HAW HAMBURG

Fakultät Technik und Informatik

Karl-Ragmar Riemschneider , Thomas C. Schmidt (Hrsg.)

16. GI/ITG KuVS Fachgespräch Sensornetze

Proceedings

ISBN: 978-3-96305-001-5

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN HAMBURG
(HAW HAMBURG)
BERLINER TOR 5
20099 HAMBURG

HAW-HAMBURG.DE

HAW HAMBURG

16. GI/ITG KuVS Fachgespräch Sensornetze

der GI/ITG Fachgruppe
Kommunikation und Verteilte Systeme

07.-08. September 2017 in Hamburg

Organisation

General Chair
Karl-Ragmar Riemschneider
Thomas Schmidt

Technical Program Commitee
Falko Dressler
Christian Renner
Bettina Schnor
Lars Wolf

Organizational (Publication, Local Arrangements, Social Event)
Peter Kietzmann
Valentin Roscher
Thorben Schüthe

Preface

Welcome to Hamburg and welcome to FGSN'17, the 16th edition from the series of GI/ITG KuVS events on Wireless Sensor Networks and the Internet of Things. It is a great pleasure to host this community event and provide an open atmosphere for new ideas and discussions, demonstrations and exchange–finally a professional environment for maintaining the kicker tradition introduced by Georg Wittenburg 2008 in Berlin.

As traditional in the field, this year's program reflects the multifaceted works about constrained nodes and their wireless communication. Various experiments and implementations tackle these challenges in a lively manner. With the advent of complex IoT solutions, though, aspects of system and data management came closer into focus. Here, we are particularly happy about a lightening talk by Gabe Rodriguez (Google) on Google's IoT strategy. On the overall, we have to emphasize our gratitude to the program team Falko Dressler, Christian Renner, Bettina Schnor, and Lars Wolf, who managed to assemble a convincing program.

FGSN'17 would not have been possible without our efficient organising team: We are very grateful to Peter Kietzmann, Valentin Roscher, and Thorben Schüthe for their fantastic job on everything that was needed. We also thank the 'Förderverein Elektrotechnik und Informatik der HAW Hamburg e.V.' for the generous support of this event.

We hope you all will enjoy two days of exchange and inspiration in the beautiful city of Hamburg!

Karl-Ragmar Riemschneider &
Thomas C. Schmidt &
FGSN'17 Chairs

Contents

miniDTN: A DTN Stack for 5€-WiFi-Nodes

Stephan Rottmann, Alexander Willecke, Jan Käberich, Georg von Zengen, and Lars Wolf
Institut für Betriebssysteme und Rechnerverbund
TU Braunschweig, Germany
Email: [rottmann | willecke | kaeberic | vonzengen | wolf]@ibr.cs.tu-bs.de

***Abstract*—Deployments of wireless networks consisting of small nodes offer many possibilities in the fields of Internet of Things (IoT), smart farming applications or Body Area Networks. To cope with implied challenges like low energy resources, failing links, limited computation power and many more, a robust communication stack is beneficial. An example for such a stack is the Delay/Disruption Tolerant Networking (DTN) architecture.**

In many cases, the sensor nodes consist of extreme low power nodes, 8 bit controllers. However, for several applications, more resources would be beneficial, at least for some components of the networks. In this paper, we focus on more powerful systems, such as 32 bit controllers bridging the gap between traditional Wireless Sensor Networks and PCs. We present *miniDTN*, a DTN implementation based on FreeRTOS for many kinds of nodes, enabling communication via different media, such as Ethernet, WiFi or LoRa.

I. Introduction

Wireless sensor and actuator networks can be useful for a vast amount of applications. They can be used in structural health monitoring of buildings, for home automation purposes or in the field of smart farming as well as the Internet of Things (IoT). Depending on the actual use case of a network installation, the amount of nodes deployed may also vary. The more nodes are deployed, the higher the probability of failures will be. Nodes may disappear from the network due to power failures resulting from empty batteries, software errors or radio interference.

Being able to cope with these problems, a robust communication protocol is necessary to ensure high reliability for data transmissions – even if some nodes fail.

A robust communication approach for networks, which may have unreliable links due to mobility or shortage of energy, is the Delay/Disruption Tolerant Networking (DTN) architecture [1], [2]. Moreover, DTN is actually able to benefit from node mobility by storing packets – in DTN called bundle – on the node and physically carrying them to another location before forwarding them. Several implementations targeting a broad range of platforms exist (ION-DTN [3], IBR-DTN [4], µDTN [5], JDTN, ByteWalla, µPCN [6], DTN2, ...). Only few of them can be used on very "small" systems like an 8 bit microcontroller, while others are designed for full-featured operating systems like Linux running on PCs.

During the last years, more and more relatively cheap, but powerful hardware platforms emerged on the market. Examples are 32 bit ARM controllers and their development boards or the *ESP32*, a dual-radio WiFi node running at up to 240 MHz with two cores. Due to the low price of these boards, DTN becomes suitable to applications where traditional Wireless Sensor Network (WSN)-nodes were not powerful enough and PCs were not affordable, like gathering vast amounts of data in smart farming applications. To be able to exploit the capabilities of these boards, we developed *miniDTN*, based on *µDTN* [5].

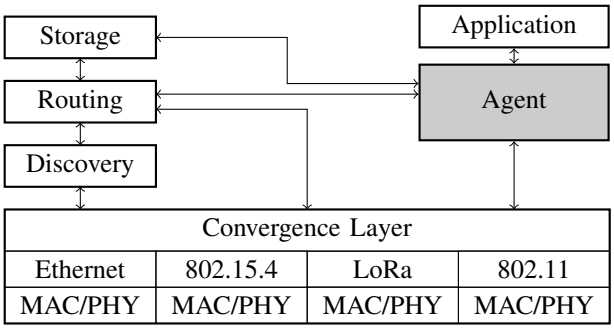


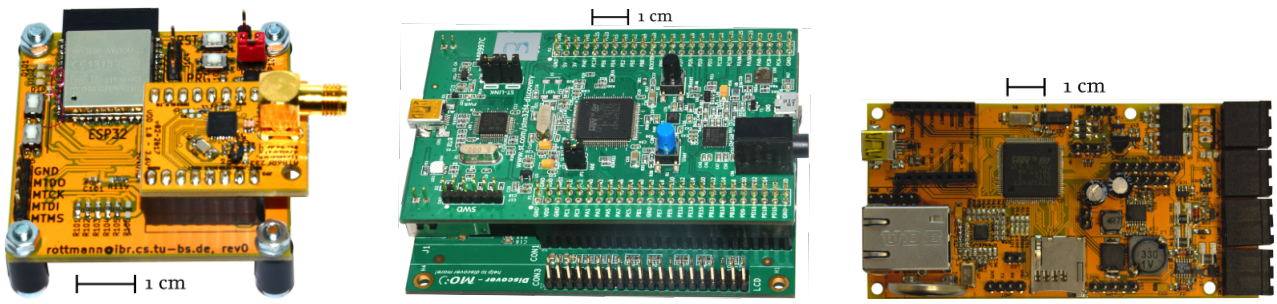
Fig. 1: Architecture of *miniDTN* stack

An example smart farming application and use case has been presented in [7], [8]. The task of the sensor network is to collect data about temperate and soil humidity on a field and forward this data when a tractor is working on that field is passing by to a mobile node. After the tractor returns to the farm, it forwards this data to a sink node, where it might be processed or stored in any kind of database.

In Section II, we describe in more detail how *miniDTN* works and what its architecture looks like. The platforms we target with *miniDTN* and what their features are is pointed out in Section III. To show which applications can benefit from *miniDTN*, we perform an evaluation and discuss the results in Section IV. Section V concludes the paper.

II. Architecture

In *miniDTN* the agent – shown in Fig. 1 – is the central instance which coordinates all the other components shown. As we implemented *miniDTN* for *FreeRTOS*, the agent is the part to be launched from the *FreeRTOS* environment. Modules are shown as blocks in Fig. 1. Each of them is responsible for one major task within *miniDTN*. The modules are: Storage, Routing, Discovery and Convergence Layer. The coordination between the different modules is event driven, thus expanding or exchanging modules is eased. That comes in handy if an application has special routing needs or a platform offers a specialized kind of storage. For most modules only one implementation is allowed at compile time. The only



(a) ESP32 with IEEE 802.15.4 radio (b) STM32F4Discovery (c) Amphisbaena

Fig. 2: Different platforms currently supported by *miniDTN*

exception is the Convergence Layer (CL). Depending on the platform, multiple CLs can be loaded, using different kind of communication interfaces.

- **Application**: To use the DTN stack, the user needs to write an application which registers an endpoint at the agent. Bundles received for this endpoint are made accessible for the application.
- **Agent**: The agent is the central process started on the actual platform from the user code. It manages the interaction of the other components.
- **Storage**: The storage module's task is to make received or generated bundles accessible for the application or routing. Bundles may be stored in the microcontroller's RAM or on a SD card in the FAT file system, for example. The latter one is slower but bundles are stored nonvolatile and thus are still available after a reset.
- **Routing**: The routing module decides which bundles should be sent to which neighbors. A simple *flooding* forwards bundles to all nodes except the originating one.
- **Discovery**: The discovery part of the implementation keeps track of the neighbors and their convergence layers. Beacons are sent out announcing nodes on a broad- or multicast channel of the actual medium. By now, we implemented a DTN IP Neighbor Discovery (IPND) [9] conform mechanism.
- **Convergence Layer**: In *miniDTN*, multiple CLs can be implemented. At the moment `dgram:udp` for communication in IP networks as well as `dgram:lowpan` for transfers via IEEE 802.15.4 or LoRa are available. The two latter ones are using the same frame format, only the radio driver needs to be modified. The TCPCL [10] is in development.

III. Target Platforms

We have different target platforms shown in Fig. 2, which *miniDTN* is ported to. Fig. 2a is an ESP32 on a development board with a pluggable Atmel AT86RF233 radio. Thus, it is able to communicate via IEEE 802.11b/g/n as well as IEEE 802.15.4 or LoRa. Fig. 2c is our *Amphisbaena* platform [11], which is also able to carry different types of wireless transceivers. That way these platform might work as a bridge between networks using different communication mediums. Another application is the use of different wireless standards for according tasks, e.g. using a low power standard for heartbeats and smaller bundles and IEEE 802.11 for large bundles. Fig. 2b shows the well-known STM32F4Discovery board with an Ethernet shield. *Amphisbaena* uses the same central hardware components and thus, the performance of both is the same.

miniDTN can be ported to any other hardware where *FreeRTOS* is available. It is open source and available in a GIT[1] repository. For Ethernet and WiFi communication, the LwIP stack in version 2.0.0 is used.

For the mentioned platforms, examples are present online[2]. It also runs on commonly available ESP32 development boards. To use the implementation in a new project, the *miniDTN* code needs to be added to the build-system. Low level definitions for the hardware interface have to be configured in a single header file.

IV. Performance Evaluation

In smart framing applications, networks in the farm itself consist of stationary nodes like servers or workstations. These nodes are typically connected via IEEE 802.3. Additional network nodes might be used to monitor either the storage of the crop, for example. As a farm often is spread over a large area, wireless and mobile nodes might be used to connect the wired networks with each other by physically carrying bundles using mobile nodes such as tractors. The use of wireless nodes is not limited to carry bundles, they can also be used to monitor cattle and plants on the fields. In our evaluation we will show that *miniDTN* is suitable for all these applications.

To do so, we evaluated a minimal setup shown in Fig. 3. A router running OpenWRT is used to connect all nodes:

- **Raspberry Pi**: A PC style node based on a Raspberry Pi 3 running IBR-DTN on Linux. It uses the integrated WiFi device to connect to the AP.
- **Amphisbaena**: The low-power-part of our dual-platform node *Amphisbaena*, connected via Ethernet. It is based on the same components as the STM32F4Discovery.

[1] https://gitlab.ibr.cs.tu-bs.de/minidtn/minidtnStack
[2] https://gitlab.ibr.cs.tu-bs.de/minidtn

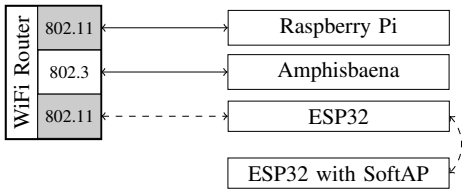


Fig. 3: Evaluation Setup for UDP: Continuous lines depict Ethernet, the dashed ones IEEE 802.11n. All node can communicate via IEEE 802.15.4 with each other as well.

- **ESP32**: An ESP-WROOM-32 based platform.
- **ESP32 with SoftAP**: The same as before, but not connected to the AP, since it opens its own AP.

Thus, the *Amphisbaena* can be seen as a stationary device and the ESP32 acts as a wireless node. Both are using the same code base of *miniDTN*. All our evaluations are done within this setup, all nodes use IPND to discover their neighbors, flooding routing to find the route – which in our case is always the next hop – and the bundles are transferred using the `dgram:udp` CL. As storage we use the RAM module to get the best performance in our measurements.

A. Results

To show how long a contact between two nodes needs to last until data is transmitted, we measured the time between power-on and the first bundle transmission. This includes the boot process, the network configuration and the time the discovery takes to discover neighboring nodes, we call this time start-up time. Fig. 4 shows the start-up time for the *Amphisbaena* and the ESP32. We did not test the start-up time for the Raspberry Pi, as such PC like devices take a lot longer to boot and as servers they are running most of the time, anyways.

For the *Amphisbaena* we measured a maximum start-up time of 3.2 s, the ESP32 needs with 8.9 s a lot longer. This is due to the WiFi association process that the *Amphisbaena* does not need. These result show that a contact time of 10 seconds is sufficient to transfer data.

The 3.2 s of the *Amphisbaena* give an idea how much a stationary device can be duty cycled to save energy and put the node into a deep sleep mode.

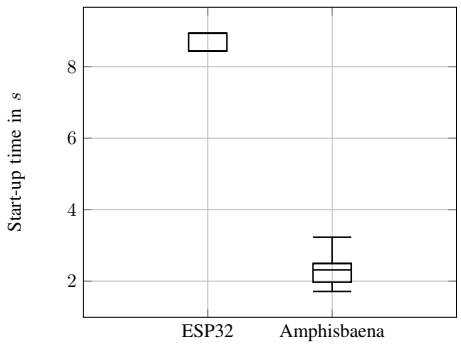


Fig. 4: Comparison of start-up time between the ESP32 and Amphisbaena.

Next, the goodput is evaluated between the platforms using different communication links. The goodput is defined as the bundle payload size, i.e., the data actually generated by the application. The evaluation took place in an university lab environment. This means, that many other WiFi stations exist causing interference. In a real world scenario, this would be the case, too, depending on the density of nodes in the deployment.

The results are shown in Fig. 5 (using UDP in IP networks) and Fig. 6 (for IEEE 802.15.4). In both figures the three graphs show the goodput to be achieved when sending data *to* a platform. This representation is used to point out the differences in the architectures. The platforms *ESP32* and *Amphisbaena* are running the code base on microcontrollers. For the former one, some overhead exists from the SDK which cannot be influenced by the programmer; it manages the WiFi interface, for example.

Due to the rather small amount of RAM available on the microcontroller based platforms, a bundle size larger than 8 kByte will become unfeasible, since the whole bundle has to be kept in the memory for processing.

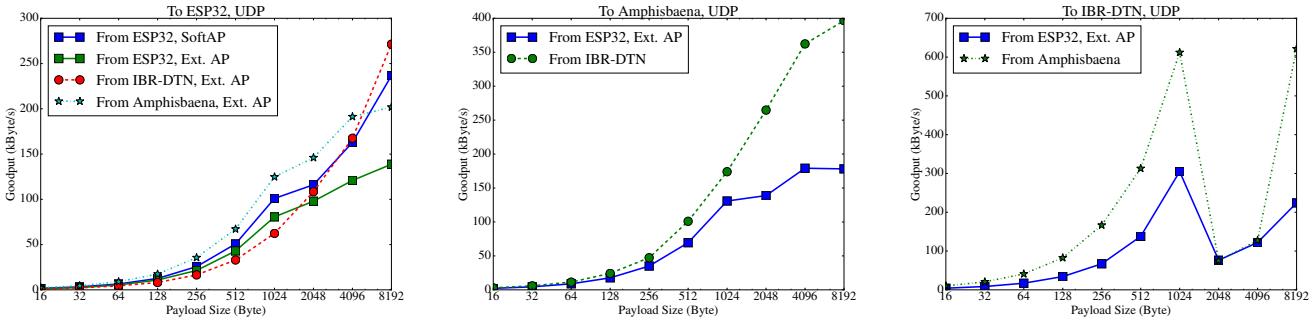
In Fig. 5a to Fig. 5c, we can see that when the `dgram:udp` CL is used, the goodput which can be achieved is strongly dependent on the bundle's payload size and the combination of nodes. Fig. 5c reveals that the performance decreases as soon as a bundle does not fit into the MTU of an Ethernet or WiFi-frame when sending data to IBR-DTN. When sending bundles with a payload size of 1024 Byte, 300 kByte/s to 600 kByte/s can be achieved, but at a payload size of 2048 Byte, it drops to less than 100 kByte/s. The difference between the source nodes (*Amphisbaena* vs. ESP32) results from the fact that the *Amphisbaena* is connected via Ethernet to the AP, and thus only *one* WiFi link is used. The same effect can be observed when looking at Fig. 5a. If one of the ESP32 creates a software AP, the data has to be sent only once over the air, and not to the external AP which transmits the data to the second ESP32.

Fig. 6a to Fig. 6c show the same evaluation setup as before, but for the use of IEEE 802.15.4. As expected, the goodput is much lower since the data rate of the physical layer is only 250 kBit/s. When sending data to *miniDTN*, only around 12 kByte/s can be achieved, while IBR-DTN is able to accept data with around twice the speed – as long as bundles are not larger than 2 kByte. For larger bundles, the performance decreases to a similar level as when sending data to *miniDTN*.

Using IEEE 802.15.4 consumes less energy than WiFi. If only small amounts of data (like heartbeats or temperature values) should be transmitted by nodes deployed in the field, the low power link should be chosen. Data can be aggregated and transmitted in once in large bundles via WiFi, which can be turned on for example, when a mobile node arrives. The presence of such a node can be detected via the low power link.

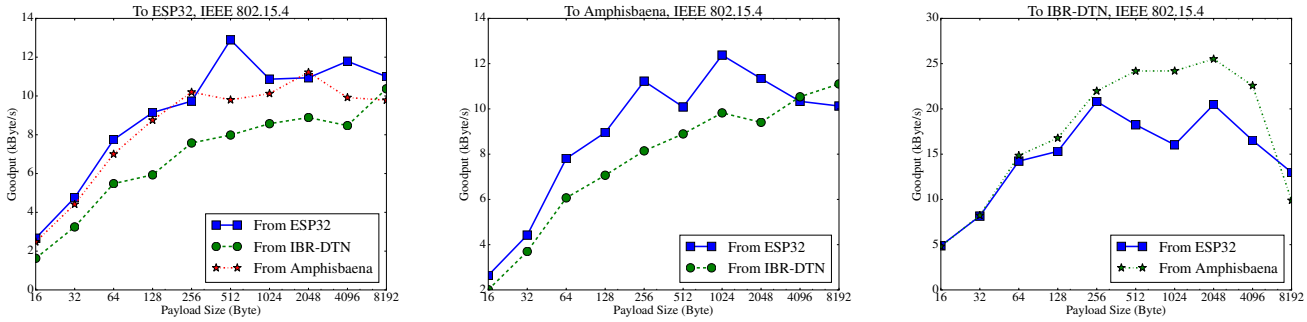
V. Conclusion

In this paper, we presented *miniDTN*, a DTN implementation based on *FreeRTOS*, running on many different kinds



(a) (b) (c)

Fig. 5: Measurements with UDP over WiFi/Ethernet



(a) (b) (c)

Fig. 6: Measurements with IEEE 802.15.4

of platforms. Common platforms are already supported, while it can be ported to any custom board, supporting *FreeRTOS*. The evaluation has shown that it is able to transfer sufficient amounts of data in rather short contacts. Especially if the speed of tractors while field work and the amount of data gathered is taken into account.

When using `dgram:udp`, the bundles containing the data to be transferred should be chosen as large as possible to achieve the best goodput, however when sending data to IBR-DTN, the performance decreases significantly as soon as the MTU of Ethernet or WiFi is exceeded. All in all, one can expect a goodput of at least 150 kByte/s to 200 kByte/s, which is sufficient for the applications targeted by the presented platforms.

References

[1] K. Scott and S. Burleigh, "Bundle Protocol Specification," RFC 5050 (Experimental), Internet Engineering Task Force, Nov. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc5050.txt

[2] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "Delay-Tolerant Networking Architecture," RFC 4838 (Informational), Internet Engineering Task Force, Apr. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4838.txt

[3] S. Burleigh, "Interplanetary Overlay Network: An Implementation of the DTN Bundle Protocol," in *2007 4th IEEE Consumer Communications and Networking Conference*, Jan 2007, pp. 222–226.

[4] S. Schildt, J. Morgenroth, W.-B. Pöttner, and L. Wolf, "IBR-DTN: A lightweight, modular and highly portable Bundle Protocol implementation," *Electronic Communications of the EASST*, vol. 37, pp. 1–11, Jan 2011.

[5] G. von Zengen, F. Büsching, W.-B. Pöttner, and L. Wolf, "An Overview of μDTN: Unifying DTNs and WSNs," in *Proceedings of the 11th GI/ITG KuVS Fachgespräch Drahtlose Sensornetze (FGSN)*, Darmstadt, Germany, 9 2012.

[6] M. Feldmann and F. Walter, "upcn: A bundle protocol implementation for microcontrollers," in *2015 International Conference on Wireless Communications Signal Processing (WCSP)*, Oct 2015, pp. 1–5.

[7] B. Gernert, S. Rottmann, and L. C. Wolf, "Poster: PotatoMesh - a solar powered WSN testbed," in *The 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing: MobiHoc 2016 - Posters (ACM MobiHoc 2016 - Posters)*, Paderborn, Germany, Jul. 2016, pp. 391–392.

[8] U. Kulau, S. Schildt, S. Rottmann, B. Gernert, and L. Wolf, "Demo: Potatonet – robust outdoor testbed for wsns: Experiment like on your desk. outside." in *Proceedings of the 10th ACM MobiCom Workshop on Challenged Networks*, ser. CHANTS '15. New York, NY, USA: ACM, 2015, pp. 59–60.

[9] D. Ellard, R. Altman, A. Gladd, and D. Brown, "DTN IP Neighbor Discovery (IPND)," Nov. 2015. [Online]. Available: http://tools.ietf.org/pdf/draft-irtf-dtnrg-ipnd-03.pdf

[10] M. Demmer, J. Ott, and S. Perreault, "Delay-Tolerant Networking TCP Convergence-Layer Protocol," RFC 7242, Jun. 2014. [Online]. Available: https://rfc-editor.org/rfc/rfc7242.txt

[11] S. Rottmann, R. Hartung, J. Käberich, and L. C. Wolf, "Amphisbaena: A Two-Platform DTN node," in *The 13th International Conference on Mobile Ad-hoc and Sensor Systems (MASS 2016) (MASS 2016)*, Brasilia, Brazil, Oct. 2016.

