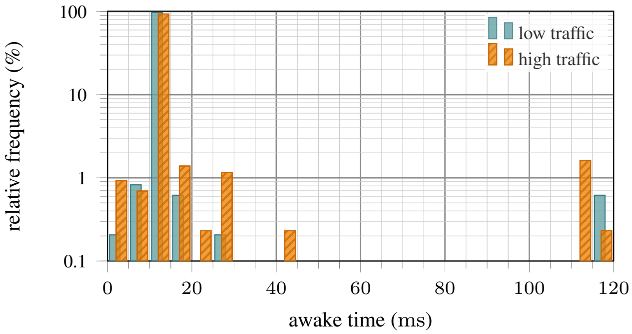

Figure 2: Distribution of awake times for beacon reception vs. background traffic of ten 5 s time windows with a beacon interval of 102.4 ms. Longer awake phases, e.g. between 20 ms and 45 ms, become more likely in higher background traffic. Note the logarithmic scale.

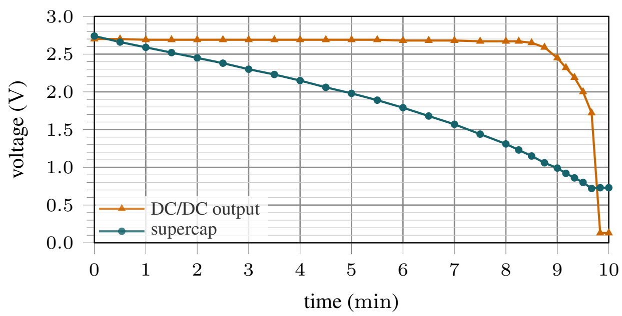

Figure 3: Discharging curve of a 50 F supercapacitor and DC/DC output voltage for a 32.9 Ω load, leading to a power consumption of 222 mW (at 2.7 V) similar to that during beacon reception of the ATWINC1500.

with input voltage. The actual cut-off voltage and hence the useable amount of stored energy therefore depends on the actual load, i.e., power consumption of the sensor node.

To analyze this aspect in more detail, we used a 32.9 Ω resistor as load to one of our harvesters and recorded the discharge trace, depicted in Fig. 3, of the supercapacitor. This setup closely resembles the consumption of an MKR1000 during reception. We charged the supercapacitor to 2.7 V and let it settle for several minutes with the supply connected, so that leakage and charge redistribution do not affect the discharge measurement. Due to the rather small charge currents of the solar panel, these effects will be small in practice. Our results reveal that the converter keeps the output voltage above 2.5 V for an input (supercapacitor) voltage down to 1.0 V with a 50 F supercapacitor. The amount of useable energy hence exceeds 86%. Better results are expected for larger capacitors. In conclusion, the MKR1000 can be powered by the harvester in 802.11g mode. In more power-hungry modes, however, hardware modification in terms of a different DC/DC converter will be required. As an example, a load of 730 mW (10 Ω @ 2.7 V) lead to a voltage drop below 2.5 V at the DC/DC output when the supercapacitor voltage had decreased to 2.37 V.

As discussed in Section 4.3, an overall power consumption (in connected state) is compatible with our harvester; e.g., reducing the beacon wake-up interval to 1024 ms reduces consumption to roughly 4 mW, including light, periodic data reporting. In this scenario, a fully-charged 50 F supercapacitor would provide enough energy to power the node for more than eight hours, cf. Fig. 3. Since the amount of energy stored in a capacitor scales linearly with capacity, a 200 F version would achieve 32 h. Even larger capacitors could be used; they would yet require a larger solar cell to allow for an entire charge cycle within a day. For example, our harvester can charge a 200 F supercapacitor in 5.8 h from 1.0 V to 2.7 V on a sunny day (30 mA solar current). Again, capacity has a linear influence on charging time. Due to these findings, we will explore the use of larger or multiple solar cells of the current size.

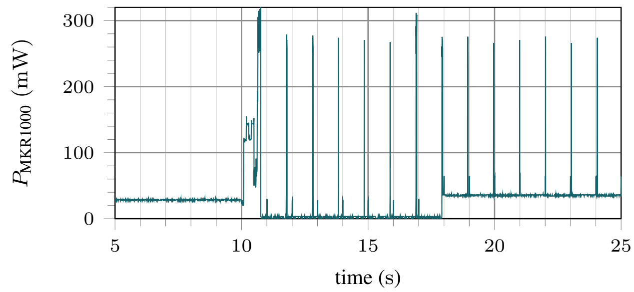

Figure 4: After association, the Arduino works as expected: periodically waking up for beacons and staying in sleep state in between. After a series of beacons, in this case seven, the Arduino ignores the sleep command and stays idle instead.

