Lars Hanschke, Jan Heitmann, and Christian Renner. On the Feasibility of WiFi-Enabled and Solar-Powered Sensors for Smart Ports. In Proc. of the 15. GI/ITG KuVS Fachgespräch Sensornetze, FGSN '16, Augsburg, Germany, Sept. 2016.

On the Feasibility 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
jan.heitmann@student.uni-luebeck.de

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.**

I. Motivation

Current forecasts of the Hamburg Port Authority (HPA) [1] 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. carbonhydroxide 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, environmental friendly and flexible solution are WSNs.

However, most WSN protocols rely on the physical layer of the IEEE 802.15.4 standard, which is designed to cover only small distances and requires dedicated gateways [2] 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 with already deployed infrastructure without additional gateways, which reduces costs and enables simple access to the equipment of the maintenance staff. 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. A regenerative and environmentally friendly option are solar panels in combination with supercapacitors. We have shown their benefits in outdoor scenarios in [3]; thus, performance and achievable power resources are well understood. However, it is unclear if a WiFi node can be supplied by an existing device or if modifications have to be applied.

This paper examines the feasibility of equipping WiFi nodes with an existing energy harvester based on a solar 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 existent IEEE 802.15.4 hardware. Additionally, we present first results of power measurements for a WiFi node and discuss options for improving power consumption.

II. 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 an established IEEE 802.11n standard is future-proof while solving the gateway problem.

TABLE I
DATASHEET POWER CONSUMPTION COMPARISON

State	ATWINC1500A (IEEE 802.11n)	AT86RF230 (IEEE 802.15.4)
TRANSMIT	880 mW@17.8 dBm	51 mW@3 dBm
RECEIVE	297 mW	48 mW
DOZE / TRX_OFF	1 mW	4.5 mW
SLEEP	2.2 μW	0.06 μW

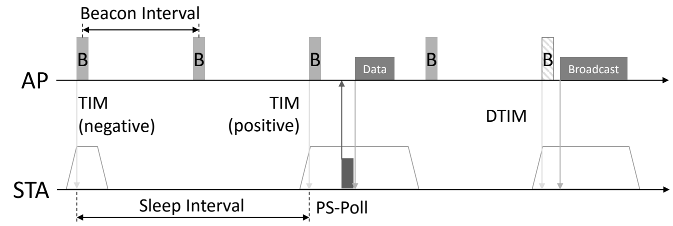

Fig. 1. Sleep Intervals in PSM

A. 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 I. A typical sensor node radio chip, such as the Atmel 68RF230 radio chip on the IRIS mote, consumes only $50\,\mathrm{mW}$ during both transmission and reception of data packets. In contrast, a low-power WiFi transceiver like the Atmel WINC1500A working on the popular Arduino/Genuino MKR1000 runs at $880\,\mathrm{mW}$ transmit power and almost $300\,\mathrm{mW}$ 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 II-C). 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 II-B 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 $54\,\mathrm{Mbit/s}$ PHY layer data rate, yielding a raw energy efficiency of $61\,\mathrm{Mbit/J}$. This is, in theory, more than ten times higher than the transceiver of the IRIS mote providing $4.9\,\mathrm{Mbit/J}$ (assuming a raw PHY rate of $250\,\mathrm{kbit/s}$). 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 time and 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 ($17.8\,\mathrm{dBm}$ compared to $3\,\mathrm{dBm}$), which leads to higher transmission ranges. This makes multi-hop network structures potentially unnecessary, consequently decreasing radio usage, since no message forwarding is needed.

B. 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)) to identify mechanisms to save baseline energy. We will examine the potential of the uplink in a follow-up work. 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 great energy cost.

The standard description of IEEE 802.11 [5] 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 this 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 $65\,\mathrm{s}$ as defined in [5]. Before this maximum idle period runs out, a specific frame should be sent to avoid reconnection to the network. Thus, the maximum sleep time is upper bounded by the maximum idle period.