End-to-End Performance Analysis for Industrial IEEE 802.15.4e-based Networks

Neda Petreska
Fraunhofer Institute for Embedded Systems and
Communication Technologies ESK
neda.petreska@esk.fraunhofer.de

***Abstract*—In this extended abstract we present an analytical solution for the end-to-end delay bound for multi-hop IEEE 802.15.4e-based networks. We base our derivation on the stochastic network calculus principles in order to provide performance guarantees for wireless industrial sensor networks. We validate the derived solution by simulations. Further, we show that the bound is convex, which, together with its monotonicity in the average SNR on the links, enables design of algorithms for optimal transmit power allocation. These two aspects: optimizing network operation while at the same time providing end-to-end delay guarantees, are of great importance for the development of reliable, stable and power-efficient wireless industrial networks.**

I. Motivation

The fast installation and easy maintenance of wireless sensor networks (WSN) makes them suitable for many industrial applications, such as process automation, monitoring and control, predictive maintenance, localization and navigation, smart grid, logistics and the Internet-of-Things. One of the main purposes of these networks, having in mind low-power battery-driven sensor devices, was to enable an energy-efficient network operation, thereby maximizing their battery lifetime [1]–[3]. In order to bridge larger distances, especially useful for the application areas of process automation, logistics, localization and navigation of autonomous vehicles, but at the same time avoid using significantly higher transmit power which would potentially lead to faster battery depletion, multi-hop communication should be used. However, with the increased industrial digitalization within "Industry 4.0", it becomes necessary that WSN enable not only an energy-efficient operation, but also provide a stable and reliable communication platform for an uninterrupted operation of factory sites and automation processes. Hence, a set of QoS requirements, mainly given by a target delay and packet error rate (PER), have to be met.

In order to design WSN properly so that the requested performance of the network is guaranteed, a solid performance analysis framework is needed. The challenge of such analysis lies obviously in the stochastic nature of the wireless fading channel. Due to the random wireless channel capacity, queuing effects have to be considered in addition. Such appropriate theoretical framework which enables performance analysis of wireless multi-hop sensor networks is the theory of stochastic network calculus [4]. A recent alternative approach for modeling the impact of channel gain models on the network-layer performance of wireless networks was presented in [5]. The so called $(\min, \times)$ algebra provides analytical tools to define the bound on the delay violation probability over a cascade of wireless fading channels.

As we are interested in wireless industrial sensor networks based on the IEEE 802.15.4e standard for low power wireless networks, while at the same time considering queuing effects, we present in this work a closed-form solution for the end-to-end delay bound as a performance metric. After a short discussion of related work in Sec. II and a brief presentation on the system model in Sec. III, we give an overview of the used stochastic network calculus and in Sec. IV present the end-to-end delay bound for IEEE 802.15.4e-based networks. In Sec. V we discuss the numerical results and conclude the paper with upcoming future work.

II. Related Work

Many research papers address performance evaluation of 802.15.4-based networks, often opting for simulation and/or real test-bed evaluation and measurements [6]–[8]. However, there are not many works presenting analytical approaches for network performance of industrial wireless systems. [9] and [10] address performance aspects of IEEE 802.15.4e networks. While [9] provides only simulation experiments to show the typical operation of the protocol, [10] analytically derives the worst-case latency for typical industrial settings. However, the latter paper neither considers multi-hop communication nor it involves the queuing delay, i.e., the random capacity of the wireless channel. [11] analyzes the performance of IEEE 802.15.4e DSME-enabled (Deterministic and Synchronous Multi-Channel Extension) and IEEE 802.15.4 beacon-enabled WSN under the influence of heterogeneous devices. However, no analytical framework is presented: the analysis is done rather via simulations. The authors of [12] focus on the network formation process in IEEE 802.15.4e TSCH networks and analyze the time needed by a node to join the network. They define a random-based advertisement algorithm and evaluate its sensitiveness to different parameters, such as the number of used channel offsets, node density, packet error rate and number of available frequencies. None of the mentioned works presents end-to-end performance analysis for IEEE 802.15.4e networks under queuing aspects due to the random nature of wireless fading channels.

III. Preliminaries

In this section we first give a brief introduction to IEEE 802.15.4e and then describe the considered system model.

A. Introduction to the IEEE 802.15.4e Standard

The IEEE 802.15.4e standard was designed as an extension of the CSMA/CA-based IEEE 802.15.4 standard [13] in order to address the emerging needs of wireless industrial applications. This standard describes a MAC protocol for low-power multi-hop wireless networks, intended to increase the reliability and decrease the delay of the network. A special feature of the standard is the *TSCH mode* (Time Synchronized Channel Hopping), which is especially targeted to process automation applications. This is why there is a big similarity between the TSMP (Time Synchronized Mesh Protocol) [14], a part of the WirelessHART standard for process automation and the TSCH. The main reliability-boost feature lies in the so called *slotframes*, which represent a time slot to channel matrix. A *(time slot, channel)* pair is allocated to each transceiver, enabling reduced interference between subsequent transmissions. Having a time slot length of 10 ms, TSCH provides an increased network capacity, high reliability and predictable latency. Due to this mode, IEEE 802.15.4e is especially suited for multi-hop communication, since neighbouring nodes can be allocated different resources within the same slotframe, without interfering with each other. While the slotframes can be of different length, the total number of available channels for frequency hopping is 16.

B. System Model

We consider a data flow originating from node a, being forwarded via a cascade of wireless links all the way to a destination node b (see Figure 1). At the beginning of each slotframe, the application running on a produces payload of size r_a bits. Several payload packets can be transmitted within a MAC frame of size $k_a = 127$ B. Due to the CRC, the instantaneous service of each wireless link is a Bernoulli distributed random variable and takes values of either 0 or k_a. In case of an unsuccessful transmission, i.e., no reception of an ACK within a time slot, the packets are being queued at the nodes. Assuming a round-robin scheduling, each node transmits its buffered packets within the next slotframe, having one time slot assigned for transmission per slotframe.

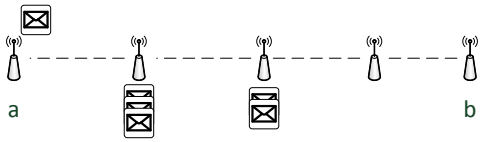


Fig. 1: Ilustration of the multi-hop path considered in the system model. The intermediate nodes can queue the packets, in case of an unsuccessful transmission.

We assume block-fading channels, where the instantaneous SNR of link j at slotframe i, $\gamma_{i,j}$ remains constant during the allocated time slot and changes from one slotframe to the other. Since two consecutive transmissions by the same node on the same link are several time slots apart and potentially taking place on different channels, the instantaneous SNR of the same link are independent and uncorrelated.

In the following we are going to define a bound on the delay violation probability. The QoS requirements of the application are given with the target delay w and the target probability for its violation ε. Hence, we seek to design the WSN in such way, such that the delay bound is always smaller than ε.

IV. End-to-End IEEE 802.15.4e-based Delay Bound

As already mentioned, we aim towards a closed-form expression of the end-to-end delay violation probability (or also called a delay bound) for the multi-hop path presented in the previous section. We first derive the delay bound for link j. For this purpose we use the $(\min, \times)$ stochastic network calculus in the SNR domain [5] and compute the Mellin transform of the arrival and service process. Assuming an industrial automation application with a constant data rate, we define the cumulative arrival process and its Mellin transform as $\mathcal{A}_j(\tau, t) = e^{r_a(t-\tau)}$ and $\mathcal{M}_{\mathcal{A}_j}(s, \tau, t) = e^{r_a(t-\tau)(s-1)}$, $s > 0$, respectively. The instantaneous service of the j-th link in the i-th slotframe is Bernoulli distributed:

$$X_{i,j} = \begin{cases} k_a, & 1 - \mathrm{P}(\gamma_{i,j}) \\ 0, & \mathrm{P}(\gamma_{i,j}) \end{cases} \tag{1}$$

where $\mathrm{P}(\gamma_{i,j})$ is the probability of erroneously received frame at link j in slotframe i. The frame error rate is a function of the BER $p(\gamma_{i,j})$, which, in turn, is a function of the instantaneous SNR. The cumulative service in the SNR domain results in

$$\mathcal{S}_j(\tau, t) = e^{\sum_{i=\tau}^{t-1} X_{i,j}} = \prod_{i=\tau}^{t-1} e^{X_{i,j}}. \tag{2}$$

Hence, its Mellin transform is as follows:

$$\begin{aligned} \mathcal{M}_{\mathcal{S}_j(\tau,t)}(s) &= \left[\mathcal{M}_{e^{X_{i,j}}}(s)\right]^{(t-\tau)} \\ &= \left(1 + (e^{k_a(s-1)} - 1)\mathrm{Pr}(X_{i,j} = k_a)\right), \end{aligned} \tag{3}$$

where $\mathrm{Pr}(X_{i,j} = k_a)$ is the marginal probability distribution given by:

$$\mathrm{Pr}(X_{i,j} = k_a) = \int_0^\infty (1 - p(y))^{k_a} \cdot \frac{1}{\bar{\gamma}_j} e^{-y/\bar{\gamma}_j} dy, \tag{4}$$

for a Rayleigh fading channel and exponentially distributed SNR with parameter $\bar{\gamma}_j$.

According to the $(\min, \times)$ network calculus, the delay bound $\mathcal{K}(s, -w)$ is defined as a function of the Mellin transform of the cumulative arrival and service process:

$$\mathcal{K}(s, -w) = \sum_{i=0}^{\min(\tau,t)} \mathcal{M}_{\mathcal{A}}(1 + s, i, t)\mathcal{M}_{\mathcal{S}}(1 - s, i, \tau). \tag{5}$$

We are interested in the delay bound for a steady state network, so we solve Eq. (5) for $t \to \infty$ and obtain:

$$\mathcal{K}(s,-w) = \frac{\left(1+(e^{-k_a s}-1)\Pr(X_{i,j}=k_a)\right)^w}{1-e^{r_a s}\left(1+(e^{-k_a s}-1)\Pr(X_{i,j}=k_a)\right)}. \quad (6)$$

The sum converges if

$$e^{r_a s}\left(1+(e^{-k_a s}-1)\Pr(X_{i,j}=k_a)\right) < 1, \quad (7)$$

which at the same time presents the system stability condition. Hence, we obtain the delay bound over a link j for a given target delay w. We encourage the interested reader to take a look at the whole derivation of the IEEE 802.15.4e-based delay bound in [15].

The end-to-end delay bound is obtained from Theorem 1 in [16], which essentially reveals its recursive nature. The recursion is especially useful for packet routing and establishing of new routes in case one link is added to the path or a pre-existing link's average SNR has been changed. In such cases, using the recursion to compute the new end-to-end service curve, saves a lot of computation time in comparison to performing the end-to-end $(\min, \times)$ convolution.

V. Numerical Results

In this section we show the validity of the obtained analytical delay bound and discuss its properties. Fig. 2 shows that the end-to-end delay bound is indeed an upper bound on the delay violation probability obtained for a IEEE 802.15.4e-based multi-hop path obtained by simulations. In each iteration i of the simulation an instantaneous frame error rate $\mathrm{P}(\gamma_{i,j})$ is computed, according to the instantaneous SNR $\gamma_{i,j}$ on the links along the path. The instantaneous service per link is drawn from a Bernoulli distribution with this probability. The packets are propagated in a round robin fashion along the path and are buffered at the nodes each time $X_{i,j} = 0$. We show the simulated and analytical delay violation probability for several target delays and observe that the tightness of the bound depends on the number of links of the path. This gap results from the union bound used in the derivation of the delay bound.

Fig. 3 shows the end-to-end delay bound for several 3-hop paths, where each path is obtained by doubling the respective link's SNR. We observe that the delay bound decreases as the SNR on the links increases and as the target delay requirement gets looser.

In Fig. 4 we present a very important characteristic of the end-to-end delay bound obtained by the $(\min, \times)$ network calculus. The bound is convex in s and the minimum of the delay bound function depends on the SNR allocation on the links. As the average SNR on the links is decreased, the minimum is shifted to the left and up. Note here that, every path has a different interval for the s parameter, i.e. $s \in (0, s_{\max})$, where $(0, s_{\max})$ follows from the stability condition given by Eq. (7). This convex behaviour opens the question about an optimal SNR allocation along the path in order to meet certain target QoS. As we can see in the figure, for a target delay violation probability of 10^{-3}, typical for applications in the area of process automation, one can design the network path by enabling SNRs of $\{5, 7, 9\}$ dB or even lower, which could potentially lead to different node placement or transmit power allocation. Having in mind the approximately one order of magnitude gap between the analytical delay violation probability and the simulated one from Fig. 2, even if the delay bound for the path $[5, 7, 9]$ dB lies above $\varepsilon = 10^{-3}$, the actual system delay violation probability will lie below ε. Allowing a lower transmit power is a very important aspect for WSN, since this leads to an extended battery and therefore, network lifetime. Note that a target delay of 6 slotframes results into a delay of 180 ms for the observed 3-hop scenarios.

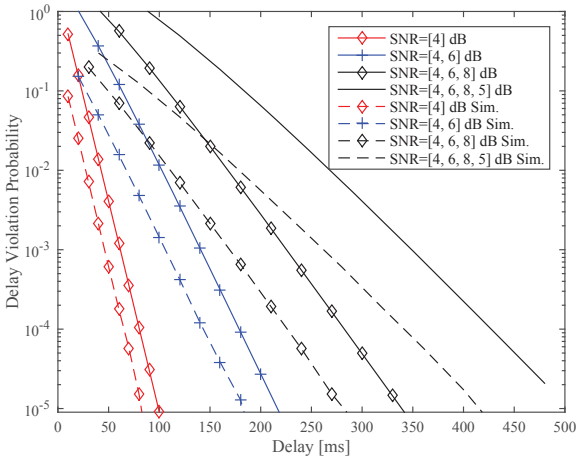


Fig. 2: Validation of the delay bound for a IEEE 802.15.4e-based system for different multi-hop scenarios.

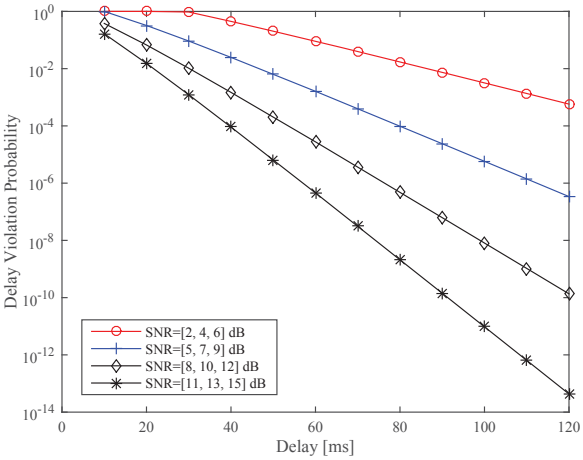


Fig. 3: Analytical delay bound vs. different target delays for several 3-hop scenarios. Doubling the SNR per link leads to two orders of magnitude lower delay bound for payloads of 10 B.

VI. Conclusion and Outlook

In this extended abstract we present a closed-form solution for the end-to-end delay bound over a cascade of

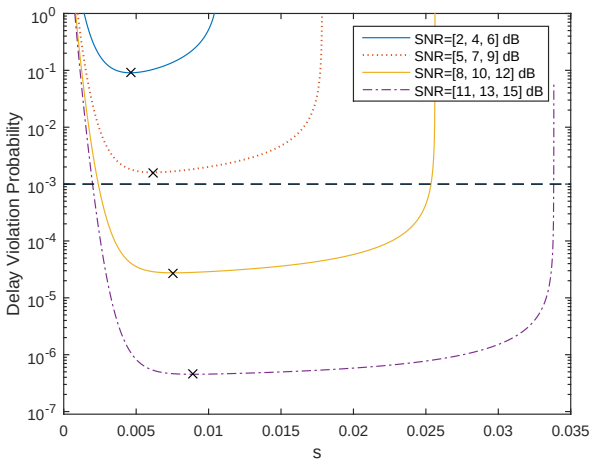


Fig. 4: The end-to-end delay bound is convex in s. The payload size is $r_a = 10$ B, the target delay $w = 6$ slotframes and $\varepsilon = 10^{-3}$.

wireless fading channels. The provided performance analysis allows for the channel gains along the path to be heterogeneously distributed, having different average SNR on the links. Based on the BER-to-instantaneous-SNR relation given in the IEEE 802.15.4e standard, we have first derived a service curve for such system and afterward analytically defined the bound on the delay violation probability. We have also shown that the provided solution is an upper bound on an analogously simulated system behaviour. We further present the convex behaviour of the end-to-end delay bound, opening the way towards determining the optimal average SNR needed on the links along the path in order to guarantee certain application delay requirement. The provided results can be used in the network design of QoS-driven wireless industrial IEEE 802.15.4e-based networks, such as WirelessHART [17] and ISA100.11a [18].

In a previous work [19] we have used the convexity of the bound in s and the monotonicity in $\bar{\gamma}$ and have defined a binary search algorithm, which determines the optimal transmit power on a single link, in order to meet certain statistical delay constraints. However, this was done for a service modeled according to the Shannon capacity. On the other hand, the IEEE 802.15.4e-based service curve and delay bound enable similar approach, only this time, for practical wireless industrial networks. Currently, we are looking into a similar algorithm which would maximize the minimal battery lifetime of the nodes along a multi-hop path for a system model as presented in this paper. Determining the optimal transmit power so that given end-to-end statistical delay constraints are guaranteed in a wireless industrial network, will open new perspectives in the design of industrial WSN. Such results would be of essential meaning for the further development of wireless industrial networks, enabling both reliability and flexibility. Performance analysis together with optimal resource allocation in WSN has the potential to enable faster acceptance of wireless communication for industrial applications and open even broader perspectives to benefit from it even in case of time critical processes.

REFERENCES

[1] H. Long, Y. Liu, Y. Wang, R. P. Dick, and H. Yang, "Battery Allocation for Wireless Sensor Network Lifetime Maximization Under Cost Constraints," in *2009 IEEE/ACM International Conference on Computer-Aided Design - Digest of Technical Papers*, Nov 2009, pp. 705–712.

[2] R. Asorey-Cacheda, A. Garcia-Sanchez, F. Garcia-Sanchez, J. Garcia-Haro, and F. Gonzalez-Castano, "On Maximizing the Lifetime of Wireless Sensor Networks by Optimally Assigning Energy Supplies," *Sensors (Basel)*, vol. 13, no. 8, pp. 10 219–10 244, August 2013.

[3] A. A. Aziz, Y. A. Sekercioglu, P. Fitzpatrick, and M. Ivanovich, "A Survey on Distributed Topology Control Techniques for Extending the Lifetime of Battery Powered Wireless Sensor Networks," *IEEE Communications Surveys Tutorials*, vol. 15, no. 1, pp. 121–144, 2013.

[4] Y. Jiang, "A Basic Stochastic Network Calculus," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 4, pp. 123–134, Aug. 2006.

[5] H. Al-Zubaidy, J. Lieberherr, and A. Burchard, "Network-Layer Performance Analysis of Multi-Hop Fading Channels," *IEEE/ACM Transactions on Networking (ToN)*, vol. 24, no. 1, pp. 204–217, Feb 2016.

[6] P. Ferrari, A. Flammini, S. Rinaldi, and E. Sisinni, "Performance Assessment of a WirelessHART Network in a Real-World Testbed," in *Instrumentation and Measurement Technology Conference (I2MTC), 2012 IEEE International*, May 2012, pp. 953–957.

[7] J. Song, S. Han, A. Mok, D. Chen, M. Lucas, M. Nixon, and W. Pratt, "WirelessHART: Applying Wireless Technology in Real-Time Industrial Process Control," in *Real-Time and Embedded Technology and Applications Symposium, 2008. RTAS '08. IEEE*, April 2008, pp. 377–386.

[8] J.-S. Lee, "Performance Evaluation of IEEE 802.15.4 for Low-Rate Wireless Personal Area Networks," *IEEE Transactions on Consumer Electronics*, Aug 2006.

[9] F. Chen, R. German, and F. Dressler, "Towards IEEE 802.15.4e: A Study of Performance Aspects," in *2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, March 2010, pp. 68–73.

[10] F. Chen, T. Talanis, R. German, and F. Dressler, "Real-Time Enabled IEEE 802.15.4 Sensor Networks in Industrial Automation," in *2009 IEEE International Symposium on Industrial Embedded Systems*, July 2009, pp. 136–139.

[11] J. Lee and W. C. Jeong, "Performance Analysis of IEEE 802.15.4e DSME MAC Protocol Under WLAN Interference," in *2012 International Conference on ICT Convergence (ICTC)*, Oct 2012, pp. 741–746.

[12] D. D. Guglielmo, A. Seghetti, G. Anastasi, and M. Conti, "A Performance Analysis of the Network Formation Process in IEEE 802.15.4e TSCH Wireless Sensor/Actuator Networks," in *2014 IEEE Symposium on Computers and Communications (ISCC)*, June 2014, pp. 1–6.

[13] IEEE 802.15.4 WPAN Task Group, "802.15.4e-2012: IEEE Standard for Local and Metropolitan Area Networks - Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer," April 2012.

[14] K. S. J. Pister and L. Doherty, "TSMP: Time Synchronized Mesh Protocol," in *IASTED Internatinal Symposium on Distributed Sensor Networks*. Acta Press, 2008, pp. 391–398.

[15] N. Petreska, H. Al-Zubaidy, B. Staehle, R. Knorr, and J. Gross, "Statistical Delay Bound for WirelessHART Networks," in *Proceedings of the 13th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor and Ubiquitous Networks*. New York, NY, USA: ACM, 2016, pp. 33–40.

[16] N. Petreska, H. Al-Zubaidy, R. Knorr, and J. Gross, "On the Recursive Nature of End-to-End Delay Bound for Heterogeneous Wireless Networks," in *IEEE International Conference on Communications 2015 (ICC 2015)*. London: IEEE, June 2015, pp. 5998–6004.

[17] H. C. Foundation, "WirelessHART®Technology," 2013. [Online]. Available: http://www.hartcomm.org/

[18] ISA100 Commitee. (2014) ISA100, wireless systems for automation. [Online]. Available: www.isa.org/isa100

[19] N. Petreska, H. Al-Zubaidy, and J. Gross, "Power Minimization for Industrial Wireless Networks under Statistical Delay Constraints," in *Teletraffic Congress (ITC), 2014 26th International*. Karlskrona: IEEE, Sept 2014, pp. 1–9.

Improving MAC Layer Reliability Through Interference-Adaptive TSCH Cell Assignment

Leo Krüger, Lotte Steenbrink and Andreas Timm-Giel
Institute of Communication Networks (ComNets)
Hamburg University of Technology
Email: {leo.krueger,lotte.steenbrink,timm-giel}@tuhh.de

***Abstract*—Channel hopping and TDMA are often used to avoid interference and thus increase the reliability of Medium Access Schemes in Wireless Sensor Network (WSN). This is a useful, but blind approach: it minimizes the probability for (repeated) collisions, but does not actively avoid them. In this paper, we propose a MAC scheme that is able to predict and adapt to interference, building on existing research in the field of Cooperative Spectrum Sensing (CSS). The proposed scheme is integrated into IEEE 802.15.4e Time Slotted Channel Hopping (TSCH) as a new 6P Scheduling Function, leveraging existing standards and Internet-Drafts optimized for the deployment in Low Power and Lossy Networks (LLNs).**

I. Introduction

The demands on Wireless Sensor Networks (WSNs) in industrial environments are changing. Whereas previous deployments of such networks encompassed mostly uncritical applications such as environmental sensing, the focus has shifted to use cases with stringent requirements with regard to delay and packet loss. Examples for this are control and monitoring of machinery as well as the usage of a WSN in networked control loops.