4.5 Hardware and Protocol Issues

The MKR1000 revealed unexpected behavior during our experiments. The first is depicted in Fig. 4 and likely a problem in the *WiFi101* library. After eleven seconds, the Arduino goes to sleep mode and triggers the power-save mode of the WiFi chip, only waking up periodically for beacon reception. After 18 s, however, the WiFi chip randomly enters the idle mode instead. This behavior appeared independently from the used power-save implementation of the libraries. The manual power save mode (`M2M_PS_MANUAL`) forces the WiFi chip to sleep, so that even beacons are ignored; the Arduino MCU has to wake up the chip periodically to avoid reconnection to the network. The wake up periods are chosen short, e.g. 102.4 ms, but instead of waking up once per beacon period, in some cases the WiFi chip enters a power consuming state even twice.

In IEEE 802.11, beacons are used by the stations to obtain information about pending traffic, synchronization offsets, beacon intervals and more. Receiving beacons is mandatory but the period between two received beacons is adjustable. Thus, missing one beacon is an issue, but the behavior afterwards should be chosen carefully. Figure 5 depicts how the Arduino MKR1000 reacts on missing beacons. Instead of returning to sleep mode after a time-out, it stays awake over a whole beacon interval to receive the next beacon. While this might be acceptable for devices with wired power supply, it is obviously not acceptable for solar-powered devices.

Another hint that the existing implementation lacks precise duty cycling is displayed in Fig. 6. While beacon reception occurs every 102.4 ms, another awake phase appears

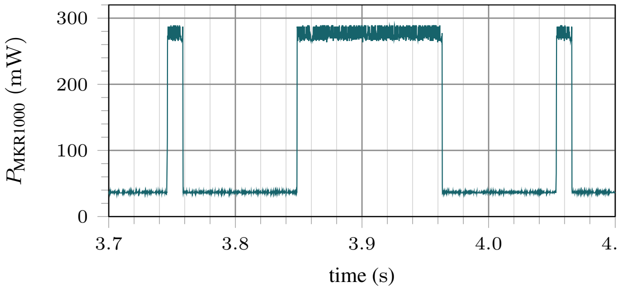

Figure 5: The absence of one beacon prevents the hardware from returning to a power saving mode. Instead, a whole beacon period is spent in the power consuming receive state. Note that the hardware stays idle between beacon reception due to the mentioned bug.

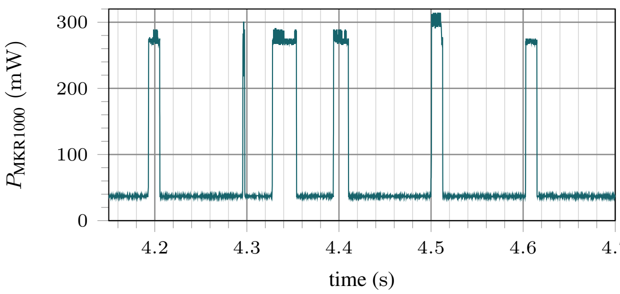

Figure 6: Instead of receiving beacons around 4.2 s **and** 102.4 ms **afterwards, the hardware wakes up shortly without beacon reception, returns to idle and wakes up again before the beacon interval expires. Additionally, the mentioned bug is also visible; thus the hardware remains idle instead of returning to sleep**

between 4.3 s and 4.4 s. At this point, it is unclear why the transceiver enters the receive mode, since no additional traffic exists. Unfortunately, the underlying library for the platform lacks the ability of monitoring the state of the WiFi chip, hence complicating effective debugging.