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. Furthermore the wakeup intervals in X-MAC can not be adjusted according to the needs of the nodes.

C. 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. We presented a miniature solar harvester with a supply voltage of $2.7\,\mathrm{V}$ and a $25\,\mathrm{F}$–$200\,\mathrm{F}$ supercapacitor as energy buffer in [3]. 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 $140\,\mathrm{mW}$ in perfect conditions with an average of only about $5\,\mathrm{mW}$ in a partly shaded position throughout the year. While there is some room for improvement, power-saving techniques are still 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 [3] 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.

III. 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 these periods when experiencing high loads [6]. 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, e.g. HTTP or VoIP, 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 and simply wakes up for the beacons each beacon interval T_{BCN}, calculates as follows when assuming constant power over time:

$$P = P_{\mathrm{doze}} + \frac{L_{\mathrm{BCN}}}{R_{\mathrm{BCN}} \times T_{\mathrm{BCN}}} \left(P_{\mathrm{Rx}} - P_{\mathrm{doze}}\right) . \qquad (1)$$

Here, P_{doze} and P_{Rx} denote the power consumption in sleep and receive mode, respectively, L_{BCN} is the length of a beacon (including MAC and PHY header, and beacon frame body), and R_{BCN} is the data rate.

Assuming typical values ($L_{\mathrm{BCN}} = 250\,\mathrm{B}$, $R_{\mathrm{BCN}} = 1\,\mathrm{Mbit/s}$) combined with the power consumption values of Table I 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. Nevertheless, increasing the beacon interval has to be balanced against the delay criteria of the network.

IV. Preliminary Evaluation

In the following, we report on our first experiences with a WiFi-enabled sensor node, namely the Arduino/Genuino MKR1000 [7], evaluate its power consumption, and discuss the required dimensioning of the solar power supply.

A. 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 $3.3\,\mathrm{V}$, so that the input voltage of the MKR1000 stays above $3\,\mathrm{V}$ while receiving and above $2.5\,\mathrm{V}$ when transmitting. We verified a proper function of the device in these conditions. We'd 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 $102.4\,\mathrm{ms}$ and $1024\,\mathrm{ms}$.

B. Power Consumption

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

Section III shows that the time spent in the receive state to obtain beacons is a crucial factor for the overall power consumption. Since beacons are also transmitted using CSMA/CA, 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 the University of Lübeck. We created high traffic by downloading large files on the university WiFi network on the same channel as our setup to simulate interference from colocated networks. The figure shows that awake times are similarly distributed with a distinct peak in the bin from $10\,\mathrm{ms}$ to $15\,\mathrm{ms}$. 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 $100\,\mathrm{ms}$—i.e., a missed beacon—increased from less than 1% to almost 2%.

When assessing power consumption, we faced several issues discussed in Section IV-D. 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 $102.4\,\mathrm{ms}$, 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 $5\,\mathrm{s}$ 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.

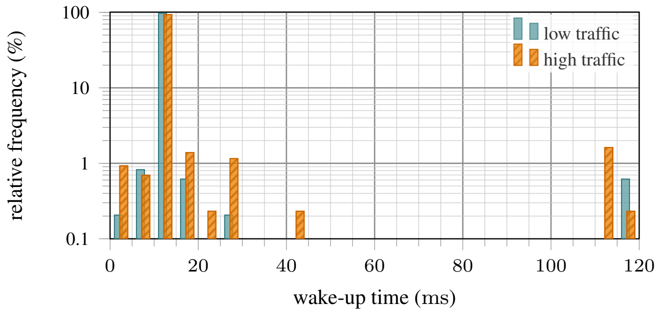

Fig. 2. Distribution of wake-up intervals for beacon reception vs. background traffic of ten 5 s time windows with a beacon interval of 102.4 ms.