To satisfy these requirements, Time Division Multiple Access (TDMA) and/or channel hopping-based Medium Access Schemes are used. This helps optimize medium utilization and avoid packet collision as timeslots are assigned only once, but it also introduces delays: First, the allocation of timeslots defines the time a packet has to wait before it can be transmitted on the link. This is especially relevant for nodes forwarding packets, as the receive timeslot of a packet is not necessarily close to its forwarding timeslot. Second, retransmissions lead to transmission delays which are amplified if the time until the next allocated timeslot is particularly long. Third, non-empty queues on nodes might lead to a queuing delay when either the offered load cannot be satisfied by the slot allocations or retransmissions lead to more timeslots being required. Testbed measurements [1] already have shown that retransmissions as well as queuing have a significant impact on the delay in networks as considered here.

Consequently, delays can be reduced by either optimizing the slot allocation, e.g. by taking into account the paths in the network, their offered load, and an over-provisioning to cope with a certain amount of retransmissions. Or by reducing the amount of retransmissions through avoiding inter-network-interference. In this paper we will focus on the latter, presenting a Scheduling Function aimed at avoiding interference from nodes outside the network. This Scheduling Function is a heuristic where nodes autonomously coordinate the use of `(channel, timeslot)` resources (so-called *cells*) according to their local knowledge about interferences on different channels. Therefore, the allocation of cells adapts to the interference present. In a second step, interference prediction is also taken into account.

Our application scenario incorporates industrial environments in which several hundred mobile IEEE 802.15.4e nodes are deployed without strict prior planning. Co-located IEEE 802.11 (WLAN) and 802.15.1 (Bluetooth) networks are the main source of interference. It has already been shown that IEEE 802.15.4 is affected by such networks. It is expected that the effect is mitigated by channel hopping, but not neutralized completely [2]. The traffic model is peer-to-peer communication without any central sink. Among others, it includes time critical applications. The used hardware is equipped with two radios, which allows for parallel spectrum sensing and sending. Nevertheless, single-radio setups are not to be ignored.

For the moment, we aim at reducing delays in a network with only best-effort service by decreasing the number of necessary retransmissions. For that, we present an approach where nodes autonomously and dynamically adjust their scheduling according to sensed interference. In the future, additionally guaranteeing application dependent delay bounds will be the focus of our work.

This paper is organized as follows. First, related work is summarized. In the subsequent section the corresponding prerequisites such as the Time Slotted Channel Hopping (TSCH) Medium Access Scheme and existing Scheduling Functions are introduced. In Section IV a novel interference-adaptive Scheduling Function is proposed. The paper concludes with an outlook on envisioned evaluation methodology and our future research direction.

II. Related Work

Proposals to mitigate network interference can be categorized into two approaches: channel estimation and spectrum sensing.

In the former, network characteristics such as the Packet Delivery Ratio (PDR) are monitored to filter out channels that negatively affect the network performance. To do this, Muraoka et al. Muraoka et al. propose a housekeeping mechanism that tracks packet collision rates and relocates affected cells

[3], which is a way to mitigate interference without directly measuring or predicting it. It is also important to note that cells are assigned randomly on initialization. It is up to the housekeeping mechanism to improve the initial choice over time.

In the latter, the spectrum is sensed – either by a dedicated antenna or in between transmissions – in order to detect and avoid interferences. Proposals to leverage spectrum sensing for WSNs can be found. Yoon et al. group nodes into parent-child clusters which switch their shared hopping sequence when an interference is detected [4]. Du et al. propose A-TSCH [5], an extension that uses spectrum sensing to detect channels with high noise levels: nodes blacklist either a fixed number of channels or those that fall below a certain quality threshold. Blacklists do not appear to contain information about the channel quality and are distributed among all neighboring nodes. Testbed experiments with an unspecified number of nodes and fixed blacklist sizes (i.e. the N worst channels are blacklisted regardless of their overall quality) suggest that this technique improves the average PDR and stabilizes the network time synchronization when compared to blind channel hopping.

Spectrum sensing for interference avoidance has also been researched extensively in the field of Cooperative Spectrum Sensing (CSS), which is used in Cognitive Radios to enable Secondary Users (SUs) to use a frequency assigned to a Primary User (PU) when the PU is not active. This is done by sensing the medium for any activity from the PU, and stopping any transmissions. SUs may compare their measurements to collaboratively determine whether the PU is actually present and predict its future activities. Usually, a dedicated channel is used for control traffic. This is by definition a best-effort service only; the optimization lies in making the channel available at all. The collaborative interference measurement evaluation of CSS is not needed in the scenario at hand – if one node is experiencing interference, this affects the network, regardless of who else has similar measurements.

The intersection between CSS and Interference-Adaptive Frequency Hopping lies in interference prediction, which has been discussed more widely in the Cognitive Radio world: several authors [6], [7] [8], [9] propose and analyze models to predict WLAN interference. For IEEE 802.15.4e networks, only Vasseur et al. describe a mechanism to exchange prediction information [10], but the predicted data is mainly changes in the traffic forwarded by the node itself.

III. Prerequisites

Multiple standards specifying Medium Access schemes for WSNs are already available. Adaptions for the the original IEEE 802.15.4 standard MAC and Network layer such as WirelessHART, ISA 100.11a and the IEEE 802.15.4e amendment have been specified. All of the above support TDMA and partially also channel hopping: time-frequency resources – so-called timeslots – are exclusively assigned to certain communication partners. This helps reduce intra-network interference, as traffic is divided over multiple frequencies which

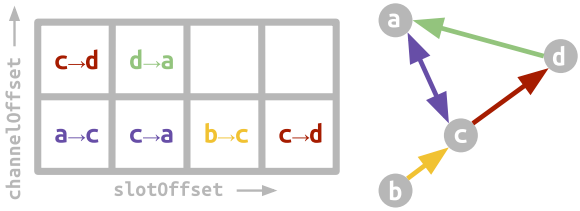


Fig. 1. TSCH cell assignment.

are dedicated to specific nodes at specific times, avoiding collisions. Being able to operate on different channels per timeslot also expands the amount of available resources. If all nodes employ different hopping patterns, they will only briefly be exposed to external interference if it remains on one channel of their hopping pattern.

The original IEEE 802.15.4 standard already defines a mode where timeslots in a star topology are used. Within a superframe the contention free period allows for 7 exclusive timeslots. Since this does not scale well for a higher number of nodes and a mode for multi-hop networks is not defined, IEEE 802.15.4e introduces Time Slotted Channel Hopping (TSCH) – a MAC scheme providing the infrastructure for distributed TDMA channel hopping. However, many parameters such as slotframe size and time synchronization intervals as well as the exact joining and negotiation process are not defined by the standard. The IETF 6TiSCH Working Group aims to fill this void by standardizing protocols and best practices for the use of TSCH with IPv6 in Low Power and Lossy Networks (LLNs). Its 6top sublayer defines the 6top Protocol (6P) [11], which specifies *6P Transactions* to negotiate cells for data transmission off the network's default hopping pattern. Cells are `(channelOffset, slotOffset)` tuples that mark a window in which directed communication between two nodes can take place. `channelOffset` is always relative to the default hopping pattern. Fig. 2 illustrates how the 6top sublayer introduced by 6TiSCH is integrated into the network stack.

A. Scheduling Functions

6TiSCH defines Scheduling Functions, which initiate 6P Transactions to schedule new cells or reschedule existing ones. They determine which cells (if any) are most suitable for all participants. Scheduling Functions can be divided into two categories: interface-based and link-based. The former assigns channels locally per-interface, while the latter assigns dedicated channels to specific transmissions.

So far, two Scheduling Functions have been proposed – Scheduling Function 0 (SF0) [12] and Scheduling Function 1 (SF1) [13]. The former is an active Working Group document, while the latter is an independent draft. The main difference is that SF0 is interface-based and provides a best-effort service where cells are scheduled hop by hop and can be used by multiple links, while SF1 is a link-based Scheduling Function, using the Resource Reservation Protocol (RSVP) to allocate the available bandwidth. SF1 is optimized for industrial net-

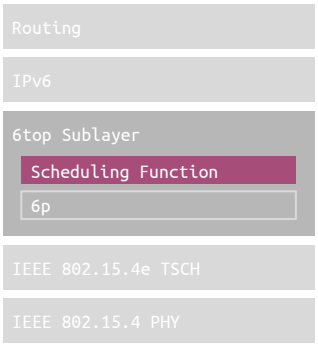


Fig. 2. TSCH/6top Stack. The proposed Scheduling Function is a part of the 6top sublayer, which completes the functionalities provided by TSCH. It uses 6P for cell negotiation.

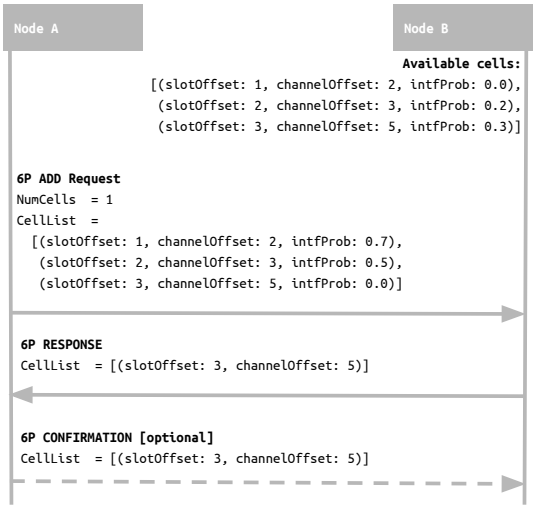


Fig. 3. An example 6P Transaction negotiating a cell to use between nodes A and B by including SF2 interference information `intfProb` as mandated by SF2.

works and aims to make hard real-time guarantees, but it appears to be in an early stage.

IV. Scheduling Function

In order to obtain interference-adaptive cell allocation, a new Scheduling Function SF2 on the basis of SF0 is proposed. The goal is to investigate whether the proposed mechanism is able to reduce delays and packet loss in the best-effort service offered by a hop-by-hop distributed Scheduling Function without dedicated cells. This is why SF0 was chosen as a preliminary basis: due to its simplicity, it can be adapted and customized easily. Any positive effects will still be visible. If the proposed optimization proves to be useful, the new Scheduling Function can then be adapted to support dedicated link-based cell allocation if required, enabling it to make real-time guarantees.

SF2 introduces two new SF Triggering events: *Interference Detected* and *Interference Predicted (optional)*. They are initiated by an Interference Detection Mechanism (IDM) which sweeps all channels for interferences and informs an Interference Prediction Mechanism (IPM). If the IDM detects an interference, it notifies SF2 by triggering the Interference Detected event. SF2 then adds all cells affected by this interference in the near future to a blacklist, and triggers a rescheduling which moves transmissions currently using the affected cells to different, usable cells.

Interference is not a binary matter, but a question of whether a channel is noisy enough to disrupt data transmission or not. Therefore, the blacklisted cells are provided with an interference probability metric which denotes the probability that a transmission over this cell will be faulty. The same goes for predictions: since the IPM cannot always be 100% confident about its predictions, the interference probability metric here refers to the probability that the predicted (critical) interference actually occurs.

The interference probability metric is also included in the CellList of the 6P transaction process illustrated in Fig. 3. This way, nodes can find cells that might work for both of them even in the face of slight interference. Cells whose interference probability is above a certain threshold are not taken into account when negotiating new links with neighbors.

Fig. 3 illustrates the negotiation process for a new set of cells: First, node A sends an ADD request which includes the interference probability in the CellList, along with the number of cells to be scheduled (`NumCells`). Node B then picks those `NumCells` cells from the list that have minimal interference probability for both sides and communicates its choice in the 6p response. The cell selection takes into account both the absolute interference probability of both sides as well as the difference between them: among those cells with the lowest interference probability on both sides, pick the cell with balanced interference probabilities to avoid cases where the interference probability is high on one side while it is low on the other. In such a case the high probability on one side most probably leads to a retransmission, either due to a lost packet or acknowledgment.

Had Node A only communicated the `slotOffset` and `channelOffset` of its available cells, Node B might have picked (1, 2), since it is the first entry in CellList and has an interference probability `intfProb` of 0 for Node B. For A, however, this would have been the worst outcome, disrupting communications for both of them.

SF2 has a basic mode operation (SF2.0) as well as an optional add-on (SF2.1).

A. SF2.0: Interference Detection

In this basic configuration, SF0 is equipped only with the IDM which samples the medium and notifies the Scheduling Function if it detects an interference. If the interference probability is above a certain threshold, SF2 adds all affected cells to a blacklist, and triggers a rescheduling which moves transmissions currently using the affected cells to different, usable cells. Since the IDM is re-run periodically, affected cells will be re-added in case the interference persists. Otherwise, the blacklisting expires automatically.

B. SF2.1: Interference Prediction

As an optional feature, SF2 should not just detect interferences, but also predict when and for how long they might re-occur. This is especially valuable in the face of a co-located IEEE 802.11 network, which periodically broadcasts beacons with a maximum transmission power that easily overwhelms IEEE 802.15.4e networks. The IPM therefore analyzes past IDM data to predict future interferences strong enough to disrupt data transmission. Cells with an interference probability metric value value above a certain threshold are blacklisted or re-scheduled. Various IPMs could be used for this purpose; one of them is the binary time series-based tactic proposed by [6]. It has to be verified that these approaches can also be used on embedded devices.

Note that since the actual channel numbers are abstracted away in 6TiSCH, information form the PAN Information Base (PIB) may have to be used to be able to target the problematic channels identified by the IDM and IPM. The exact mechanics of the IDM and IPM are out of the scope of this paper.

V. Evaluation

The proposed Scheduling Function will be evaluated by discrete event simulation first. In particular, it has to be assessed whether SF2.0 and SF2.1 can improve reliability compared to SF0 and SF1 while outweighing the additional effort introduced. SF2.1 heavily relies on the assumption that external interferer exist that follow a periodic traffic model. However, if interference would be seen mostly at irregular intervals a prediction might be infeasible.

We are considering a simulation scenario where the WSN consists of randomly placed nodes. These try to acquire one slot assignment towards each direct neighbor. A full buffer traffic model is assumed, so that each slot assignment is used. Routing will be disregarded for the moment, as the proposed interference-adaptive Scheduling Function only works on a per-link basis. Multiple IEEE 802.11 networks are co-located reassembling a realistic industrial scenario. Here, different traffic models have to be tested as this is an input for the IDM.

Metrics of interest are e.g. the ratio of non-satisfiable assignments (due to nodes only having blacklisted or already assigned slots left), the packet loss rate, and the mean quality of assigned slots throughout the network.

VI. Conclusion and Future Work

Previous work has suggested that interference detection could be a feasible approach to increasing MAC-layer reliability in WSN and that interference-prediction mechanisms have successfully been used in CSS deployments. This document proposes a way to integrate these results into the TSCH/6TiSCH protocol suite.

The obvious next step is to study and adapt the presented concept. The proposed blacklisting mechanism might be vulnerable to a Denial-of-Service attack jamming specific channels for short time periods. Mitigating this will be part of subsequent protocol design iterations. For a stable environment to compare and smooth out parameters such as the used IPM or cell selection formula, simulation is to be used in the fist step. In case of promising results, they should be verified on real hardware in a testbed.

If shown to be useful, the interference prediction might be extended to be done collaboratively by sharing observations and conclusions among neighboring nodes. However, this approach likely requires a lot of control traffic for status updates (especially if the nodes are not mobility-aware) and introduces a reporting delay, as noted by [14].

Additionally, it would be desirable to remove the dependency on dual-radio hardware. This could be done by introducing a hybrid approach that senses during unused time slots and tries to use packet loss data from cells that are in use to extrapolate interference information.

Acknowledgment

This publication is a result of the research work in the project DRAISE which is funded by German Federal Ministry of Education and Research (BMBF) FKZ 16KIS0429.

References

[1] M. Vilgelm, M. Grsu, S. Zoppi, and W. Kellerer, "Time Slotted Channel Hopping for smart metering: Measurements and analysis of medium access," in *2016 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Nov 2016.

[2] D. Yang, Y. Xu, and M. Gidlund, "Wireless coexistence between IEEE 802.11- and IEEE 802.15.4-based networks: A survey," in *IJDSN*, 07 2011.

[3] K. Muraoka, T. Watteyne, N. Accettura, X. Vilajosana, and K. S. J. Pister, "Simple Distributed Scheduling with Collision Detection in TSCH Networks," *IEEE Sensors Journal*, vol. 16, no. 15, 2016.

[4] S. U. Yoon, R. Murawski, E. Ekici, S. Park, and Z. H. Mir, "Adaptive Channel Hopping for Interference Robust Wireless Sensor Networks," in *2010 IEEE International Conference on Communications*, May 2010.

[5] P. Du and G. Roussos, "Adaptive time slotted channel hopping for wireless sensor networks," in *2012 4th Computer Science and Electronic Engineering Conference (CEEC)*, Sept 2012.

[6] S. Yarkan and H. Arslan, "Binary Time Series Approach to Spectrum Prediction for Cognitive Radio," in *2007 IEEE 66th Vehicular Technology Conference*, 2007.

[7] L. Yang, L. Cao, and H. Zheng, "Proactive channel access in dynamic spectrum networks," *Physical Communication*, vol. 1, no. 2, 2008.

[8] Y. Song, S. Member, J. Xie, and S. Member, "ProSpect : A Proactive Spectrum Handoff Framework for Cognitive Radio Ad Hoc Networks without Common Control Channel," *IEEE Transactions on Mobile Computing*, vol. 11, no. 7, 2012.

[9] S. Geirhofer, S. Member, L. Tong, and B. M. Sadler, "Cognitive Medium Access : Constraining Interference Based on Experimental Models," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 1, 2008.

[10] J. Vasseur, P. Thubert, and P. Wetterwald, "Predictive time allocation scheduling for tsch networks," Jan. 21 2016, US Patent App. 14/336,250.

[11] Q.Wang and X. Vilajosana, "6top Protocol (6P)," Internet-Draft draft-ietf-6tisch-6top-protocol-05, May 2017, work in progress.

[12] D. Dujovne, L. Grieco, M. Palattella, and N. Accettura, "6TiSCH 6top Scheduling Function Zero (SF0)," Working Draft, IETF Secretariat, Internet-Draft draft-ietf-6tisch-6top-sf0-03, March 2017.

[13] S. Anamalamudi, M. Zhang, A. Sangi, C. Perkins, and S. Anand, "Scheduling Function One (SF1) for hop-by-hop Scheduling in 6tisch Networks," Working Draft, IETF Secretariat, Internet-Draft draft-satish-6tisch-6top-sf1-03, February 2017.

[14] I. F. Akyildiz, B. F. Lo, and R. Balakrishnan, "Cooperative spectrum sensing in cognitive radio networks : A survey," *Physical Communication*, vol. 4, no. 1, 2011.

Stop Waiting: Mitigating Varying Connecting Times for Infrastructure WiFi Nodes

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

Jan Heitmann
Research Group smartPORT
Hamburg University of Technology
jheitmann@tuhh.de

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

***Abstract*—Increased spreading of the Internet of Things (IoT) in industrial applications but also home automation impairs the problem of a variety of different gateways and interoperability between connected devices. Connecting smart devices without additional infrastructure among each other and to the Internet is possible relying on existing WiFi networks. To ensure prolonged lifetime of battery- or environmentally-powered sensor nodes, multiple demand-planning algorithms exist. Currently, exact demand planning of WiFi nodes is hindered by highly fluctuating connecting times to their central gateway. This leads to wrong estimation of the actual energy consumption and consequently shortened lifetime or increased latency. We show that the high energy demand of long connecting times can be tackled by aborting the connection attempt and retrying later. Additionally, this reduces the variance of connecting times, making prediction of the energy demand more reliable. Our investigations show that power consumption can be reduced by up to 30% dependent on application. This leads to longer lifetime and paves the way for smart infrastructure relying on WiFi sensor nodes.**

I. Introduction

The growing demands on efficiency of industrial environments, e.g. port areas or larger production plants, lead to spreading of the Internet of Things (IoT) in various scenarios. While classical sensor networks require several gateways with different radio standards [1], the established WiFi standard offers seamless integration of Internet access and interoperability between different vendors. Furthermore, many cities offer or plan public WiFi networks, most industry sites maintain their own WiFi infrastructure and nearly every household has its private WiFi network.

While the research on intermittent computing draws considerable attention in literature, e.g. in [2] or [3], many industrial applications require their systems to be always available and down-times are unwanted. Intelligent infrastructure for transport systems as presented in [4] needs to be always available but not necessarily as fast as possible. A smart road sign may be used for monitoring environmental conditions, e.g. monitoring fine dust pollution, but also needs to transmit status information about battery level or if it fell on the ground due to a collision. Since reacting on these events, e.g. prohibiting diesel-fueled cars from entering inner city zones, takes a considerable time, latency of communication is not the biggest concern. However, a minimum degree of availability has to be ensured, which is hardly feasible with intermittent computing. Thus, duty cycle adjusting algorithms may slow down reaction

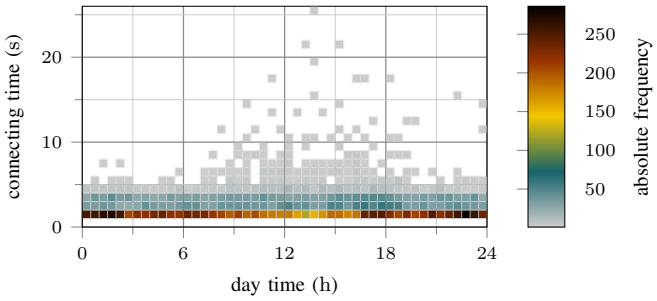


Fig. 1. Connecting times to access point by time of the day; connecting times are influenced by load of surrounding networks producing traffic during regular working times.

time to ensure perpetual operation. To employ these sensor networks with low maintenance effort, a self-sustained energy resource is needed. Whether this is solar energy as presented in [5] or wind in [6], the limited amount needs careful usage of energy. To ensure prolonged lifetime, the highly fluctuating nature of the energy resource has to be tackled with an exact knowledge of sensor nodes future energy demand. Any source of miscalculation has to be eased to facilitate self-sustained operation while simultaneously matching application requirements of latency and reception ratio.

With this in mind, we present the issue of varying connecting times with small WiFi sensor nodes relying on these algorithms and show a strategy how to reduce this impact.

II. Analysis

To enable permanent availability of the sensor system, carefully adjusting the duty cycle of the sensor node is mandatory. The classical power saving mode in infrastructure WiFi with star-topology requires periodical wake-up for receiving beacons of the associated access point. As shown in [7], the power consumption of a WiFi sensor node staying connected during sleep mode is heavily influenced by the beacon interval of the access point. E.g, the power consumption of the Arduino Nano rises from $4\,\mathrm{mW}$ to $7\,\mathrm{mW}$ while halving the beacon interval. A typical beacon interval is $100\,\mathrm{ms}$; thus, duty cycling is very limited. Actuators, requiring faster response time, might use this operating mode, but for classical sensing-

only applications, e.g. waking up every 60 s gathering sensor data, this approach is inefficient. Consequently, disassociating from the access point, going to sleep and re-associating upon next wake-up promises lower energy demand for longer sleep periods. The time at which re-associating is beneficial varies with the used hardware platform. For the ESP 8266 [8], we obtained through early measurements that re-associating is beneficial with sleep times longer than 10 s.