5. CONCLUSION AND FUTURE WORK

Modern IEEE 802.11 radios combine low power consumption with seamless integration in existing WiFi networks, hence being attractive for wireless sensing. Their remote and maintenance-free deployment, e.g. in smart ports, offers several benefits from low costs to a larger coverage area. Nevertheless, operation requires careful adjustment of the given power-saving techniques of the MAC. Our efforts show the potential of these networks, but at this point precise duty cycling suffers from inaccuracies of the implemented power-saving modes. Our measurements reveal that a consumption of 4 mW is achievable; but the identified hard- and software issues hamper ultra low-power consumption. While this may be solved by manufacturers in the future, the current software issues prevent in-field deployment. Generally, we demonstrated the power compatibility of a small-size, low-power energy harvester with the Arduino MKR1000 and comparable WiFi nodes. Next, we plan to derive consumption models for dynamic load adaptation, investigate the influence of WiFi uplink mechanisms on power consumption and develop sensing applications based on common Internet protocols. This will be integrated in our scheduled outdoor experiments with the solar harvester and a WiFi sensor node in a local port area.

6. REFERENCES

[1] Arduino LLC. Arduino/Genuino MKR1000 Product Description. https://www.arduino.cc/en/Main/ArduinoMKR1000. accessed: 2016/08/15.

[2] Atmel Corp. AT86RF230 Datasheet. http://www.atmel.com/images/doc5131.pdf. accessed: 2016/08/22.

[3] Atmel Corp. ATWINC1500 Datasheet. http://www.atmel.com/Images/Atmel-42502-SmartConnect-WINC1500-MR210PB_Datasheet.pdf. accessed: 2016/08/22.

[4] M. Buettner, G. V. Yee, E. Anderson, and R. Han. X-MAC: A Short Preamble MAC Protocol for Duty-cycled Wireless Sensor Networks. In *Proc. 4th Intl. Conf. on Embedded Networked Sensor Systems (SenSys)*, Boulder, Colorado, USA, 2006.

[5] M. Ceriotti, M. Corra, L. D'Orazio, R. Doriguzzi, D. Facchin, S. T. Guna, G. P. Jesi, R. L. Cigno, L. Mottola, A. L. Murphy, M. Pescalli, G. P. Picco, D. Pregnolato, and C. Torghele. Is There Light at the Ends of the Tunnel? Wireless Sensor Networks for Adaptive Lighting in Road Tunnels. In *Proc. 10th Intl. Conf. on Information Processing in Sensor Networks (IPSN)*, Stockholm, Sweden, April 2011.

[6] P. Dutta, J. Hui, J. Jeong, S. Kim, C. Sharp, J. Taneja, G. Tolle, K. Whitehouse, and D. Culler. Trio: Enabling Sustainable and Scalable Outdoor Wireless Sensor Network Deployments. In *Proc. 5th Intl. Conf. on Information Processing in Sensor Networks (IPSN)*, Nashville, Tennessee, USA, 2006.

[7] Free and Hanseatic City of Hamburg — State Ministry of Economic Affairs, Transport and Innovation, Hamburg Port Authority. The Port Development Plan to 2025. http://www.hamburg-port-authority.de/de/presse/broschueren-und-publikationen/Documents/port-development-plan2025.pdf, Oct. 2012.

[8] Y. He, R. Yuan, X. Ma, and J. Li. The IEEE 802.11 Power Saving Mechanism: An Experimental Study. In *Proc. 2008 IEEE Wireless Communications and Networking Conf.* IEEE, 2008.

[9] IEEE Standards Association and others. 802.11-2012-IEEE Standard for Information technology–Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks–Spec. Req. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 2012.

[10] R. Krashinsky and H. Balakrishnan. Minimizing Energy for Wireless Web Access with Bounded Slowdown. *Wireless Networks*, 11(1-2), 2005.

[11] H.-H. Lin, M.-J. Shih, H.-Y. Wei, and R. Vannithamby. DeepSleep: IEEE 802.11 Enhancement for Energy-Harvesting Machine-to-Machine Communications. *Wireless Networks*, 21(2), 2015.

[12] C. Renner, S. Unterschütz, V. Turau, and K. Römer. Perpetual Data Collection with Energy-Harvesting Sensor Networks. *Transactions on Sensor Networks (TOSN)*, 11(1):12:1–12:45, Nov. 2014.

[13] T. Zachariah, N. Klugman, B. Campbell, J. Adkins, N. Jackson, and P. Dutta. The Internet of Things has a Gateway Problem. In *Proc. 16th Intl. Wksp. on Mobile Computing Systems and Applications (HotMobile)*, Santa Fe, NM, USA, Feb. 2015.