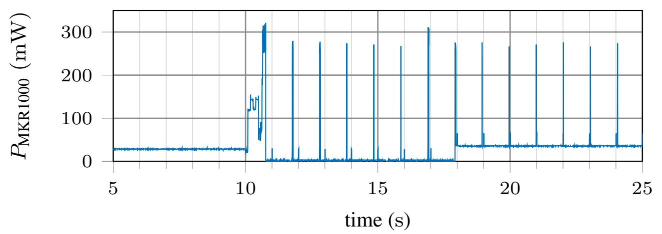

Fig. 3. Wrong implemented standby mode of Arduino in standard library.

C. Solar Power Supply

As a first step, we verified proper function of the MKR1000 with only 2.7 V, the output voltage of our harvester. As discussed in Section IV-B, an overall power consumption (in connected state) compatible with our harvester is achievable; e.g., reducing the beacon wake-up interval to 1024 ms reduces consumption to roughly 4 mW, leaving sufficient room for sending data packets every few seconds or minutes; e.g., sending a 100 ms packet every minute would increase consumption by only 1.2 mW. We are currently preparing long-term tests for running the MKR1000 with our harvester.

D. Hardware and Protocol Issues

The MKR1000 revealed unexpected behavior during our experiments. The first is depicted in Fig. 3. 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 connect mode instead. This behavior appeared independently from the used power-save implementation.

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 4 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. 5. While beacon reception

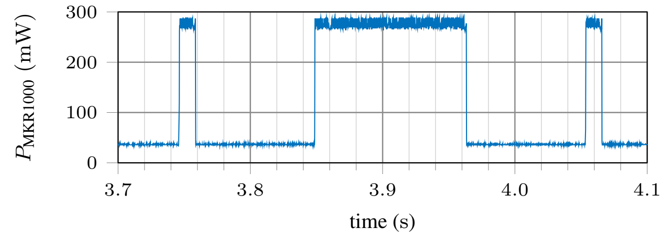

Fig. 4. Missing one beacon leads to a whole period spent in receive mode.

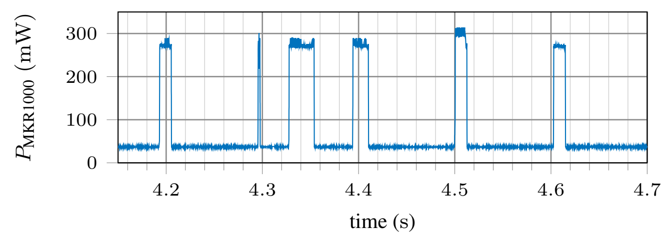

Fig. 5. Strongly varying reception times in power save mode.

occurs every 102.4 ms, another receive phase appears 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.

V. 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 large-scale, remote, maintenance-free deployment, e.g. in smart ports, can be realized with miniature energy harvesters, yet requiring power-saving techniques of the MAC. We made a first effort to show its feasibility. Our measurements reveal that a consumption of 5 mW is generally achievable; but we also identified hard- and software issues, currently hampering low-power consumption. Next, we plan to derive consumption models for adaptive load adaptation, and we will perform outdoor experiments with our solar harvester and a WiFi sensor node.

References

[1] Free and Hanseatic City of Hamburg — State Ministry of Economic Affairs, Transport and Innovation, Hamburg Port Authority, "The Port Development Plan to 2025," Oct. 2012.

[2] T. Zachariah, N. Klugman, B. Campbell, J. Adkins, N. Jackson, and P. Dutta, "The Internet of Things has a Gateway Problem," in *Proc. HotMobile*, Santa Fe, NM, USA, Feb. 2015.

[3] C. Renner, S. Unterschütz, V. Turau, and K. Römer, "Perpetual Data Collection with Energy-Harvesting Sensor Networks," *Transactions on Sensor Networks (TOSN)*, vol. 11, no. 1, Nov. 2014.

[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. SenSys '06*. New York, NY, USA: ACM, 2006.

[5] 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.

[6] R. Krashinsky and H. Balakrishnan, "Minimizing Energy for Wireless Web Access with Bounded Slowdown," *Wireless Networks*, vol. 11, no. 1-2, 2005.

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