Association to the access point includes several steps and differs upon used authentication protocol [9]. Furthermore, if the Dynamic Host Configuration Protocol (DHCP) is used, obtaining a valid IP address also takes time. Thus, re-associating has to be carefully balanced against the classical power save mode in WiFi.

The resulting connecting times to the network during our long-term test are shown in Fig. 1. To emulate easy-to-use operation, we used a standard WiFi router with open-source firmware `dd-wrt` and enabled DHCP inside our University building at an operational frequency of 2.4 GHz. During times of high traffic, e.g. in the afternoon, connecting times to the network are considerably higher; in some cases up to 25 s. The resulting energy consumption increases significantly: the whole connecting phase is spent in a power-hungry state. For comparison, this consumes about 5% of the energy stored in a 50 F super-capacitor at 2.7 V.

Furthermore, algorithms adjusting the duty cycling of the sensor node, e.g. [10] or [11], rely on accurate planning of the energy demand. This makes their calculation prone to outliers within short-term prediction horizons. Thus, preventing the node from a fluctuating energy demand is inevitable for correct duty cycling.

To assess the impact of varying connecting times, we present a theoretical model and show a strategy for alleviating this impact. Our simulation results show that the variation of energy demand can be decreased, which leads to more accurate demand planning. The first step of our approach is to assess the point in time at which aborting the connecting process potentially reduces the energy demand of the sensor node.

III. Model Description

The Cumulative Distribution Function (CDF) of connecting times, presented in Fig. 2, shows that connection to the network can be established either quite fast or takes considerably longer, e.g. being connected in at most 1300 ms has a probability of 0.7. Consequently, it is more likely to establish a connection during the first part of the trying period compared to the later section. Finding the border between these two phases is the key to enable the following procedure: aborting the connecting process and trying again afterwards. This is promising, since the overall time spent in the connecting state with active radio module may be reduced.

A. Determining the Lower Bound

Mathematically, we are looking for the time t_a at which aborting the connecting process and retrying at a later point

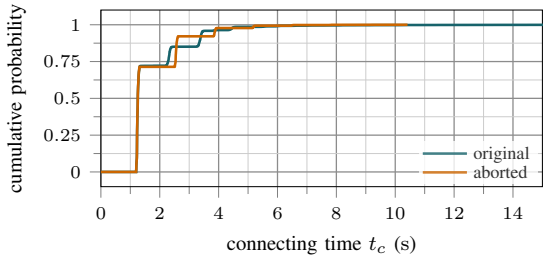


Fig. 2. Comparison of CDFs between originally recorded data and connecting times achieved when aborting and retrying to connect after t_a.

leads to an overall decreased connecting timespan. This decreases the energy consumption of the node.

Assuming the connecting times t_c can be described by a random variable C with Probability Density Function (PDF) $f_C(t)$; the expectation value of the connecting time without aborting is defined as

$$\mathbf{E}[C] = \int_0^\infty t \cdot f_C(t)\mathrm{d}t. \tag{1}$$

If we reached t_a, the expected value of the connecting time additional to t_a is defined as

$$\mathbf{E}[C \geq t_a] = \frac{1}{1 - F_C(t_a)} \int_{t_a}^\infty (t - t_a) \cdot f_C(t)\,\mathrm{d}t, \tag{2}$$

which can be rewritten as

$$\mathbf{E}[C \geq t_a] = \frac{1}{1 - F_C(t_a)} \int_{t_a}^\infty t \cdot f_C(t)\,\mathrm{d}t - t_a, \tag{3}$$

with $F_C(t_a)$ denoting the CDF at time t_a. Basically, Eq. (3) is the expectation value of being connected after already waiting for t_a scaled with the probability that this event occurs. The desired time t_a is reached, if the expectation value of aborting and retrying is smaller or equal than trying infinitely. Consequently, we combine Eq. (3) and Eq. (1) so that

$$\frac{1}{1 - F_C(t_a)} \int_{t_a}^\infty t \cdot f_C(t)\,\mathrm{d}t - t_a \leq \int_0^\infty t \cdot f_C(t)\mathrm{d}t$$

$$\Rightarrow \int_{t_a}^\infty t \cdot f_C(t)\,\mathrm{d}t \leq (1 - F_C(t_a)) \left(t_a + \int_0^\infty t \cdot f_C(t)\,\mathrm{d}t \right) \tag{4}$$

holds. This equation can be numerically solved with values obtained from long-term measurements. Note that t_a is the first point in time, at which the energy consumption decreases when aborting the connecting process; thus, t_a is a lower bound for all reasonable aborting times.

B. Impact on Delay

A classical sensor network application gathers sensor data in a constant time interval τ and reports it regularly. Depending on the application, the receiving side has fixed latency requirements, meaning a packet arriving before a critical delay d_c. The accepted amount of outdated packets defines a demanded

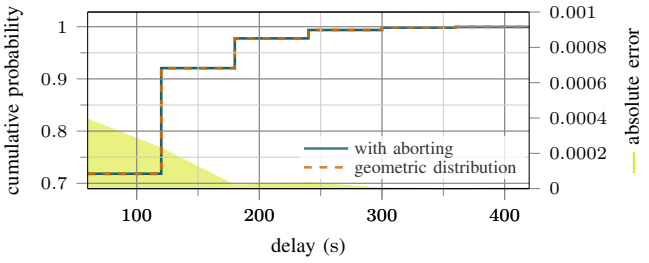


Fig. 3. Cumulative probability of delay; aborting connecting and delaying report to next report attempt leads to geometrically distributed delay.

reception ratio. In case of a fixed reception ratio, the benefit of packets arriving much faster is minor. This can be exploited by delaying a transmission to the next regular reporting time instead of spending a large amount of energy on trying to transmit the packet as fast as possible.

Beginning with the first transmission attempt, the situation is similar to throwing a coin with success probability p. Each of the connection attempts n is expected to have the same success probability p. Since delay is primarily influenced by the number of transmission attempts until one success occurs, it is described by a geometric distribution with PDF

$$f_D(n) = (1-p)^{n-1} \cdot p. \tag{5}$$

In our case, the success probability is determined by the aborting time t_{ab}, which is lower bounded by t_a, so that $t_a \leq t_{\text{ab}}$ holds. Consequently, $p = F_C(t_{\text{ab}})$. Furthermore, the delay introduced by one transmission attempt $n \in \mathbb{N}, n \leq n_{\text{max}}$ increases the delay by $t = n \cdot \tau$. This rewrites Eq. (5) to:

$$f_D(t) = (1 - F_C(t_{ab}))^{(t/\tau - 1)} \cdot F_C(t_{ab}). \tag{6}$$

The overall reporting delay also has a second component. Once the connection to the access point is established, the queued packets have to be transmitted in addition to the newest packet. As stated in [12], the service time of the common Distributed Coordination Function (DCF) can be approximated by an exponential distribution with PDF

$$f_S(t;\lambda) = \lambda e^{-\lambda t} \quad \text{for} \quad t \geq 0. \tag{7}$$

We determined the parameter λ by performing a curve fitting of the observed delay distribution, which we omit here due to space constraints. The resulting reporting delay distribution is displayed in Fig. 3 along with the connecting delay obtained by simulations.

C. Impact on Current Consumption

The primary goal for alleviating the impact of fluctuating connecting times is to decrease the energy consumption of the sensor node. Our model assumes the node to be in one of three different states: deep-sleep, active or associating to a WiFi network, and actively transmitting. This results in three approximated currents I_{ds}, I_{ac} and I_{tx} drawn by the communicating device. The power consumption follows

TABLE I
CURRENT CONSUMPTION ESPRESSIF ESP 8266

state	variable	consumption
deep sleep	I_{ds}	26 µA
actively connecting	I_{ac}	69.47 mA
transmitting	I_{tx}	204.10 mA

directly given a constant supply voltage. The actual values in these states can be obtained from measurements and differ between devices, e.g. $I_{\text{ac}} \approx 84.1\,\text{mA}$ and $I_{\text{tx}} \approx 105\,\text{mA}$ for the Arduino MKR1000. Consequently, the resulting overall current drawn during T can be calculated as follows:

$$I^* = \frac{1}{T} \cdot \Big(I_{\text{ds}} \cdot (T - t_{\text{ac}} - t_{\text{tx}}) + (I_{\text{ac}} - I_{\text{ds}}) \cdot t_{\text{ac}} + (I_{\text{tx}} - I_{\text{ds}}) \cdot t_{\text{tx}} \Big). \tag{8}$$

t_{ds}, t_{ac} and t_{tx} denote the time spent in the corresponding states. Note that $T = t_{\text{ds}} + t_{\text{ac}} + t_{\text{tx}}$. The key idea behind aborting and delaying transmission attempts is to reduce the current consumption by reducing time spent in the costly states while simultaneously satisfying the application inherited reception ratio determined by the delay boundary d_c.

IV. EVALUATION

A brief overview of our measurement platform is covered in the first part of the section, whilst the second presents the simulation results of the approach introduced in Section III.

A. Hardware and Setup

The Espressif ESP 8266 [8] offers a 32 bit 160 MHz CPU with 160 kB RAM and is available for less than €3. It is fully compatible with IEEE 802.11b/g/n at a size of $16 \times 26\,\text{mm}$.

We determined the current consumption with an INA139 measurement amplifier with measurement shunts $R_S = 1\,\Omega$ in active states and $R_S = 1\,\text{k}\Omega$ in deep-sleep state. We summarized the measurement results in Table I.

The connecting and reporting delays were recorded during a 16-days long-term test in our university building. The WiFi node attempts to connect to a private access point with no other wireless clients in the network. We used the data to determine the parameters of our model presented in Section III and performed simulations with MATLAB to evaluate our approach to reduce the current consumption of the node.

B. Results

Based on the recorded connecting time data already presented in Fig. 2, the algorithm evaluating Eq. (3) yields a lower bound of aborting times $t_a = 1313\,\text{ms}$ with $F_C(t_a) = 0.72$. Evaluating the approach of aborting after t_a and attempting to connect directly afterwards, yields connecting times with CDF shown in Fig. 2. While the mean value of connecting times can only be reduced from 1790 ms to 1767 ms, the standard deviation can be reduced from 1164 ms to 995 ms.

Delaying the transmission attempt to the next regular sampling time decreases the connecting times significantly: the

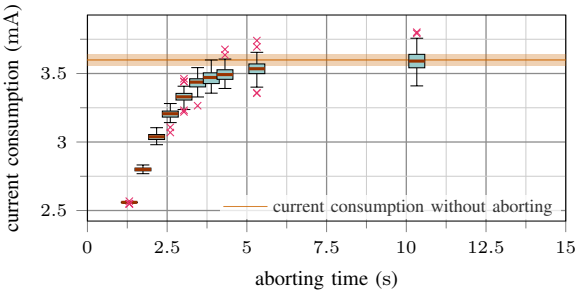


Fig. 4. Mean current consumption w.r.t. different aborting times; aborting reduces time spent in energy-costly state; thus, current consumption decreases by 30% to 4%; current consumption without aborting is shown as median; shaded area indicates upper and lower quartiles.

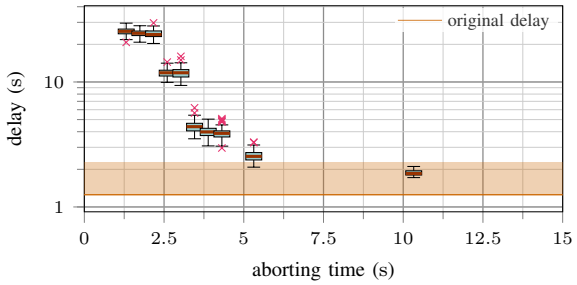


Fig. 5. Packet delay increased with short aborting times and relaxes for longer aborting times; original delay is shown as median; shaded area indicates upper and lower quartiles; note that the upper quartile is more than twice the median.

mean connecting time is 1265 ms with standard deviation of only 37 ms. This enables demand planning algorithms to be much more reliable.

Figure 4 shows the current consumption of this method with different aborting times beginning with t_a. Using a report interval $\tau = 60\,\mathrm{s}$, the benefit of aborting is clearly visible; e.g. aborting after 1742 ms saves 22% current.

The downside of this approach is increased packet delay. E.g., for an aborting time of 1742 ms the median delay is more than 24 s. Since it is mainly influenced by the time difference between two transmission attempts, the delay scales with τ. We plotted the results for $\tau = 60\,\mathrm{s}$ in Fig. 5 for co-existence with Fig. 4, but the situation slightly relaxes for smaller sampling intervals. This shows that power saving mainly affects delay-uncritical applications; but if the main focus is reliable prediction of energy demand, this approach is highly advisable.

As mentioned earlier, many applications require a fixed reception ratio w.r.t. a critical delay d_c at which a packet is discarded, since it is outdated. We depict the needed current consumption for satisfying different reception ratios or percentile of packets with latency smaller than d_c in Fig. 6. Dependent on the critical delay, the power savings can be considerable. E.g., for an application sampling every 30 s and requiring 90% of the packets to arrive with a delay less than 60 s, current consumption can be reduced by 30%.

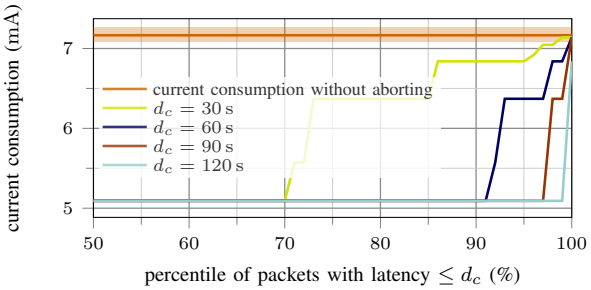


Fig. 6. Current consumption to fulfill demanded reception ratio dependent on delay d_c; relaxing delay requirements or reception ratio highly reduces the current consumption; reporting interval $\tau = 30\,\mathrm{s}$; original current is shown as median; shaded area indicates upper and lower quartiles.

V. Conclusion

We showed that variating WiFi connecting times impact the operation of duty-cycle-adjusting sensor nodes. Our strategy reduces the current consumption for applications with relaxed delay requirements considerably. Additionally, we ease the fluctuation of connecting times, helping demand planning algorithms to work more reliable. This prepares the ground for smart infrastructure using self-sustaining WiFi sensor nodes with low energy budget. Additionally, it allows us to perform real-world tests in harbor environments.

References

[1] 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.

[2] B. Lucia and B. Ransford, "A Simpler, Safer Programming and Execution Model for Intermittent Systems," in *Proceedings of PLDI*. New York, NY, USA: ACM, 2015.

[3] J. Hester, K. Storer, L. Sitanayah, and J. Sorber, "Towards a Language and Runtime for Intermittently-Powered Devices," *sleep*, vol. 9, 2016.

[4] Free and Hanseatic City of Hamburg — State Ministry of Economic Affairs, Transport and Innovation, " Verkehr 4.0 - ITS-Strategie für Hamburg," http://www.its2021.hamburg/downloads/ITS%20Strategie%20Hamburg.pdf, May 2016.

[5] L. Hanschke, J. Heitmann, and C. Renner, " On the Feasibility of WiFi-Enabled and Solar-Powered Sensors for Smart Ports ," in *Proceedings of the 15th GI/ITG KuVS Fachgespräch "Sensornetze"* , ser. FGSN, Augsburg, Germany, September 2016.

[6] C. Park and P. H. Chou, "Ambimax: Autonomous Energy Harvesting Platform for Multi-Supply Wireless Sensor Nodes," in *Sensor and Ad Hoc Communications and Networks, SECON'06.*, vol. 1. IEEE, 2006.

[7] L. Hanschke, J. Heitmann, and C. Renner, " Challenges of WiFi-Enabled and Solar-Powered Sensors for Smart Ports ," in *Proceedings of ENSsys*, Stanford, CA, USA, November 2016.

[8] Espressif Inc., *ESP8266EX Datasheet*, Apr. 2016, rev. 5.4. [Online]. Available: https://espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf

[9] IEEE 802.11 Working Group, "IEEE 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," *IEEE Std*, vol. 802, no. 11, 2016.

[10] C. Moser, D. Brunelli, L. Thiele, and L. Benini, "Real-time Scheduling for Energy Harvesting Sensor Nodes," *Real-Time Systems*, vol. 37, 2007.

[11] 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.

[12] H. Zhai, Y. Kwon, and Y. Fang, "Performance Analysis of IEEE 802.11 MAC Protocols in Wireless LANs," *Wireless Communications and Mobile Computing*, vol. 4, no. 8, 2004.

Einfluss der Gradverteilung bei Multicast Growth Codes

Isabel Madeleine Grimm
Department of Computer Science
University of Wuerzburg
Am Hubland, 97074 Wuerzburg, Germany
Email: grimm@informatik.uni-wuerzburg.de

Reiner Kolla
Department of Computer Science
University of Wuerzburg
Am Hubland, 97074 Wuerzburg, Germany
Email: kolla@informatik.uni-wuerzburg.de

***Abstract*—Multicast Growth Codes (MCGC) basieren auf dem Prinzip der Growth Codes, welche eine spezielle Form der Netzwerkkodierung darstellen. Hierbei werden zuvor empfangenen Daten geeignet kombiniert und anschließend als Codewort an alle direkten Nachbarknoten gesendet. Bei MCGC wird dem System Redundanz hinzugefügt, wodurch Knoten- und Verbindungsausfälle kompensiert werden können. Die Performanz des MCGC Verfahrens wird durch mehrere Parameter beeinflusst. Entscheidend ist hierbei die für die Erzeugung von Codeworten verwendete Gradverteilung. Aus diesem Grund werden im folgenden Artikel die Grundlagen zu der bei MCGC verwendeten Gradverteilung genauer erläutert und deren Einfluss auf die Performanz des Algorithmus beschrieben.**

I. Einleitung

Im Bereich des Internet of Things (IoT) wird häufig eine Vielzahl an kleinsten, kostengünstigen Sensorknoten mit beschränkter Rechenleistung verwendet. Große Sensornetze mit hunderten oder tausenden Knoten dieser Art stellen dabei eine Herausforderung an die Realisierung zuverlässiger drahtloser Kommunikation dar. Häufig auftretende Kollisionen führen zu einer hohen Wahrscheinlichkeit von Übertragungsverlusten. Insbesondere bei der Verwendung unter rauen Bedingungen sind die vielen kleinen Sensorknoten häufig unzuverlässig. Folglich kommt es nicht selten zu Knoten- oder Verbindungsausfällen im Netz. Außerdem müssen alle erhobenen Daten gesammelt werden, um sie später weiter verarbeiten zu können. Werden alle Daten an einer oder wenigen Stellen gesammelt, oder über ähnliche Wege geroutet, entstehen Flaschenhälse, welche extrem anfällig für Ausfälle sind. Betrachtet man ein hochdynamisches Netz mit ständig ändernder Topologie, so kommen noch weitere Herausforderungen hinzu. Um die Zuverlässigkeit großer drahtloser Sensornetze unabhängig von der vorliegenden Dynamik, und Wahrscheinlichkeit von Knoten- oder Verbindungsausfällen zu verbessern, wurde in [1] die Methode der sogenannten Multicast Growth Codes (MCGC) entwickelt. Als Kommunikationsprinzip verwenden MCGC eine all-to-all Kommunikation und vermeiden, durch Multicast-Sendevorgänge ohne zielgerichtetes Routen der Pakete, das Erzeugen von Flaschenhälsen im Netz. MCGC basieren auf dem Prinzip der Growth Codes (GC) [2], [3], welche eine spezielle Form der Netzwerkkodierung [4] darstellen, die zu den Erasure Codes [5] gehört. Bei einer Netzwerkkodierung werden zuvor empfangenen Daten kombiniert und anschließend in kombinierter Form an Nachbarknoten gesendet. Durch die MCGC Netzwerkkodierung wird dem System Redundanz hinzugefügt, wodurch Knoten- und Verbindungsausfälle kompensiert werden können. MCGC benötigen keine komplexen oder aufwändigen Rechenschritte, anders als beispielsweise die, bei Random Linear Network Codes [6], [7] verwendete, Gauß-Elimination. Aus diesem Grund sind MCGC auch auf kleinen Sensorknoten mit geringer Rechenleistung ausführbar. Bei MCGC ist es das Ziel jedes Knotens zu jeder Zeit möglichst viele der, durch das Netz erhobenen, Daten zu speichern. Auf diese Weise wird eine zuverlässige Datenhaltung auch im Falle von Knoten- oder Verbindungsausfällen gewährleistet. Die Performanz des MCGC Verfahrens wird durch mehrere Parameter beeinflusst. Entscheidend hierbei ist die verwendete Gradverteilung. Aus diesem Grund werden im Folgenden die mathematischen Grundlagen zu der Gradverteilung, welche bei den MCGC in [1] verwendet wird, genauer erläutert.

Die vorliegende Arbeit ist wie folgt strukturiert: Nach einer Beschreibung der notwendigen Grundlagen zu den MCGC in Section II, folgt in Section III der Hauptabschnitt über den Einfluss der MCGC Gradverteilung. Hierbei wird Inhalt und Länge der Wartelisten in Section III-A, die Minimierung der Wahrscheinlichkeit redundante Pakete zu empfangen in Section III-B, die Maximierung der Wahrscheinlichkeit sofort dekodierbare Pakete zu erhalten in Section III-C und die letztendlich gewählte Gradverteilung in Section III-D thematisiert. Abschließend folgen in Section IV Stichpunkte zu aktueller und geplanter, zukünftiger Arbeit.

II. Präliminarien

Im Folgenden werden die, für das Verständnis notwendigen, Definitionen und das Prinzip der MCGC beschrieben.

A. Systembeschreibung und Definitionen

Das hier betrachtete System besteht aus n homogenen, drahtlos-kommunizierenden mobilen Sensorknoten mit der Fähigkeit Daten zu erheben, zu speichern, zu kombinieren und weiterzusenden. Im Folgenden wird außerdem angenommen, dass jeder Knoten nur einmal Daten erhebt, d.h. der Zeitpunkt der Datenerhebung kann vernachlässigt werden. Die von einem Knoten selbst erhobenen Daten werden als das *eigene Symbol des Knotens* bezeichnet. Jedes Datenpaket, welches ein Knoten

sendet, beinhaltet ein sogenanntes *Codewort* als Payload und einen Header mit allen nötigen Informationen für die Kodierung und Dekodierung. Ein solches Codeword wird durch XOR-Kombination von Symbolen erzeugt. Die Anzahl der in einem Codewort kombinierten Symbole bezeichnet man als *Grad* des Codewortes. Ein Knoten kann empfangene Codeworte dekodieren und speichert die rekonstruierten, ursprünglichen Symbole ab. Zu diesem Zweck besitzt jeder Knoten einen Speicher S für bereits rekonstruierte Symbole, und eine sogenannte Warteliste für empfangene, noch nicht komplett dekodierte Codeworte.

B. Prinzip der Datenübertragung bei MCGC

Die hier betrachteten MCGC basieren auf unidirektionaler Datenübertragung via Multicast an alle direkten Nachbarknoten. Der besseren Vergleichbarkeit mit den ursprünglichen GC wegen, wurde die Zeit bei MCGC in diskrete Runden unterteilt. Eine Runde ist dabei definiert als die Zeit, die nötig ist, damit alle Knoten nacheinander ein Codewortpaket an jeweils alle Nachbarknoten senden können. Hierzu wird jedem Knoten ein Sendezeitschlitz innerhalb der Runde zugeteilt. Jeder Knoten überträgt einmal pro Runde ein Paket via Multicast an alle Nachbarknoten einmal in der Runde und wartet die restliche Zeit auf Pakete der Nachbarn. Das verwendete Protokoll basiert auf Zeitschlitzen und ist durch synchronisierte Knoten kollisionsfrei. Die Simulationsergebnisse des synchronisierten, rundenbasierten Ansatzes können leicht in Sekunden umgerechnet werden. Diese Umrechnung lässt sich durch Multiplikation der Anzahl benötigter Runden und der Paketübertragungszeit sowie der Gesamtanzahl an Knoten umsetzen.

Ein nicht-synchronisiertes, kollisionsbehaftetes MAC-Protokoll ohne Zeitschlitze, bei welchem Knoten den Sendezeitpunkt zufällig wählen, wurde ebenfalls implementiert und evaluiert. Hierbei wurden die gleichen Simulationsergebnisse erzielt wie im rundenbasierten Ansatz mit entsprechender Verbindungsausfallwahrscheinlichkeit und in Sekunden umgerechneten Werten. In [1] wurde gezeigt, dass MCGC Verbindungsausfälle sehr gut tolerieren können und damit also auch das kollisionsbehaftete MAC-Protokoll verwendbar ist.

C. Überblick Kodierung and Dekodierung bei MCGC

Ziel der MCGC ist es, zu jedem Zeitpunkt, und insbesondere bei jeder Anzahl in einem Knoten empfangener Codeworte, möglichst viele ursprünglich Symbole in diesem bereits rekonstruiert zu haben. Es wird also betrachtet, wie viele Codeworte empfangen werden mussten, bevor eine gewisse Anzahl von Symbolen in dem betrachteten Knoten rekonstruiert werden konnte.

Für die Kodierung werden bei den MCGC die Codeworte, unter Berücksichtigung des momentanen Netzzustandes, speziell zusammengesetzt.

Ein Knoten sendet in einem Codewortpaket, neben dem Codewort selbst, auch den für ihn zum jeweiligen Zeitpunkt optimalen Grad d_{opt}. Hierbei gibt der Knoten mit d_{opt} den Grad an, welchen das nächste von ihm empfangene Codewort im Optimalfall haben sollte. Für die Bestimmung dieses optimalen Grades d_{opt} verwendet der Knoten eine sogenannte Gradverteilung, welche in Section III mathematisch genauer erläutert wird.

Algorithm 1 Creation and transmission of a codeword packet

```
1: procedure CREATEANDSEND
2:     d_opt ← getOwnDesiredDegree()
3:     d ← getCurrentProposedDegree()
4:     tmpList ← chooseDataSets(d)
5:     cw ← createCodeword(tmpList)   // via XOR of chosen data sets
6:     packet ← createPacket(cw, d_opt)
7:     transmitPacket(packet)         // send packet to all neighbors
8: end procedure
```

Bevor ein Knoten ein neues Codewort erzeugt, bestimmt er den für dieses Codewort vorzugebenden Grad d. Hierbei wird d als das Minimum der Wunschgrade d_{opt} ermittelt, welche seit der letzten Codeworterzeugung von den Nachbarknoten an ihn gesendet wurden. Der Knoten wählt anschließend d-viele Symbole aus dem Speicher S, welcher die von ihm bereits rekonstruierten Symbole beinhaltet. Diese d gewählten Symbole werden dann mittels XOR zu einem neuen Codewort zusammengesetzt. Das genaue Verfahren, nach welchem die Symbole für ein Codewort ausgewählt werden, findet sich in [1] und basiert auf der Wahrscheinlichkeit, dass ein jeweiliges Symbol bei den Nachbarn noch nicht empfangen wurde. Algorithm 1 beschreibt das Verfahren wie ein Codewort erzeugt und gesendet wird zusätzlich formal.

Die Dekodierung in einem Knoten beginnt bei den MCGC sobald der betrachtete Knoten ein erstes Codewort empfangen hat. Alle Symbole, welche bereits von diesem Knoten rekonstruiert wurden, werden aus dem empfangenen Codewort via XOR gelöscht. Falls das verbleibende Codewort mehr als ein für den Knoten noch nicht bekanntes Symbol beinhaltet, wird es in nun reduzierter Form in der Warteliste des Knotens gespeichert. Falls das verbleibende Codewort nur genau ein noch nicht rekonstruiertes Symbol beinhaltet, wird das Codewort als dekodiert betrachtet und das Symbol als nun rekonstruiertes Symbol in S gespeichert. Anschließend wird das neu rekonstruierte Symbol aus allen Codeworten der Warteliste gelöscht, welche dieses beinhalten. Auf diese Weise können Codeworte der Warteliste dekodiert und gegebenenfalls weitere Symbole von dem Knoten rekonstruiert werden.

III. Beschreibung und Einfluss der MCGC Gradverteilung

Um zum jeweiligen Zeitpunkt optimal dekodieren zu können, ist es entscheidend, welchen Grad ein ankommendes Codewort besitzt und welche Symbole in diesem enthalten sind. An dieser Stelle soll speziell der mathematische Hintergrund der verwendeten Grade untersucht werden. Die genaue Symbolzusammensetzung wurde bereits in [1] beschrieben.

A. Inhalt und Länge der Wartelisten

Die Warteliste eines Knotens beinhaltet im Optimalfall nur Grad-2-Codeworte. Aus einem empfangenen Codewort werden alle bereits rekonstruierten Symbole via XOR herausgelöscht. Beinhaltet das Codewort anschließend kein noch

nicht rekonstruiertes Symbol, so war das Codewort redundant und wird verworfen. Besteht das Codewort nur noch aus genau einem unbekannten Symbol, so wird dieses Symbol als rekonstruiert und das Codewort als dekodiert betrachtet. Andernfalls wird das verbleibende Codewort mit dem nun reduzierten Grad in der Warteliste gespeichert. Codeworte in der Warteliste besitzen also einen Grad ≥ 2. Außerdem können Codeworte der Warteliste, bei n verschiedenen zu rekonstruierenden Symbolen und r bereits rekonstruierten Symbolen nur maximal einen Grad von $n - r$ haben. Sobald ein neues Symbol rekonstruiert wurde, wird dieses aus jedem Codewort der Warteliste, in dem es enthalten ist, gelöscht und das Codewort wird dadurch ggf. dekodiert. Codeworte der Warteliste sollen so schnell wie möglich dekodiert werden, damit zu jedem Zeitpunkt und mit jeder Anzahl empfangener Codeworte möglichst viele Symbole bereits rekonstruiert sind. In einem Codewort der Warteliste sind ausschließlich noch nicht bekannte Symbole enthalten, da rekonstruierte Symbole zuvor aus dem Codewort gelöscht wurden. Aus diesem Grund ist für ein Codewort in der Warteliste nicht relevant, wie viele Symbole zu dem jeweiligen Zeitpunkt bereits rekonstruiert wurden. Aus Codeworten mit höherem Grad lassen sich ggf. mehr bereits rekonstruierte Symbole kürzen. Es ist jedoch nicht entscheidend, wie viele Symbole aus einem Codewort gelöscht wurden, sondern wie schnell ein Codewort komplett dekodiert werden konnte. Codeworte der Warteliste sollten also einen möglichst niedrigen Grad besitzen, im Optimalfall folglich Grad 2.

Neben dem Grad der Codeworte aus der Warteliste wurde auch die Länge der Warteliste für die optimale Dekodierbarkeit berücksichtigt. Bei n verschiedenen zu rekonstruierenden Symbolen und r bereits rekonstruierten Symbolen liegt die optimale Wartelistenlänge l bei $\frac{n-r}{2} \leq l \leq n - r$.

Aufgrund des Schubfachprinzips aus [8] beinhaltet eine Warteliste mit einer Wartelistenlänge von $l > n - r$ mindestens ein redundantes Codewort, da bereits alle in dem Codewort kombinierten Symbole in anderen Codeworten der Warteliste enthalten sind. Mit redundanten Codeworten kann kein weiteres Symbol neu rekonstruiert werden. Hier würden also bei höherer Anzahl empfangener Codeworte nicht mehr Symbole rekonstruiert werden können als ohne dieses empfangene Codewort. Damit läge keine optimale Dekodierbarkeit vor.

Bei einer Wartelistenlänge von $l = \frac{n-r}{2}$ und der Annahme, dass ausschließlich Grad-2-Codeworte in der Warteliste enthalten sind, wird mit einem neu rekonstruierten Symbol erwartungsgemäß genau 1 Codewort aus der Warteliste dekodiert. Dies lässt sich wie folgt beweisen. Die Wahrscheinlichkeit, dass das neu rekonstruierte Symbol in einem betrachteten Codewort enthalten ist, liegt bei $p = \frac{1}{n-r} \cdot \binom{2}{1} = \frac{2}{n-r}$. Bei einer Wartelistenlänge von $l = \frac{n-r}{2}$ bedeutet dies also, dass mit einem neu rekonstruierten Symbol erwartungsgemäß $l \cdot p = \frac{n-r}{2} \cdot \frac{2}{n-r} = 1$ Codewort aus der Warteliste dekodiert wird. Tatsächlich besitzen jedoch nicht alle Codeworte der Warteliste einen Grad von 2, sondern der Durchschnittsgrad von Codeworten der Warteliste liegt bei ≥ 2. Daher ist mit jedem rekonstruierten Symbol erwartungsgemäß mindestens 1 Codewort aus der Warteliste dekodierbar. Da es das Ziel ist, zu jeder Zeit und mit jeder Anzahl empfangener Codeworte möglichst viele Symbole zu rekonstruieren, sollte nicht weniger als 1 Codewort mit einem neu rekonstruierten Symbol dekodiert werden können. Aus diesem Grund wird eine Wartelistenlänge von $l \geq \frac{n-r}{2}$ angestrebt.

B. Distanz-0-Minimierung

Damit zu jeder Zeit und mit jeder Anzahl empfangener Codeworte möglichst viele Symbole rekonstruiert werden können, sollten bei einem Knoten möglichst keine redundanten Codeworte ankommen. Die Wahrscheinlichkeit, dass ein ankommendes Codewort im Vergleich zur Menge der bereits rekonstruierten Symbole eine Distanz von 0 besitzt, d.h. dass es nur bereits rekonstruierte Symbole beinhaltet, sollte folglich möglichst gering sein. Aus diesem Grund wird dieser Ansatz im Folgenden als Distanz-0-Minimierung bezeichnet. Sei n die Anzahl insgesamt zu rekonstruierender Symbole und r die Anzahl bereits rekonstruierter Symbole. Dann lässt sich die Wahrscheinlichkeit, dass ein empfangenes Codeword von Grad d ein Distanz-0-Codewort darstellt, entsprechend der Wahrscheinlichkeit einer Zufallsgröße bei der Hypergeometrischen Verteilung berechnen:

$$p_{Dist_0}(d) = \frac{\binom{n-r}{0} \cdot \binom{r}{d}}{\binom{n}{d}}. \tag{1}$$

Hierbei wurde angenommen, dass jedes Symbol mit gleicher Wahrscheinlichkeit in einem Codewort enthalten ist.

Betrachtet man Equation (1), so lässt sich erkennen, dass die Wahrscheinlichkeit $p_{Dist_0}(d)$ bei einer gegebenen Anzahl rekonstruierter Symbole r umso geringer ist, je größer der Grad d eines Codewortes ist. Allerdings sollte der Grad von empfangenen Codeworten nicht so hoch sein, dass diese aufgrund ihres Grades nicht mehr dekodierbar sind. Daher wurde eine Schranke p_{min} hinzugefügt, welche von der Wahrscheinlichkeit $p_{Dist_0}(d)$ nicht unterschritten werden darf. Simulationen haben gezeigt, dass $p_{min} = 0.10$ die besten Ergebnisse bezüglich Geschwindigkeit und Zuverlässigkeit liefert. Bei der jeweiligen Anzahl bereits rekonstruierter Symbole r ist der optimale Grad für ein nächstes Codewort der größtmögliche Grad d, für welchen die Bedingung $p_{Dist_0}(d) \geq p_{min}$ noch erfüllt ist. Unter Verwendung von Equation (1) lässt sich außerdem durch Widerspruchsbeweis zeigen, dass der optimale Grad mit fortschreitender Anzahl bereits rekonstruierter Symbole monoton steigt.

C. Distanz-1-Maximierung

Um das Ziel der optimalen Dekodierbarkeit zu erreichen, ist anzustreben, dass ein ankommendes Codewort direkt dekodierbar ist. Die Wahrscheinlichkeit, dass dieses Codewort im Vergleich zur Menge der bereits rekonstruierten Symbole eine Distanz von 1 besitzt, d.h. dass nur genau 1 noch

nicht rekonstruiertes Symbol im Codewort enthalten ist, sollte möglichst groß sein. Aus diesem Grund wird dieser Ansatz als Distanz-1-Maximierung bezeichnet, und wurde bereits bei den GC in [3] als Basis für die Gradverteilung verwendet. Die Wahrscheinlichkeit, dass ein empfangenes Codeword von Grad d direkt dekodierbar ist, lässt sich ebenfalls entsprechend der Hypergeometrischen Verteilung berechnen:

$$p_{Dist_1}(d) = \frac{\binom{n-r}{1} \cdot \binom{r}{d-1}}{\binom{n}{d}}. \tag{2}$$

Hierbei gelten die gleichen Annahmen wie in Section III-B. Durch Einsetzen und Umformung lässt sich berechnen, dass

$$p_{Dist_1}(i+1) \geq p_{Dist_1}(i) \Leftrightarrow r \geq \frac{in-1}{i+1}. \tag{3}$$

Ab $r > \frac{in-1}{i+1}$ bereits rekonstruierten Symbolen ist die Wahrscheinlichkeit, dass ein Codewort direkt dekodierbar ist, größer, wenn das Codewort den Grad $i+1$ statt den Grad i besitzt. Ein Beweis hierzu lässt sich unter [3] finden. Außerdem wurde in [3] über Equation (2) gezeigt, dass der optimale Grad mit der Anzahl bereits rekonstruierter Symbole monoton steigt. Ankommende Codeworte sollten also anfangs den Grad 1 haben. Ab $\frac{n-1}{2}$ bereits rekonstruierten Symbolen sollten Codeworte den Grad 2 und anschließend einen, entsprechend Equation (3), stetig anwachsenden Grad besitzen, um mit größter Wahrscheinlichkeit direkt dekodierbar zu sein.

1) Gradverteilung bei Distanz-1-Maximierung: Die Formel für die verwendete Gradverteilung basiert auf dem sogenannten Coupon Collector's Problem, welches unter anderem in [8] behandelt wird. Details zur Herleitung der im Folgenden verwendeten Formel finden sich im Anhang. Überträgt man die Equation (6) aus dem Anhang auf unser Szenario, so bedeutet dies, dass wir, um r Symbole rekonstruieren zu können, erwartungsgemäß $\sum_{i=0}^{r-1} \frac{n}{n-i}$ Grad-1-Codeworte empfangen müssen. Hierbei wurde angenommen, dass jedes Symbol mit gleicher Wahrscheinlichkeit in einem Codewort enthalten ist und alle Codeworte mit gleicher Wahrscheinlichkeit empfangen werden. Ausgehend von obigem Erwartungswert sind laut [3] erwartungsgemäß maximal

$$K_j = \sum_{h=1}^{j} \sum_{i=R_{h-1}}^{R_h-1} \frac{\binom{n}{h}}{\binom{i}{h-1}\binom{n-i}{1}} \tag{4}$$

Codeworte nötig, um $R_j = \frac{jn-1}{j+1}$ Symbole zu rekonstruieren. Hieraus ergibt sich, dass erwartungsgemäß K_1 empfangene Grad-1-Codeworte nötig sind, um R_1 Symbole zu rekonstruieren. Außerdem werden erwartungsgemäß maximal $K_2 - K_1$ Grad-2-Codeworte für $R_2 - R_1$ Symbole und erwartungsgemäß maximal $K_3 - K_2 - K_1$ Grad-3-Codeworte für $R_3 - R_2 - R_1$ Symbole benötigt etc. Bei den ursprünglichen GC wurden diese K_i als Transitionspunkte gewählt und somit die Gradverteilung entsprechend $p_{Dist_1}(d)$ umgesetzt.

2) Perfect Source Setting: In [3] wird für die Analyse des GC Algorithmus von einem sogenannten 'perfect source setting' ausgegangen. Hierbei wird unter anderem angenommen, dass jedes Symbol mit gleicher Wahrscheinlichkeit in einem Codewort enthalten ist. Außerdem wird jedes Codewort von Grad d mit gleicher Wahrscheinlichkeit empfangen.

Im eigentlichen GC und MCGC Algorithmus werden die Symbole für ein Codewort jedoch aus der Menge der im Knoten bereits rekonstruierten Symbole, und nicht aus der Menge aller Symbole des Netzes, gewählt. Damit treten diese nicht mit gleicher Wahrscheinlichkeit in einem Codewort auf. Im Falle von statischen oder wenig dynamischen Netzen werden auch Codeworte von Grad d nicht mit gleicher Wahrscheinlichkeit empfangen. Aus diesem Grund stellt insbesondere Equation (4) nur eine Orientierung dar und dient als Vergleich mit einer idealisierten Annahme. Werden die Symbole, wie bei den GC, zufällig aus den bereits rekonstruierten Symbolen gewählt, so erhält man schlechtere Ergebnisse als in Equation (4) vorgegeben. Da bei den MCGC jedoch nicht zufällige Symbole aus der Menge aller Symbole, sondern gezielt die passenden Symbole für die Nachbarknoten ausgewählt werden, kann diese Formel bei den MCGC eingehalten und sogar unterschritten werden, obwohl nur aus der Menge der bereits rekonstruierten Symbolen gewählt wird.

D. Gewählte Gradverteilung

Laut Section III-A liegt die optimale Wartelistenlänge l bei $\frac{n-r}{2} \leq l \leq n-r$. Aus diesem Grund soll die Warteliste für $l < \frac{n-r}{2}$ anwachsen, weshalb sich die verwendete Gradverteilung bei MCGC zu Beginn an der Distanz-0-Minimierung orientiert. Am Anfang der Simulation ist entscheidend, dass möglichst keine redundanten Codeworte übertragen werden. Auf diese Weise sind zu jedem Zeitpunkt und mit jeder Anzahl im Knoten empfangener Codeworte möglichst viele Symbole in diesem bereits rekonstruiert. Da die Warteliste niemals mehr als $n-r$ Codeworte beinhalten sollte, cf.Section III-A, muss der Grad ab einer Wartelistenlänge von $l = \frac{n-r}{2}$ weniger stark ansteigen, damit die Warteliste wieder abgebaut wird. Dies wird durch die anschließende Verwendung der Distanz-1-Maximierung realisiert, da hierbei möglichst viele Codeworte mit Grad 1 ankommen, welche direkt dekodierbar sind. Diese Grad-1-Codeworte liefern ein neues rekonstruiertes Symbol, mit welchem die Codeworte aus der Warteliste ggf. dekodiert werden können. Somit sinkt die Länge der Warteliste wieder.

Die in [1] beschriebene, ursprüngliche MCGC Gradverteilung bestand aus drei verschiedenen Gradverteilungen mit Distanz-0-Minimierung. Nun wird jedoch nur zwischen zwei verschiedenen Gradverteilungen gewechselt. Die erste minimiert die Wahrscheinlichkeit Distanz-0-Codeworte zu empfangen, und die zweite maximiert die Wahrscheinlichkeit von Distanz-1-Codeworten. Dies führt zu besseren Simulationsergebnissen und lässt sich, durch nur eine Schranke bei der Distanz-0-Minimierung, einfacher an die Dynamik eines Netzes anpassen.

IV. Future Work

Aktuell arbeiten wir daran, die im System vorliegende Dynamik durch jeden Knoten selbst zu erkennen, um anschließend die verwendeten Parameter, der jeweiligen Dynamik entsprechend, optimal zu wählen. Der zuvor betrachtete, reaktive Ansatz wird folglich um einen proaktiven Ansatz erweitert und damit das Verfahren dynamisch an den Systemzustand angepasst. Zusätzlich wird untersucht, erhobene Daten nur noch lokal weiterzuverbreiten, statt anzustreben, dass jeder Knoten alle Daten des kompletten Netzes besitzt. Für die Zukunft sind neben umfassenden Simulationen auch Tests mit tatsächlichen Sensorknoten geplant.

Appendix

Im sogenannten Coupon Collector's Problem aus [8] wird z.B. der Fall betrachtet, dass bei gleicher Auftrittswahrscheinlichkeit der einzelnen Coupons k verschiedene Coupons zu sammeln sind. Fehlen noch m Coupons, d.h. wurden bereits $k-m$ Coupons gesammelt, sind erwartungsgemäß n_m Coupons nötig, um das im nächsten Moment fehlende $(k-m+1)$-te Coupon zu erhalten. Laut [8] gilt hierfür:

$$E(n_m) = k \cdot (\frac{1}{k} + \frac{1}{k-1} + \cdots + \frac{1}{m}). \tag{5}$$

Dies lässt sich umformen zu

$$E(n_m) = \sum_{i=0}^{k-m} \frac{k}{k-i}. \tag{6}$$

Das Coupon Collector's Problem sagt also aus, dass um r verschiedene Coupons zu erhalten, man bei gleichverteilter Auftrittswahrscheinlichkeit der Coupons erwartungsgemäß $\sum_{i=0}^{r-1} \frac{n}{n-i}$ Coupons sammeln muss.

References

[1] I. M. Grimm, M. van gen Hassend, and R. Kolla, "MCGC: Zuverlässige Datenhaltung in unzuverlässigen Netzen," in *Proceedings of the 15. GI/ITG KuVS Fachgespräch Sensornetze*, 2016, pp. 22–26.

[2] A. Kamra, V. Misra, J. Feldman, and D. Rubenstein, "Growth codes: Maximizing sensor network data persistence," in *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4. ACM, 2006, pp. 255–266.

[3] A. Kamra, J. Feldman, V. Misra, and D. Rubenstein, "Data persistence for zero-configuration sensor networks," in *ACM Special Interest Group on Data Communications (SIGCOMM)*, 2006.

[4] R. Ahlswede, N. Cai, S.-Y. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on information theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[5] L. Rizzo, "Effective erasure codes for reliable computer communication protocols," *ACM SIGCOMM computer communication review*, vol. 27, no. 2, pp. 24–36, 1997.

[6] T. Ho, R. Koetter, M. Medard, D. R. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," 2003.

[7] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.

[8] H. Von Schelling, "Coupon collecting for uneqal probabilities," *The American Mathematical Monthly*, vol. 61, no. 5, pp. 306–311, 1954.

Managing IoT device capabilities based on oneM2M ontology descriptions

Kristina Sahlmann
University of Potsdam, HTW Berlin
Potsdam, Germany
Email: sahlmann@uni-potsdam.de

Thomas Scheffler
Beuth-Hochschule für Technik
Berlin, Germany
Email: scheffler@beuth-hochschule.de

Bettina Schnor
University of Potsdam
Potsdam, Germany
Email: schnor@cs.uni-potsdam.de

***Abstract*—One of the challenges in the Internet of Things (IoT) is the heterogeneity of network devices. In order to facilitate automatized management when devices join and leave a network, a network manager should be notified about changes and given the ability to discover device capabilities. The NETCONF protocol, developed by the IETF, manages device configurations in the network. We investigate how this network management approach can be re-used to solve the interoperability problem in the IoT. We developed a mechanism for the semantic annotation of device configurations based on the oneM2M ontology. The realization of our approach is based on an existing NETCONF to MQTT bridge implementation. In this paper we discuss the results and optimization possibilities.**

I. Introduction

IoT systems may consist of many constrained devices [1], equipped with sensors, which can join and leave a network. There are billions of IoT devices expected[1] and an individual network may contain thousands of devices from many different vendors. The challenge is to reduce or even to avoid a human intervention for configuration of new devices in such a highly dynamic network.

Different standards for network management exist and are in active use. The Network Configuration Protocol (NETCONF) [2] is a new standard for IP-based networks defined by the IETF which provides mechanisms to install, manipulate, and delete the configuration of network devices. The NETCONF protocol uses the YANG [3] data modeling language to specify data models and protocol operations. The NETCONF protocol operations are realized as remote procedure calls (RPCs).

The Message Queue Telemetry Transport (MQTT) protocol is very common in the IoT domain[2]. The MQTT protocol was developed by IBM and later standardized by Oasis [4]. MQTT is a Machine-to-Machine (M2M) protocol and applies the Publish/Subscribe pattern for data dissemination in distributed systems. Messages reference so-called *Topics* of interests. Please refer to the reference [4] for more details on the MQTT protocol.

This paper describes an extension to an experiment by Scheffler and Bonneß [5]. It was their aim to manage dynamic IoT networks using a standards based approach. They developed a NETCONF bridge which translates between the IoT specific MQTT and the NETCONF protocol using a dynamically generated YANG model.

In contrast to the original experiment which used a proprietary format for the description of device capabilities, this paper investigates the usage of the oneM2M ontology for the description of device capabilities and configuration. While in the original experiment a LIFX LED bulb was controlled via WLAN, we use a microcontroller board with LEDs which is connected by 6LoWPAN to the network.

II. Deployment Setup

As shown in Figure 1, our system architecture consists of four components. The end node is a CC2538EM microcontroller board of the Development Kit[3] from Texas Instruments (TI). The TI-board is an ARM Cortex M3-based and supports IEEE 802.15.4 standard. Therefore, 6LoWPAN [6] is the communication protocol on the left side. The TI-Board is a very constrained device with 32 kiB RAM and 512 kiB Flash memory.
Further, there are the typical MQTT instances: a broker and two clients (the microcontroller board and the NETCONF server). The MQTT broker manages publication and subscription between the TI-Board and the NETCONF server. The NETCONF server is responsible for the network management. A user can control the TI-board via a NETCONF client.

The software implementation is based on existing Python libraries for MQTT, NETCONF and YANG, for details see [5]. The TI-Board uses Contiki OS[4] and its implementation of MQTT[5]. Our main software contribution is the device profile, described with the help of the oneM2M ontology, published in Step 1 (see Figure 1) and the ontology processing and mapping to the NETCONF configuration and YANG model in Step 3.

III. Ontology

A. What is an Ontology?

Basically, an ontology is a structured vocabulary. The vocabulary describes a certain domain of interest (e.g. metering, medicine, sensors). The ontology defines concepts and their

[1] http://spectrum.ieee.org/tech-talk/telecom/internet/popular-internet-of-things-forecast-of-50-billion-devices-by-2020-is-outdated
[2] https://heise.de/-3629650
[3] http://www.ti.com/tool/cc2538dk
[4] http://contiki-os.org/
[5] https://github.com/contiki-os/contiki/tree/master/apps/mqtt

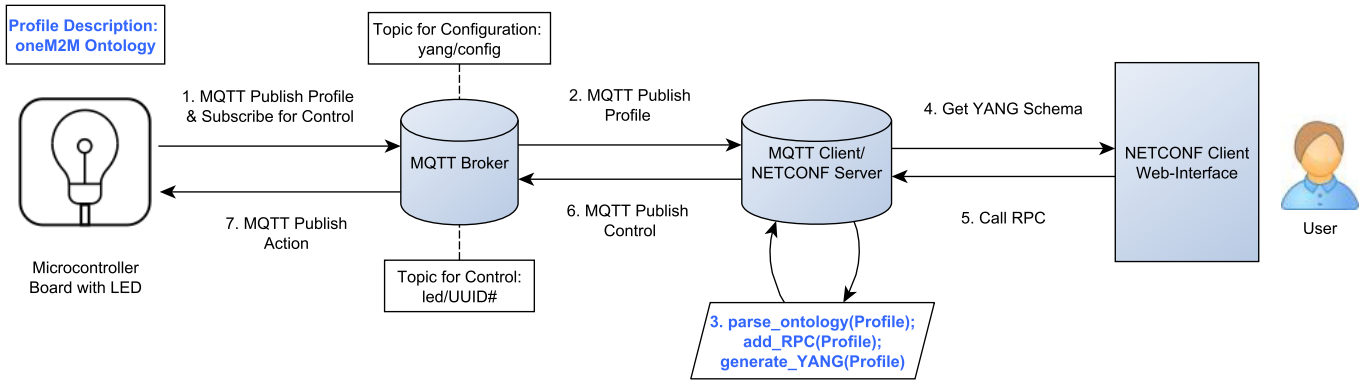


Fig. 1. System-Architecture with MQTT and NETCONF

relationships. *Concept* is a shared understanding of a term. Two concepts are connected by a directed relationship. Thus, they build so-called *triples*:

subject concept → ***relationship*** → ***object concept***

Concepts identify classes of individuals. An example for *classes* within IoT is:

Device → ***hasFunctionality*** → ***ControllingFunctionality***

An example for corresponding *individuals* is:

myDevice → ***hasFunctionality*** → ***changeLEDColor***

Additionally, there are *Data Properties* used, when the object concept is data (typed or untyped), for example:

Device → ***hasManufacturer*** → ***Literal***

Such ontologies, built on triples, represent formal models of knowledge areas.

B. Why Web Ontology Language (OWL)?

The W3C defined OWL [7] as a standard for ontology description. The OWL standard provides not only a standard for structuring vocabulary, but also has some main advantages:

- information is represented in a formal, machine-readable way;
- a reasoner can check consistency and infer new information;
- the W3C SPARQL [8] query language (i.e. SQL-similar notation) can be used for search on certain individuals using classes and relationships.
- two concepts in different ontologies and similar meaning can be mapped (i.e. using the property `owl:sameAs`)

Therefore, data described by an ontology helps to understand the domain of interest, can be processed in a structured way, and new facts can be inferred. Hence, the device capabilities can be described by an ontology defined for a certain domain.

In the presented approach, we make use of the first advantage to represent the description in a machine-readable way. The other advantages may play off in more complex scenarios with several devices and even several ontologies.

C. Ontologies for the IoT

Many ontologies are defined for different purposes even in the sensor and IoT domain (see for example [9], [10], [11]). The new version of the Semantic Sensor Network Ontology (SSN) [9] is defined by W3C and OGC as a vocabulary for sensors descriptions and their observations. W3C submitted an IoT-Lite Ontology[6] which represents IoT resources, entities and services and is an instantiation of the old SSN ontology[7]. Another example is the SensorML 2.0[8] from OGC which is not an ontology but a model language describing semantical meaning of sensors, actuators and processors. Janes proposed an IOTDB vocabulary based on JSON which describes the Things in the IoT (i.e. facets, purposes and units) [10]. Additionally, he provides a framework for developers which uses the underlying vocabulary to control devices. FIESTA-IoT Ontology[9] is proposed within the EU H2020 FIESTA-IoT project. This ontology aims to seamlessly support the federation of testbeds through the usage of semantic-based technologies. The M3-lite Taxonomy[10] is a lite version of M3 ontology, also designed for FIESTA-IOT H2020 EU project. This is a taxonomy for kinds of quantity (i.e. physical and environmental phenomenons), unit of measurements, different types of sensor and different types of domain of interests. Finally, the oneM2M organization develops standards for M2M and the IoT and proposed the oneM2M Base Ontology [11].

IV. How do we use the ontology?

The common recommendation is to reuse existing ontologies before defining a new one. In our case we want to describe the capabilities of the Things[11] rather than the Things themselves: send commands to the Thing and receive responses from the Thing. We have chosen the oneM2M Base

[6]https://www.w3.org/Submission/iot-lite/
[7]https://www.w3.org/2005/Incubator/ssn/ssnx/ssn
[8]http://www.opengeospatial.org/standards/sensorml
[9]http://ontology.fiesta-iot.eu/ontologyDocs/fiesta-iot/doc
[10]http://ontology.fiesta-iot.eu/ontologyDocs/fiesta-iot/doc
[11]The term Thing is used here as a device in terms of Internet of Things, and not as a class of an ontology

Ontology [11] (hereinafter referred to as oneM2M ontology) for two reasons: (i) it is a small ontology for service and functionality description of devices; and (ii) it is represented by the OWL standard.

The oneM2M ontology offers a vocabulary for the description of application services and device functionality. When all devices in a network use this ontology for describing themselves, a query can easily extract necessary information. We derive an ontology from the oneM2M Ontology to describe how the LED on the TI-Board can be controlled. There are two control methods: to turn on the light in a certain color and to switch the LED off. For simplicity, Figure 2 shows the part of our ontology which describes the switch-off control of the LED. Our device individual with the name `myDevice` is an instance of the oneM2M ontology class *Device*. In OWL, each resource (i. e. individual, class, property) is identified by an URI or IRI. In our example, the annotations in OWL are http://yang-netconf-mqtt#myDevice and http://www.onem2m.org/ontology/Base_Ontology/base_ontology#Device. The `myDevice` has a property `deviceUuid` which is an instance of the oneM2M class *ThingProperty*.

Further, the oneM2M Ontology distinguishes between machine interpretable exposures and human understandable meanings. A *Device* has a *Functionality* (*Controlling* or *Measuring*) which provides human understandable meaning what the device ”does”. A *Functionality* has a *Command* which is the human understandable name of an action that is invoked in a device. We name our functionality `switchOff` and the command `mqttCommandOff`.

On the other side, a *Device* provides a Service which is an electronic representation of a *Functionality* in a network. A *Service* is the machine interpretable exposure of *Functionality* and therefore it is discoverable, registerable and remotely controllable in the network. A *Service* has an *Operation* which is the machine interpretable exposure of a *Command* to the network. Our service individual is called `netconf` and the operation `offOperation`. *OperationInput* and *OperationOutput* of the *Operation* can parameterize the *Command*. This is the `input` individual which has the `deviceUuid` as an input data. Furthermore, we define two customized data properties for the operation: `mqttMethod` and `mqttTopic` with the pre-defined values. We use them for the NETCONF control over MQTT. We highlight the single steps of processing our ontology in Figure 1:

- Step 0, pre-configuration: create the device profile based on the ontology (e.g. using an ontology tool like Protégé[12]) ;
- Step 1: publish this device profile to the MQTT broker’s *Topic* ”yang/config” as payload ;
- Step 2: the MQTT client of the NETCONF server gets notified about this publication;
- Step 3: parse the device profile using ontology knowledge, add a RPC to the NETCONF server and generate a YANG model;

[12]http://protege.stanford.edu/

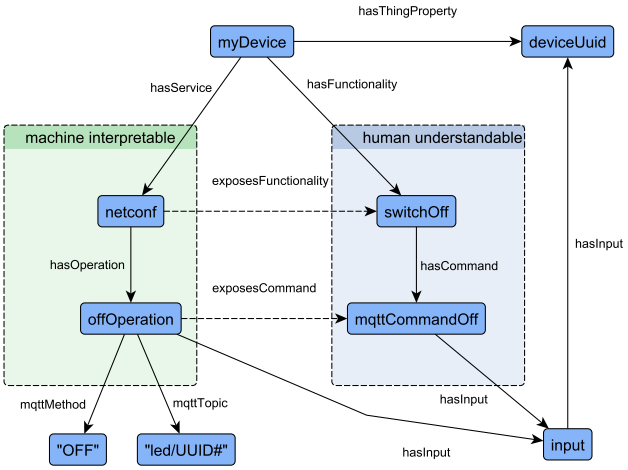


Fig. 2. Derived instances from the oneM2M Base Ontology

In the step 4, a user can obtain the YANG schema including all possible RPC calls over a web-interface. When the user trigger a certain RPC call in the step 5, this will be published to the MQTT broker in the step 6. The TI-Board will be notified because it is subscribed to the associated Topic.

V. Results and Discussion

The NETCONF bridge approach in [5] shows that the translation between the MQTT protocol and NETCONF is possible. This feature is particularly useful when many IoT devices must be managed together with other network devices. The IoT devices can describe their capabilities and disclose them to others on the network. NETCONF operates as a network manager and exposes an interface to the users.

We extended this approach by applying an ontology for the description of device capabilities. By using a semantic annotation, we achieved that this information is self-descriptive and machine interpretable. OWL processing tools (e.g. in Python) can be used for parsing and processing of ontology files.

The challenging part is the file size of the ontology for constrained devices. Every individual and class in OWL is addressed by an IRI (e.g. http://yang-netconf-mqtt#myDevice and http://www.onem2m.org/ontology/Base_Ontology/base_ontology#Device). Thus, the namespace causes redundancy. As a result, the file size grows.

We use the JSON-LD [12] format for our OWL file. The file size can be optimized by using compaction algorithms[12] for JSON-LD. These algorithms shorten IRIs and values as well as optimize the structure. Unfortunately, there is no existing tool which supports the full specification of compaction algorithms so far. We could reduce the file size from 6.66 KB to 4.44 KB by manually replacing namespaces with prefixes without loosing information. OWL files can be represented in different formats. Hence, we can reduce the file size by using the Turtle [13] format down to 3.5 KB because the syntax is more compact than JSON-LD.

The original work by [5] uses a proprietary format loosely inspired by YANG and is less than 1 KB in size. Therefore further optimization is necessary to approximate this size. One possibility to reduce the file size could be to omit some information, for example the human understandable description on the right side of Figure 2. An alternative could be the relocation of the file with device capabilities to a web server where it can be accessed by the NETCONF server. The device has to publish only an URL with a parameter for the device UUID. It may be a common occurrence that multiple devices share the same capabilities but have different UUIDs (e.g. individual IoT light-bulbs installed in a building). Therefore, the device UUID should replace the placeholder in the description file and returned to the NETCONF server.

VI. Related Work

Before the release of the oneM2M ontology in 2016, some research groups already highlighted the need for semantics in the oneM2M architecture.

Datta et al. presented in [14] a framework integrated into the oneM2M architecture for mobile devices. This implementation includes automatic discovery of the mobile devices, provisioning of sensors and IoT domains, semantic reasoning on sensor data and actuation based on the suggestions. They showed that semantic based computing is feasible on mobile devices and can be integrated into the oneM2M architecture. However, they used the early version of SenML[13] for the description of sensor measurements. This Markup Language is still an IETF draft. SenML has a compact syntax in JSON, XML and Efficient XML Interchange (EXI). Some parameters are defined, but no ontology is used.

Alaya et al. proposed in [15] an expressive ontology for the IoT called IoT-O. They argued that standards (i.e. among them oneM2M) are not interoperable at the semantic level and need an improvement. IoT-O claims to cover all IoT concepts and defines a set of ontologies for sensor, observation, actuator, actuation, and service models. These ontologies are connected by relationships. Furthermore, they propose a semantics extension to the oneM2M standard (i.e. the oneM2M Ontology was defined later). Additionally, they proposed a self-managing M2M system based on semantic configuration descriptions. Summarizing, they showed that the employment of ontology in the IoT can be utilized for several purposes. However, oneM2M recognized the need for semantics and defined the oneM2M Base ontology afterwards.

Li et al. proposed in [16] also an architecture for integrating semantics into the oneM2M architecture for the IoT. They also identify the need to integrate different ontologies for different domains and add an *ontology reference* field which for example may point to the URI of a weather ontology.

VII. Conclusion

This paper shows how to manage sensor devices in the IoT with conventional network management tools. Device capabilities are described with the help of the oneM2M ontology and published via a MQTT broker to the network. NETCONF RPCs can be dynamically deduced using ontology knowledge. We have shown that this approach is feasible, but further optimizations to reduce the ontology file size are required.

In the future work, we want to compare the presented approach with other methods for connecting devices in the IoT using semantics (e.g. with the self-organizing protocol MQTT-SN and the data-centric protocol ASIP [17]). Overall, the application of the ontology for constrained devices is in the early stages of development and has many potentials for further research.

[13]https://tools.ietf.org/html/draft-jennings-core-senml-00

Acknowledgment

This research is funded by the Federal Ministry of Education and Research under grant number Professors Program II. The responsibility for the content of this publication lies with the authors.

References

[1] "RFC 7228 - Terminology for Constrained-Node Networks," 2014. [Online]. Available: http://tools.ietf.org/html/rfc7228

[2] "RFC 6241 - Network Configuration Protocol (NETCONF)," 2011. [Online]. Available: https://tools.ietf.org/pdf/rfc6241.pdf

[3] "RFC 6020 - YANG - A data modeling language for NETCONF," 2010. [Online]. Available: https://tools.ietf.org/pdf/rfc6020.pdf

[4] "MQTT Version 3.1.1," 2014. [Online]. Available: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html

[5] T. Scheffler and O. Bonneß, "Manage resource-constrained IoT devices through dynamically generated and deployed YANG models," in *Applied Networking Research Workshop (ANRW) 2017*. IETF, 2017.

[6] "RFC 4944 - Transmission of IPv6 Packets over IEEE 802.15.4 Networks," 2007. [Online]. Available: http://tools.ietf.org/html/rfc4944

[7] "OWL 2 Web Ontology Language Document Overview (Second Edition)," 11 December 2012. [Online]. Available: https://www.w3.org/TR/owl2-overview/

[8] "SPARQL 1.1 Query Language," 21 March 2013. [Online]. Available: https://www.w3.org/TR/sparql11-query/

[9] "Semantic Sensor Network Ontology," 04 May 2017. [Online]. Available: https://www.w3.org/TR/vocab-ssn/

[10] D. P. Janes, "Semantic Metastandards Will Unlock IoT Interoperability," in *Internet of things. IoT infrastructures*, ser. Lecture notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Switzerland: Springer, 2016, vol. 170, pp. 396–402.

[11] "Base Ontology: oneM2M Technical Specification: TS-0012-V2.0.0," 30 August 2016. [Online]. Available: http://www.onem2m.org/technical/published-documents

[12] "JSON-LD 1.0, A JSON-based Serialization for Linked Data," 16 January 2014. [Online]. Available: https://www.w3.org/TR/json-ld/

[13] "RDF 1.1 Turtle, Terse RDF Triple Language," 25 February 2014. [Online]. Available: https://www.w3.org/TR/turtle/

[14] S. K. Datta, A. Gyrard, C. Bonnet, and K. Boudaoud, "oneM2M architecture based user centric IoT application development," in *Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on*. IEEE, 2015, pp. 100–107.

[15] M. B. Alaya, S. Medjiah, T. Monteil, and K. Drira, "Toward semantic interoperability in oneM2M architecture," *IEEE Communications Magazine*, vol. 53, no. 12, pp. 35–41, 2015.

[16] H. Li, D. Seed, B. Flynn, C. Mladin, and R. Di Girolamo, "Enabling Semantics in an M2M/IoT Service Delivery Platform," in *2016 IEEE Tenth International Conference on Semantic Computing*. Piscataway, NJ: IEEE, 2016, pp. 206–213.

[17] K. Sahlmann, T. Schwotzer, and B. Schnor, "The Ad hoc Semantic Internet Protocol (ASIP) for Constrained Devices," in *Proceedings of the 15. GI/ITG KuVS Fachgespräch Sensornetze*, ser. Technical Reports, Augsburg, September 2016, pp. 49–52.

Ein Leistungsmesssystem für verteilte Sensorknoten zur Unterstützung bei der Implementierung von Protokollen in WSN

Max Frohberg, Mario Schölzel
IHP
Im Technologiepark 25
Frankfurt (Oder), Germany
Email: {frohberg, schoelzel}@ihp-microelectronics.com

***Zusammenfassung*—Die Minimierung des Energieverbrauchs in batteriebetriebenen drahtlosen Sensornetzwerken (WSN) ist eine der größten Herausforderungen und erfordert eine Bewertung bereits während der Entwicklungsphase. Diese Abschätzung kann in frühen Entwurfsphasen durch Simulationen und Modelle erfolgen. Ein abschließender Test unter realen Umgebungsbedingungen kann diese Annahmen überprüfen. In diesem Beitrag wird die WISDOM-Plattform vorgestellt, die selbst als Sensorknoten fungiert und mit einem 24 bit AD-Wandler zur Messung der Leistung von Sensorknoten in einem WSN ausgestattet ist. Messungen können drahtlos an eine Senke übermittelt und ein Verbrauchsprofil für ein großflächiges WSN unter realistischen Lastbedingungen erstellt werden. Die WISDOM-Plattform bietet einen sehr hohen dynamischen Messbereich, der für die Überwachung von Sensorknoten, die häufig zwischen stromsparendem und aktivem Modus wechseln, wichtig ist.**

I. Einleitung

Die Ressourcenlimitierung von Sensorknoten (Knoten) in einem *drahtlosen Sensornetzwerk (WSN)* hinsichtlich Herstellungskosten, Größe, Rechenleistung und Stromverbrauch wurden bereits in einer Vielzahl von Arbeiten wie in [1], [2] und [3] analysiert und diskutiert. Einen Knoten in verteilten und schlecht erreichbaren Sensornetzen möglichst lange oder durch Energy-Harvesting-Methoden sogar dauerhaft betreiben zu können, steht dabei an erster Stelle. Die Lebenszeit eines Knotens wird durch verschiedene Faktoren beeinflusst. Einen Großteil der Energie wendet ein Knoten in der Regel für die drahtlose Kommunikation innerhalb eines Netzwerkes auf. Auch der Verbrauch im Sleep-Modus beeinflusst das Energiebudget. Die Implementierung von stromsparenden Kommunikationsprotokollen und Anwendungen hat daher großen Einfluss auf die Lebensdauer, Verfügbarkeit und die gesamte Netzwerkstabilität. Zur Evaluierung einer Anwendung hinsichtlich Energieeffizienz gibt es integrierte Software-Tools und Frameworks wie *TI's MSP Energy-Trace* oder Simulatoren wie *PowerTOSSIM*, die durch Verwendung von Modellen und unter Annahme von Nutzungsparametern, eine Abschätzung des Energieverbrauchs eines Knotens treffen können. Auch durch eine Evaluierung, meist unter Laborbedingungen, kann die Zuverlässigkeit eines WSN erhöht und die Laufzeit eines Knotens berechnet werden. Deren Ergebnisse können von denen in der realen Anwendung abweichen. Gründe dafür können zum Beispiel Designfehler, Seiteneffekte oder äußere Störeinflüsse sein, die beim Testen nie aufgefallen sind oder nicht bzw. nur unzureichend berücksichtigt wurden. Daher ist ein abschließender Test unter realen Bedingungen und möglichst im Einsatzgebiet erforderlich. Eine präzise Energiemessung kann den Entwickler dabei unterstützen, den Energieverbrauch des gesamten Knotens unter realen Bedingungen zu ermitteln, fehlerhaftes Verhalten zu entdecken und zu beseitigen. Dabei sollte der beobachtete Knoten nicht oder zumindest nur geringfügig beeinflusst werden.

Kabelgebundene Testbeds, basierend auf Server-, LAN- und USB-Infrastrukturen, wie *Twist* von der TU Berlin [4] oder sogar Cross-Plattformen darüber wie im *Wisebed-Projekt* [5], bieten ein hohes Maß an Testabdeckungen und ermöglichen den automatischen Betrieb und die statistische Auswertung von Testabläufen. Aufgrund ihrer Architektur sind sie für stationäre WSN-Anwendungen geeignet. Für mobile Anwendungen können einzelne Hardware-Module wie zum Beispiel aus dem *Nemo-* [6] oder *SEM-Projekt* [7] verwendet werden, die ihre Messergebnisse über einfache Interfaces direkt an das *Device Under Test (DUT)* zurückgegeben. Das Nemo-Modul misst beispielsweise den Strom eines DUT und leitet die Ergebnisse über einen einzelnen Pin als direktes Feedback an das DUT zurück. Die Messung des Stroms alleine reicht nicht für eine präzise Verbrauchsanalyse aus. Leistungsspitzen können die Spannung von batteriebetriebenen Geräten aufgrund ihres Innenwiederstandes stark beeinflussen. Zudem sinkt die Spannung batteriebetriebener Geräte mit abnehmender Ladung. Das Modul aus dem SEM-Projekt greift dieses Problem auf, misst zusätzlich die Spannung und überträgt die Ergebnisse durch Frequenzmodulation über einen Pin an das DUT. Die Demodulation durch das DUT ist ggf. recht aufwändig. Die Messwerte können entweder an ein stationäres Host-System oder per Funk über das Radio-Modul des DUT versendet werden. Sowohl eine Demodulation, als auch die Nutzung des Radio-Moduls, beeinflussen das DUT durch zusätzliche Rechenleistung und Kommunikationsaufwand in seinem Verhalten.

In diesem Paper wird die WISDOM-Plattform zur genauen Leistungserfassung von Sensorknoten vorgestellt, deren Ein-

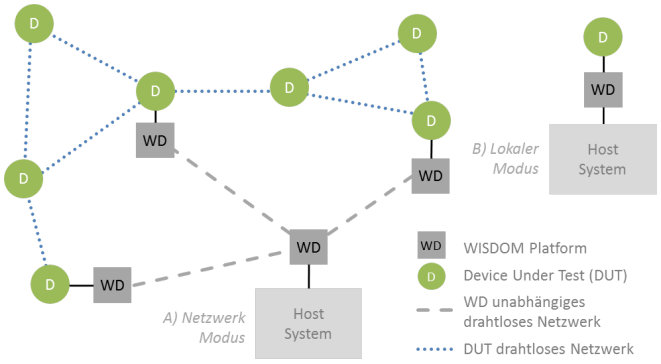


Abbildung 1. WISDOM in einer WSN-Anwendung unter realen Umgebungsbedingungen (A), Nutzung zur Evaluation einzelner Knoten im Entwicklungsprozess (B).

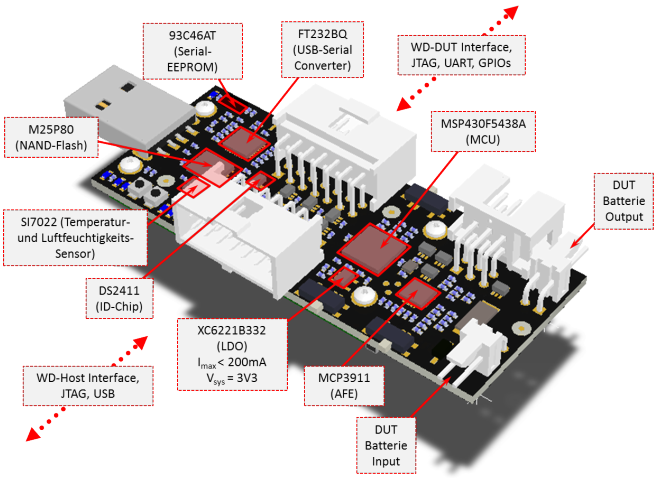


Abbildung 2. WISDOM-Plattform-Architektur. Auf der linken Seite die USB- und JTAG-Schnittstelle für die Kommunikation mit dem Host-System, auf der rechten Seite die Schnittstelle zur Kommunikation und Messung des DUT.

satz sowohl für lokale Messungen während der Entwicklungsphase, als auch für verteilte Messungen in einem operativen WSN konzipiert ist. Das Paper ist wie folgt gegliedert. In Abschnitt II wird das allgemeine Konzept und die Plattform vorgestellt. In Abschnitt III wird die Plattform hinsichtlich Genauigkeit und Verbrauch evaluiert.

II. Konzept

Das *Wireless Debugging and Power Measurement System* (WISDOM) wurde entwickelt, um die Leistungsaufnahme von kleinen elektronischen Geräten, insbesondere Sensorknoten in WSNs, für Analysezwecke zu erfassen. Ein *WISDOM Device* (WD) wird zwischen DUT und dessen Stromversorgung geschaltet. Somit können Strom und Spannung eines DUT mit hoher Abtastrate gemessen und an ein zentrales Host-System übertragen werden. Das Konzept, wie in Abbildung 1 dargestellt, sieht sowohl eine lokale Verwendung während der Entwicklungszeit, als auch in einem Netzwerk während eines abschließenden Tests unter realen Bedingungen vor.

A. Plattform

Ausgestattet mit Sensoren zur Leistungserfassung und zur Messung von Luftfeuchtigkeit und Temperatur, handelt es sich bei der WISDOM-Plattform, wie in Abbildung 2 dargestellt, selbst um einen Sensorknoten. Die Plattform wird zwischen die Stromversorgung des DUT und das DUT selbst gesteckt und ist für die Untersuchung von batteriebetriebenen Sensorknoten ausgelegt. Der Messbereich für den Strom ist auf $\pm 150\,mA$ begrenzt. Somit sind sowohl Lade- als auch Entladezyklen und Energy-Harvesting-Maßnahmen eines DUT evaluierbar. Ein WD kann entweder über USB oder die Batterie des DUT betrieben werden. Im USB-Betrieb ist die Spannung des DUT unabhängig vom WD und kann zwischen $0\,V$ und $4,25\,V$ liegen. Wird die Batterie des DUT für die Versorgung des WD verwendet, so muss die Spannung zwischen $3,4\,V$ und $4,25\,V$ liegen. In beiden Fällen ist der operative Bereich von 1-Zellen Lithium-Polymer-Akkumulatoren (LiPo) abgedeckt. Strom und Spannung des DUT werden zeitgleich mit dem MCP3911 $24\,Bit$ (circa $15\,Bit$ ENOB lt. Datenblatt) Analog Front End (AFE) über den Spannungsabfall eines $0,5\,\Omega$ Shunt-Widerstands, bzw. eines Spannungsteilers, mit einer Abtastrate von $31,25\,kHz$ gemessen. Das AFE ermöglicht theoretisch eine maximale Abtastrate von bis zu $83,3\,kHz$. Da der Stromverbrauch eines DUT stark schwanken kann, wird ein intern konfigurierbarer Vorverstärker verwendet, um sehr kleine Spannungen im Bereich von wenigen mV zu messen.

Temperatur und relative Luftfeuchtigkeit können je nach Lage eines DUT stark schwanken und dessen Verhalten beeinflussen. Daher werden solche Metadaten auf jedem WD erfasst. Messungen werden entweder über USB oder ein zusätzliches High-Power Radio-Modul an ein Host-System gesendet. Das Radio-Modul kann je nach Frequenz des DUT-Radios durch ein pinkompatibles mit unterschiedlichen Frequenzen ausgetauscht werden, um Störungen in der Kommunikation der Zielanwendung zu verhindern. Zur direkten Übertragung an ein Host-System wird ein USB-Seriell-Wandler verwendet.

Für Messdaten mit höchster Abtastrate die nicht schnell genug übertragen werden können, ist auf jedem WD ein $1\,MB$ Flash und ein SD-Karten-Slot vorgesehen. Bei einer Abtastrate von $31,25\,kHz$ können zum Beispiel alle Messwerte für einen Zeitraum von bis zu 3 Sekunden gespeichert und zu einem späteren Zeitpunkt abgerufen werden. Je nach verwendeter SD-Karte erweitert sich der mögliche Aufzeichnungszeitraum um ein Vielfaches.

Als optionale Schnittstellen zwischen WD und DUT stehen ein UART und zwei GPIO-Pins zur Verfügung, mit denen zusätzliche Informationen wie Marker aufgezeichnet werden können. Somit wird eine zustands- und funktionsgenaue Zuordnung des Stromverbrauchs ermöglicht. Ein DUT kann die GPIOs, in einer atomaren Operation beim Betreten einer zu untersuchenden Funktion oder eines Systemzustandes auf 1 und beim Verlassen wieder auf 0 setzen. Ein GPIO kann

auch als Auslöser (Capture-Pin) verwendet werden, um eine detaillierte Speicherung zu veranlassen.

Die Firmware der WISDOM-Plattform wurde unter Verwendung des Betriebssystems *langOS* [8] implementiert. Die Einstellung des Vorverstärkers wird dabei in einem zusätzlichen Treiber vorgenommen, nach jedem Messwert automatisch bestimmt und ggf. konfiguriert.

B. Lokaler Betrieb

Für den lokalen Betrieb wird ein WD direkt über USB an ein Host-System angeschlossen. Der Modus ist für detaillierte Analysen des Stromverbrauchs konzipiert. Die Verwendung des Radio-Moduls ist nicht notwendig. Daten werden direkt über USB übertragen. Die Übertragungsgeschwindigkeit ist durch die Verarbeitung der Daten und den Seriell-USB-Wandler begrenzt. Daher wird ein Durchschnitt der Messwerte gebildet und dieser mit $1\,kSps$ übertragen. Zusätzlich zu Strom und Spannung, werden Temperatur und relative Luftfeuchtigkeit alle 5 Sekunden an das Host-System übermittelt.

C. Netzwerkbetrieb

Für einen abschließenden Test unter realen Bedingungen kann das System Messwerte über ein eigenständiges Netzwerk an eine zentrale Senke schicken. Das Netzwerk ist in einer Stern-Topologie organisiert, wobei ein WD die Rolle der Senke einnimmt und alle eintreffenden Daten direkt über USB an ein Host-System überträgt. Bis zu 252 WDs können unter Verwendung eines einfachen TDMA-Protokolls Daten an diese Senke schicken. Der Detailgrad der Messungen ist aufgrund der Anzahl der Kommunikationsteilnehmer reduziert und bildet einen Durchschnitt der Messwerte seit der letzten Übertragung ab. Eine Anfrage zur Übertragung der Messwerte aller WDs wird in einer Endlosschleife durch die Senke initiiert und dient gleichzeitig zur Zeitsynchronisation. WDs haben eine voreingestellte Adresse und können ihren Zeitschlitz zur Datenübertragung auf deren Grundlage berechnen. Bis zu diesem Zeitpunkt und auch nach der Übertragung wird das Radio-Modul in den Stromsparmodus versetzt und erst wieder aktiviert, wenn die nächste Anfrage erwartet wird.

D. Host-System

Alle eingehenden Daten der Senke werden im Netzwerkbetrieb zusammen mit einem Zeitstempel in einer Log-Datei gespeichert und können später ausgewertet werden. Für die Evaluation eines einzelnen DUT steht, wie in Abbildung 3 dargestellt, eine grafische Nutzeroberfläche zur Verfügung in der alle Daten in Echtzeit visualisiert werden.

III. Evaluierung

Das System wurde unter den Aspekten Eigenverbrauch und Messgenauigkeit evaluiert. Zwei unterschiedliche Vertreter von Sensorknoten wurden als Beispielanwendung herangezogen, der Stromverbrauch und die Laufzeit des jeweiligen DUTs untersucht und die Messungen mit professionellem Equipment verglichen.

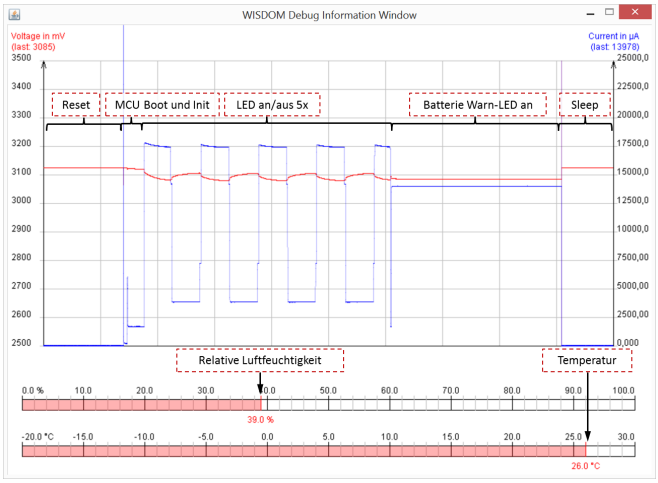


Abbildung 3. Grafische Nutzeroberfläche für die Visualisierung der Messwerte in Echtzeit (Bsp. LEO-Smartkey).

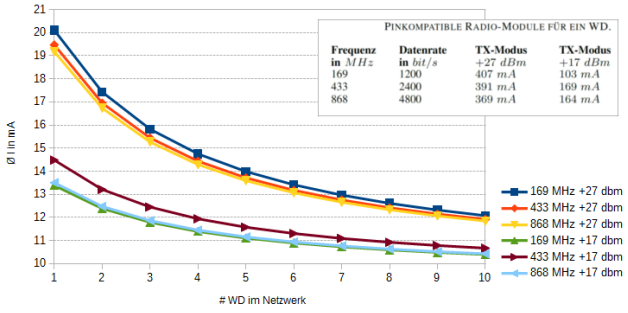


Abbildung 4. Gesamtstromverbrauch eines WD bei $+27\,dBm$ (max) und $+17\,dBm$ (min), in Abhängigkeit vom Radio-Modul und der Anzahl der Kommunikationsteilnehmer im Netzwerk.

A. Leistungsaufnahme

Im lokalen Betrieb über USB konnte für die WISDOM-Plattform ein Strom von $9,4\,mA$ gemessen werden. Für den Netzwerkbetrieb wurde der Verbrauch aufgrund von Datenblattangaben ermittelt. Er hängt von der Anzahl der Kommunikationsteilnehmer im Netzwerk ab, da diese nur in ihrem Zeitschlitz Daten senden und ihr Radio-Modul ansonsten in den Sleep-Modus versetzen. Zudem beeinflusst die Wahl des Radio-Moduls, dessen Datenrate und die eingestellte Sendeleistung den Gesamtverbrauch eines WD. Die Konfigurationen und Ergebnisse sind in Abbildung 4 dargestellt. Bei der Verwendung einer $2200\,mAh$ Batterie, einem Netzwerk von 10 WDs und einem durchschnittlichen Verbrauch des DUT von $3\,mA$ kann das System bis zu $6,7\,Tagen$ operieren.

B. Genauigkeit der Messungen

Zur Messung der Spannung wurde ein $1\,k\Omega$ Leistungswiderstand als Verbraucher und ein Labornetzteil zur Stromversorgung angeschlossen. Die Spannung wurde sukzessive reduziert und die Messergebnisse mit denen des Referenzgerätes *Agilent MSO7104B* verglichen. Bei den Ergebnissen in Abbildung

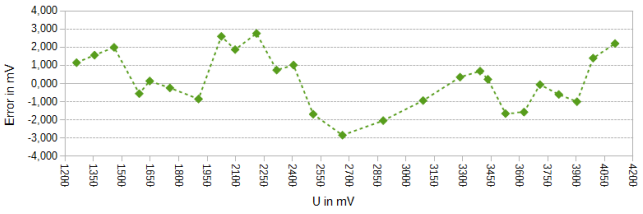


Abbildung 5. Messung der Genauigkeit über einen dynamischen Spannungsbereich, verglichen mit dem Referenzgerät *Agilent MSO7104B* (entspricht der Bezugsgröße von 100%).

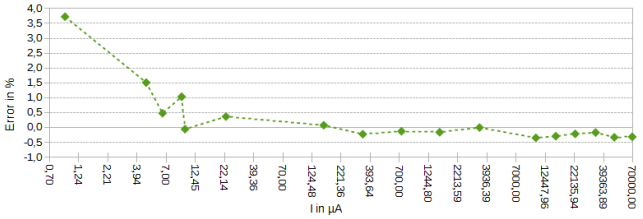


Abbildung 6. Messung der Genauigkeit mit dynamischem Lastbereich bei konstanten $3,3\,V$, verglichen mit dem Referenzgerät *Keithley 2636*.

5 liegt der Fehler im Messbereich zwischen $4250\,mV$ und $1200\,mV$ bei $\pm 3\,mV$ ($\pm 0,14\,\%$).

Der Strom wurde bei einer konstanten Spannung von $3,3\,V$, unter Verwendung verschiedener Leistungswiderstände, gemessen und mit dem Referenzgerät *Keithley 2636* verglichen. Die Messung kann zum Vergleich bei konstanten Strömen herangezogen werden und erzielte die in Abbildung 6 dargestellten Ergebnisse mit einer Abweichung von $\pm 0,5\,\%$ im Bereich zwischen $10\,\mu A$ und $70\,mA$ und bis zu $+4\,\%$ im Bereich zwischen $1\,\mu A$ und $10\,\mu A$.

Der Strom verändert sich im Vergleich zur Spannung bei batteriebetriebenen Geräten in der Regel um ein Vielfaches schneller. Strom und Spannung werden daher bereits im AFE für die Dauer der Messung integriert und ein Durchschnittswert im *Mikrocontroller (MCU)* zwischen zwei Übertragungen gebildet. Die Genauigkeit von solchen dynamischen Veränderungen wurde am Beispiel der *Fitbit Surge* Fitnessuhr mit integriertem GPS und Pulssensor unter Verwendung des gleichen Referenzgeräts evaluiert. Der Messvorgang und die Ausgangsbedingungen für das DUT waren für beide Messungen identisch. Es wurde gewartet, bis das DUT ein GPS-Signal empfangen und einen stabilen Systemzustand erreicht hat. Anschließend wurde ein Durchschnitt über den Messzeitraum von $60\,s$ gebildet und anschließend verglichen. Ein Ausschnitt dieser Messungen zum Vergleich ist in Abbildung 7 zu sehen. Mit dem Referenzgerät wurden durchschnittliche $18,543\,mA$ ermittelt. Die WISDOM-Plattform hat für den gleichen Zeitraum einen Durchschnitt von $18,458\,mA$ gemessen und weist eine Abweichung von $0,46\,\%$ auf.

IV. Zusammenfassung und Ausblick

In diesem Paper wurde die WISDOM-Plattform zur Evaluierung des Stromverbrauchs von Sensorknoten vorgestellt. WIS-

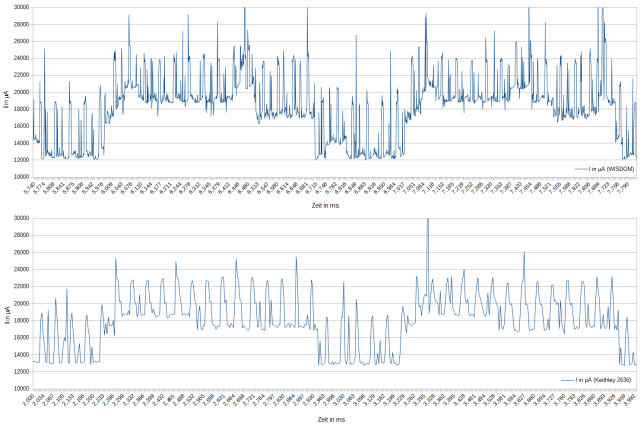


Abbildung 7. Ausschnitt der Messung ($2\,Sek.$), GPS aktiviert, Batteriespannung $4050\,mV$, WISDOM (oben), *Keithley 2636* (unten).

DOM eignet sich zur Messung angeschlossener Verbraucher sowohl während der Entwicklungsphase, als auch für einen Test einzelner Knoten in einem WSN unter realen Bedingungen. Dabei weist WISDOM einen hoch dynamischen Bereich von $1 : 150000$ bei einem Fehler von $-0,5\,\%$ bis $+4\,\%$ im Vergleich zu Referenzmessungen auf. Die Implementierung von zusätzlichen Funktionen und Verbesserungen, wie lokales Speichern von Daten, eine optimierte Datenübertragung und die Programmierbarkeit eines DUT über JTAG im Feld, sind bereits angedacht. Eine automatische Kalibrierung und Temperaturkompensation steht noch aus.

V. Danksagung

Diese Arbeit ist Teil des DIAMANT-Projekts am IHP und wurde durch das *Bundesministerium für Bildung und Forschung* unter der Referenznummer *03IPT601X* gefördert.

Literatur

[1] M. W. V. J. Hodge, S. O'Keefe and A. Moulds, "Wireless sensor networks for condition monitoring in the railway industry: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, pp. 1088–1106, June 2015.

[2] P. D. Marco, "Protocol design and implementation for wireless sensor networks," Master's thesis, KTH Royal Institute of Technology, Stockholm, Sweden, 2008.

[3] H. C. A. Davis, "A survey of wireless sensor network architectures," *International Journal of Computer Science and Engineering Survey*, vol. 3, December 2012.

[4] V. Handziski, A. Köpke, A. Willig, and A. Wolisz, "Twist: A scalable and reconfigurable testbed for wireless indoor experiments with sensor networks," pp. 63–70, 2006.

[5] G. Coulson, B. Porter, I. Chatzigiannakis, C. Koninis, S. Fischer, D. Pfisterer, D. Bimschas, T. Braun, P. Hurni, M. Anwander, G. Wagenknecht, S. P. Fekete, A. Kröller, and T. Baumgartner, "Flexible experimentation in wireless sensor networks," *Commun. ACM*, vol. 55, pp. 82–90, Jan. 2012.

[6] R. Zhou and G. Xing, "Nemo: A high-fidelity noninvasive power meter system for wireless sensor networks," pp. 141–152, April 2013.

[7] J. V. C. Fernandez, D. Bouvier, "Low-power self-energy meter for wireless sensor network," *IEEE International Conference on Distributed Computing in Sensor Systems*, pp. 315–317, May 2013.

[8] A. K. Oliver Stecklina and S. Kornemann, "langos - a low power application-specific cofigurable operating system," in *13. GI / ITG Fachgespräch Sensornetze*, pp. 15–18, Institut für Informatik und Computational Science, 9 2014.

TUCap: A Sensing System to Capture and Process Appliance Power Consumption in Smart Spaces

Andreas Reinhardt
Technische Universität Clausthal
Clausthal-Zellerfeld, Germany
reinhardt@ieee.org

***Abstract*—The collection of power consumption data from electrical appliances is a key enabling element for grid-related services such as load forecasting or anomalous consumption pattern detection. Device-level sensors (*smart plugs*) have found widespread use to collect such data. However, they prevalently report an electrical appliance's power consumption at a rate of one reading per second. With mains voltage frequencies of 50/60 Hz, undersampling and the consequent loss of spectral information result from the use of such sensor devices. Moreover, most smart plugs only report readings of an appliance's real power consumption. Important supplementary features like the phase shift between voltage and current or the magnitude of reactive power are not available for retrieval from commercially available devices. In this work, we overcome these limitations of smart plugs by presenting TUCap, an embedded sensing system to capture appliances' electrical load signatures in minute detail. Our design caters to the provision of a high information content by capturing voltage and current waveforms at a sampling rate of 36 ksps. Thus, spectral components are implicitly included in collected traces. Moreover, TUCap uses local data processing routines to detect and eliminate redundancies. Thus, a high data fidelity is maintained while achieving significant reductions of network traffic. All functionalities are implemented in a proof-of-concept system design and evaluated in practice.**

I. Introduction

Since the publication of George Hart's seminal work on non-intrusive load monitoring (NILM) in 1985 [1], research activities to extract and utilize the information content in electricity consumption data have seen an almost exponential increase. A key insight of *energy analytics* is that power consumption patterns are suitable indicators to infer information on both appliance and user activities in residential, commercial, and industrial settings. Many data processing algorithms have consequently been presented to analyze energy consumption data, e.g., to characterize household types [2, 3], infer building occupancy [4], or predict future electricity consumption [5, 6]. Their designs often also highlight one of the principal limitations of NILM [7, 8]: It relies on the usage of a single measurement point (usually a smart electricity meter). Thus, differentiating between appliances of the same type operated in different locations within a dwelling is complicated. In order to overcome this limitation, deployment strategies for sensors to capture electricity consumption in a more fine-grained fashion have emerged, e.g., by adding sensors for each electrical circuit in a home [9, 10] or even inserting appliance-level sensors (*smart plugs*) into the power cords of all relevant devices [11].

Supplementary to the option of increasing energy analytics accuracy through deploying more sensing points in a dwelling, the data processing capabilities are also strongly dependent on the sampling rate at which readings have been collected [12]. Unfortunately, the majority of today's commercially available plug-level sensors suffer from shortcomings in this regard. Smart plugs often report values only once per second or even less often, thus transients and spikes of shorter duration commonly remain undetected. Furthermore, appliances with inductive or capacitive components incur a reactive power consumption, whose detection requires the synchronous sampling of voltage and current. As most smart plugs are not fitted out with voltage sensors, they are unable to differentiate between real and reactive power consumption. At last, mains voltage and current consumption waveforms rarely resemble perfect sinusoids. In contrast, slight distortions are ubiquitous due to the wide presence of non-linear loads, e.g., switch-mode power supplies. To fully capture these characteristics, sampling frequencies have to satisfy the Nyquist sampling theorem [13], i.e., be greater than twice the highest distorting frequency. This is not the case for smart plugs that report readings at 1 Hz.

The limitations of commercially available smart plugs have motivated many researchers to design platforms to measure electrical energy consumption, such as Plug [14], ACme [15], SEM [16], SmartMeter.KOM [17], WCSN [18], or YoMo [19]. While these devices are capable of sampling appliance power consumptions at higher resolutions than their commercially available counterparts, their data processing functionalities are often limited. In fact, research platforms either forward data to a collection device without any prior processing (i.e., they report data at the native sampling rate), or apply lossy data processing algorithms to return characteristic values (e.g., RMS current, crest factor, etc) at the 1 Hz interval prevailing among commercial platforms. The former approach, however, results in an enormous bandwidth requirement, whereas the latter solution disallows for the detection of short-term fluctuations. Such features are, however, vital for energy analytics.

The unavailability of a solution to capture high-resolution data while minimizing the bandwidth demand during periods of constant power consumption has motivated us to create TUCap. We present an overview of its architecture and the hard- and software design decisions taken in Sec. II. Subsequently, we discuss considerations for the system's practical use in Sec. III. At last, we conclude this paper in Sec. IV.

II. System Overview

In order to enable energy analytics, both mains voltage and current flowing into an electrical appliance must be sampled at a high temporal and amplitudinal resolution. Therefore, three design decisions need to be made: Firstly, suitable transducers to translate voltage and current into analog signals need to be carefully selected. Secondly, an analog-to-digital conversion (*ADC*) step with high resolution must be part of the system in order to discretize samples and thereby enable their processing. Thirdly, a networked embedded system with the capabilities for local data processing is required in order to accomplish our objective of filtering out redundant readings while maintaining full data fidelity for when changes occur. We discuss our design choices for each of these aspects as follows.

A. Transducer Selection

The *load signature* of an electrical appliance, i.e., its characteristic power consumption during the course of its operation, is defined by the current flowing through the device as well as the voltage across its terminals.

TUCap samples current flows via a non-invasive AC current transformer that can be clipped on any conductor. The selected current transformer (model SCT-013) has a 2000:1 ratio, thus it transforms a primary current of 2 A into a secondary current of 1 mA. We burden the current transformer with a series of load resistors configured as a voltage divider that delivers an output voltage of ± 1 V for a ± 16 A primary current. The ± 1 V range has been chosen in order to be compliant with the input specifications of most ADC devices (cf. Sec. II-B). The current transformer has a specified frequency response up to 150 kHz, thus harmonics will be reflected in its output. As recommended per the data sheet [20], a passive first-order low-pass filter has been added on-board TUCap to avoid aliasing.

To capture the mains voltage signal, a 230 V/6 V transformer is present on TUCap. The transformer's output is burdened with a resistor network to proportionally scale voltages up to ± 433 V down to the ± 1 V input range of the ADC. Like on the current channel, a first-order anti-aliasing filter has been added to the board. The key intention behind using a transformer is to ensure galvanic decoupling, and thus render the use of a signal isolation device for the power and data connections unnecessary. A downside of using a transformer, however, is its non-linearity: Small transformers often introduce a phase shift that needs to be compensated for. We have hence measured the transformer's voltage transfer characteristics in Fig. 2, and determined the output voltage to lead (i.e., advance) the input voltage by 500μs. A compensation for this phase shift thus has to be implemented in software.

The connection points for the chosen voltage and current transducers in an appliance's power cord are shown in Fig. 1.

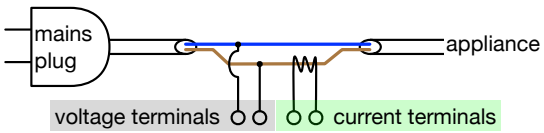


Fig. 1. Transducer connection terminals in an appliance's power cord.

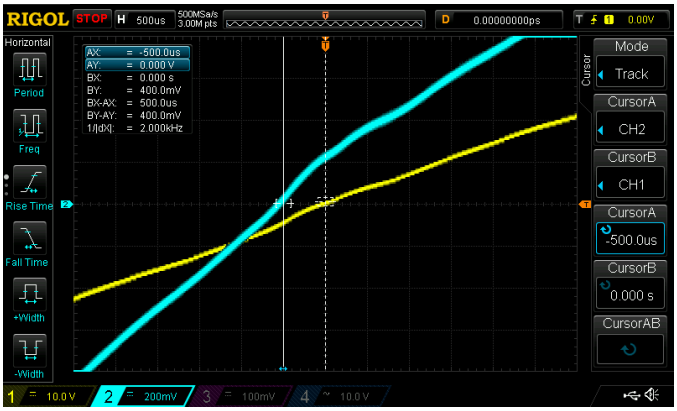


Fig. 2. Input (channel 1; yellow) and output (channel 2; cyan) voltages when supplying the transformer with 16 V$\sim$ at 50 Hz and using a 330 Ω burden resistor. The cursors show a discrepancy of 500 μs, i.e., a phase shift of 9°.

B. Analog-to-Digital Conversion

While many microcontroller-based embedded sensing solutions feature integrated ADCs, their resolution is commonly limited to a narrow value range. For example, the ADC of the well-known TelosB mote provides an amplitude resolution of 12 bits. In contrast, integrated circuits for power metering often feature sample resolutions of 20 bits (effective) or even more. As our objective was to capture voltage and current at high temporal and amplitude resolutions, we have decided in favor of a dedicated energy metering integrated circuit. The selected Microchip MCP3910 [20] device serves as the analog front end and converts values from the voltage and current transducers into digital signals. It synchronously samples voltage and current and allows for the retrieval of the readings over a three-wire serial interface. In addition to sampling the transducers' outputs, the MCP3910 features an integrated amplifier with configurable amplification (up to 32x) which can be used to monitor small currents if needed. When operated with a crystal at 18.432 MHz, the device reliably achieves a sampling rate of 36 kSps at 23 bits resolution, thus allowing for the in-depth spectral analysis of the captured signal. A photography of our TUCap board is shown in Fig. 3.

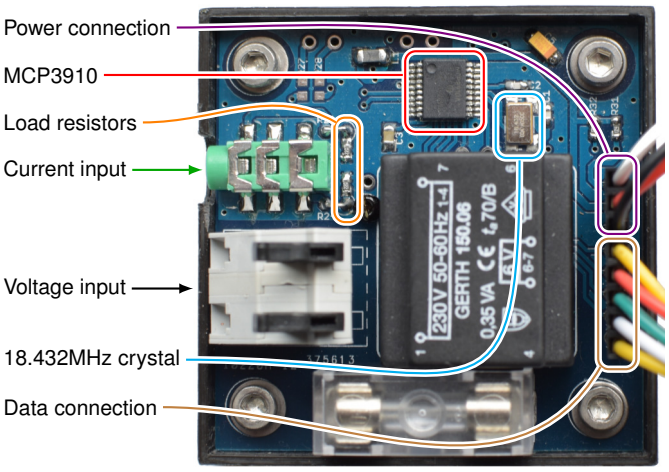
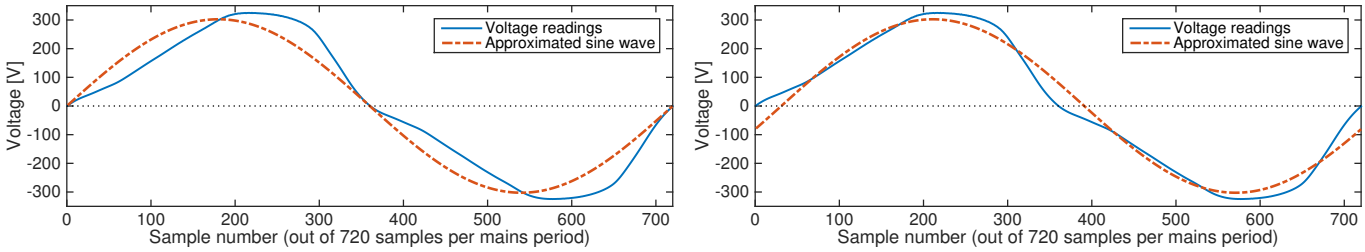


Fig. 3. Key components on the TUCap sensor interface board.



(a) Before phase correction (i.e., aligned at zero-crossings)

(b) After phase correction (minimization of discrepancies)

Fig. 4. Collected voltage waveforms and sine wave approximations before and after phase correction.

C. Data Retrieval and Preprocessing

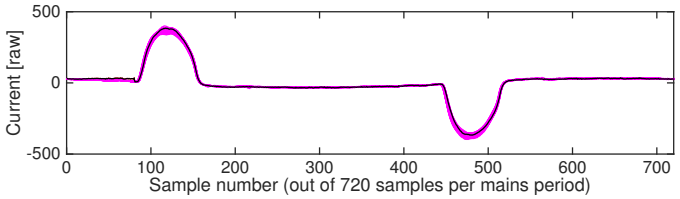
In order to allow for the collection and transmission of power consumption readings and other values related to load signatures, we connect the MCP3910 analog frontend to a Teensy 3.0 system [21]. Whenever a falling edge is detected on the ADC's data ready signal ($\overline{\mathrm{DR}}$), the Teensy retrieves the latest voltage and current values over the SPI interface and stores them in corresponding buffers. A zero-crossing detection routine has been implemented to identify the beginning of a mains voltage period. It triggers the execution of the data preprocessing operations when 720 samples have been collected with exactly one zero-crossing observed in the voltage waveform during this period.

As mentioned above, the MCP3910 is capable of sampling its voltage and current channels at a rate of 36 kSps and 23 bits of accuracy. Due to the word alignment of the Teensy's 32-bit ARM microcontroller, however, the measurements would need to be stored as 32 bit values, resulting in a data generation rate of more than 280 kB/s. The resulting data rate (2.25 MBps) is beyond the capacities of most communication links available to low-power embedded systems. To mitigate this issue, TUCap only retrieves 16 bit values from the ADC. The resulting amplitude resolution of 13 mV on the voltage channel and 0.5 mA for the current still allows for power measurements with a resolution better than 0.2 W.

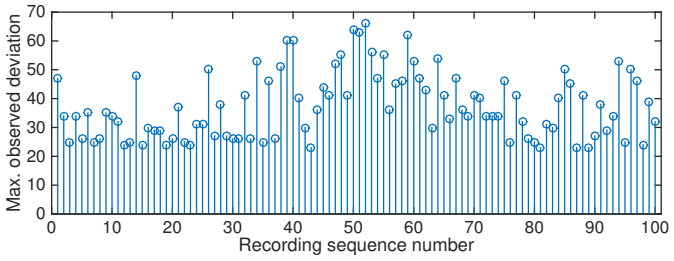
Nonetheless, the need for local data preprocessing becomes apparent when considering the volume of data generated. TUCap thus executes two data preprocessing steps:

1) The sinusoidal nature of the mains voltage enables TUCap to approximate collected waveforms by a parametrizable sine wave. Instead of transferring 1,440 bytes of raw data samples, only two values are reported: The sine wave's amplitude and its phase shift. The amplitude of the parametrizable sine wave is computed by calculating the absolute value of the area under the curve of the voltage waveform and normalizing it to the absolute area under a sine wave ($\int_0^{2\pi} |\sin x| \, \mathrm{d}x = 4$). Subsequently, the phase shift is determined by iteratively comparing the sampled voltage trace to parametrizable sine waves at different offsets until the observed discrepancy is minimal. This step is visualized in Fig. 4. Finally, the fixed offset introduced by the voltage transformer (shown in Fig. 2) is subtracted from the reported value.
2) Current consumption waveforms of electrical appliances often remain highly similar for extended periods of time. They mostly only alternate when the operation mode of the appliance changes. To demonstrate this in practice, we have used TUCap to recorde 100 periods of the waveform of a computer monitor's power consumption when playing back a video. The superposition of all recorded waveforms is shown in Fig. 5a and demonstrates this coherence. To avoid transmitting redundant data, TUCap thus captures a single waveform of the current and stores it in memory (similar to the notion of an *I-Frame* in MPEG video coding [22]). All signals captured subsequently undergo an element-wise comparison to this previously stored waveform. Unless elements show a difference greater than a user-configurable threshold value level ρ, TUCap does not report any update.

A supplementary analysis of practically determined deviations between an initial waveform and later traces is shown in Fig. 5b. It is visible that slight differences always occur due to the presence of noise, thus it is not meaningful to set $\rho < 20$. Other than that, the choice of ρ defines the frequency of reports and thus the accuracy of reported waveforms.



(a) Overlaid visualization of 100 current flow recordings taken sequentially. The black line shows the course of the waveform initially recorded.



(b) Maximum absolute differences observed between each of the 100 recordings and the initially stored current waveform.

Fig. 5. A highly similar current consumption pattern can be observed when comparing 101 current consumption recordings of a 20" TFT monitor.

III. Practical Considerations

During the design of TUCap, a set of practical insights into the operation of an energy monitoring platform were gained.

1) Choice of Components: Manufacturing tolerances of the used components make it complicated to lay both input channels out in a symmetric fashion. In particular, load resistors and voltage dividers must have closely matching values in order to attain symmetric voltage swings that feature identical amplitudes in both the positive and the negative direction. Despite the choice of resistors with 1% tolerance, we found the pre-selection of resistors through measurements using a high-accuracy LCR meter to be indispensable.

2) Calibration: An initial factory calibration step must precede the operation of TUCap in practice. To encounter component tolerances, the MCP3910 features calibration registers for both voltage and current. A reference AC power supply is inevitable to supply input voltages and currents of of known amplitude and frequency to the device in order to determine and compensate any offsets in hard- or software.

3) Shielding: During some of our practical experiments, TUCap has infrequently been unable to retrieve data over the SPI connection. Insufficient shielding of the data connectors and their cable lengths could be identified as the primary source of error. An integration of the ADC and the microcontroller on a single circuit board and/or better cable shielding is thus recommended.

IV. Conclusions

Energy informatics research has received a lot of attention in the last decade, primarily due to the increasing availability of data collected by smart meters and smart plugs. A strong limitation of most commercially available platforms, however, is their limited sampling rate of 1 Hz or even less. In order to foster energy informatics research, we have thus presented TUCap. The synergistic combination of a high-resolution ADC to capture voltage and current waveforms with an embedded system to process and transfer collected data has emerged as an ideal solution for our use case. By choosing small values for ρ, highly accurate consumption traces can be recorded; in turn, selecting a larger value for the tolerance parameter reduces generated traffic and may thus enable TUCap's use in scenarios where bandwidth is scarce. As the next step in TUCap's development, we plan to implement more algorithms for local data analytics and compression on the embedded system, such as the ones surveyed in [23, 24].

References

[1] G. W. Hart, "Prototype Nonintrusive Appliance Load Monitor," MIT Energy Laboratory and Electric Power Research Institute, Tech. Rep., 1985.

[2] M. Weiss, A. Helfenstein, F. Mattern, and T. Staake, "Leveraging Smart Meter Data to Recognize Home Appliances," in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2012.

[3] C. Beckel, L. Sadamori, T. Staake, and S. Santini, "Revealing Household Characteristics from Smart Meter Data," *Elsevier Energy*, vol. 78, 2014.

[4] W. Kleiminger, C. Beckel, and S. Santini, "Household Occupancy Monitoring Using Electricity Meters," in *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*, 2015.

[5] V. Bianco, O. Manca, and S. Nardini, "Electricity Consumption Forecasting in Italy using Linear Regression Models," *Energy*, vol. 34, no. 9, 2009.

[6] A. Reinhardt, D. Christin, and S. S. Kanhere, "Can Smart Plugs Predict Electric Power Consumption? A Case Study," in *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous)*, 2014.

[7] M. Gulati, S. S. Ram, and A. Singh, "An in Depth Study into Using EMI Signatures for Appliance Identification," in *Proceedings of the ACM Conference on Embedded Systems for Energy-Efficient Buildings (BuildSys)*, 2014.

[8] M. Kahl, A. Ul Haq, T. Kriechbaumer, and H.-A. Jacobsen, "WHITED – A Worldwide Household and Industry Transient Energy Data Set," in *Proceedings of the 3rd International Workshop on Non-Intrusive Load Monitoring (NILM)*, 2016.

[9] A. Marchiori, D. Hakkarinen, Q. Han, and L. Earle, "Circuit-Level Load Monitoring for Household Energy Management," *IEEE Pervasive Computing*, vol. 10, no. 1, 2010.

[10] D. Jung and A. Savvides, "Estimating Building Consumption Breakdowns Using ON/OFF State Sensing and Incremental Sub-meter Deployment," in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2010.

[11] T. W. Hnat, V. Srinivasan, J. Lu, T. I. Sookoor, R. Dawson, J. Stankovic, and K. Whitehouse, "The Hitchhiker's Guide to Successful Residential Sensing Deployments," in *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2011.

[12] J. Liao, G. Elafoudi, L. Stankovic, and V. Stankovic, "Non-intrusive Appliance Load Monitoring using Low-resolution Smart Meter Data," in *Proceedings of the 5th IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2014.

[13] H. Nyquist, "Certain Topics in Telegraph Transmission Theory," *Transactions of the American Institute of Electrical Engineers*, vol. 47, 1928.

[14] J. Lifton, M. Feldmeier, Y. Ono, C. Lewis, and J. Paradiso, "A Platform for Ubiquitous Sensor Deployment in Occupational and Domestic Environments," in *Proceedings of the 6th International Symposium on Information Processing in Sensor Networks (IPSN)*, 2007.

[15] X. Jiang, S. Dawson-Haggerty, P. Dutta, and D. E. Culler, "Design and Implementation of a High-Fidelity AC Metering Network," in *Proceedings of the 8th International Conference on Information Processing in Sensor Networks (IPSN)*, 2009.

[16] T. Weng, B. Balaji, S. Dutta, R. Gupta, and Y. Agarwal, "Managing Plug-loads for Demand Response Within Buildings," in *Proceedings of the 3rd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings (BuildSys)*, 2011.

[17] A. Reinhardt, D. Burkhardt, P. S. Mogre, M. Zaheer, and R. Steinmetz, "SmartMeter.KOM: A Low-cost Wireless Sensor for Distributed Power Metering," in *Proceedings of the 6th IEEE Workshop on Practical Issues in Building Sensor Network Applications (SenseApp)*, 2011.

[18] D. Porcarelli, D. Balsamo, D. Brunelli, and G. Paci, "Perpetual and Low-cost Power Meter for Monitoring Residential and Industrial Appliances," in *Proceedings of the Design, Automation Test in Europe Conference Exhibition (DATE)*, 2013.

[19] C. Klemenjak, D. Egarter, and W. Elmenreich, "YoMo: The Arduino-based Smart Metering Board," *Computer Science – Research and Development*, vol. 31, no. 1, 2016.

[20] Microchip Technology Inc., "MCP3910 – 3V Two-Channel Analog Front End," Data Sheet available online at http://ww1.microchip.com/downloads/en/DeviceDoc/20005116B.pdf, 2014.

[21] PJRC LLC., "Teensy USB Development Board," available online at https://www.pjrc.com/teensy/index.html, last access on 26 June 2017.

[22] D. Le Gall, "MPEG: A Video Compression Standard for Multimedia Applications," *Communications of the ACM*, vol. 34, no. 4, 1991.

[23] M. Kahl, A. Ul Haq, T. Kriechbaumer, and H.-A. Jacobsen, "A Comprehensive Feature Study for Appliance Recognition on High Frequency Energy Data," in *Proceedings of the Eighth International Conference on Future Energy Systems (e-Energy)*, 2017.

[24] M. Ringwelski, C. Renner, A. Reinhardt, A. Weigel, and V. Turau, "The Hitchhiker's Guide to Choosing the Compression Algorithm for Your Smart Meter Data," in *Proceedings of the 2nd IEEE Conference and Exhibition / ICT for Energy Symposium (ENERGYCON)*, 2012.

Eine Testplattform für Energy Harvesting mit RIOT

Michel Rottleuthner, Thomas C. Schmidt
Internet Technologies Group, Dept. Informatik, HAW-Hamburg, Germany
{michel.rottleuthner, t.schmidt}@haw-hamburg.de

***Zusammenfassung*—Der Begriff „Internet of Things"(IoT) beschreibt ein globales Netzwerk in dem Gegenstände des täglichen Lebens von Sensoren und Aktoren repräsentiert werden, die direkt miteinander kommunizieren. Um neue Geräte im IoT zu etablieren, ohne regelmäßig proprietäre Insellösungen zu generieren, bietet sich die Verwendung offener Standards und einer robusten Software-Basis an, die unterschiedlichste Hardware auf eine entwicklerfreundliche Ebene abstrahiert. RIOT ist ein Betriebssystem für das IoT, das diesem Grundsatz folgt und damit eine effiziente Entwicklung von plattformunabhängigen IoT-Anwendungen ermöglicht. Ein Themengebiet, das bisher nur eingeschränkt von RIOT unterstützt wird, besteht in intelligenten Mechanismen zur Energieverwaltung. Insbesondere für energieautarke Systeme, die ihre Energie aus der Umgebung schöpfen, stellen generische Lösungen zur dynamischen Energieverwaltung eine wichtige Funktion des Betriebssystems dar. Mit dieser Arbeit wird ein erster Beitrag geleistet um weitere Entwicklungen und Tests der Energieverwaltung in RIOT zu vereinfachen. Dazu wird eine modulare Testplattform für Energy Harvesting (EH) Systeme aufgebaut, die stationäre Versuche und Feldtests ermöglicht und dabei die Aufzeichnung von Messdaten zur Energiebilanz des zu untersuchenden Systems erlaubt.**

I. Einleitung

Die Bausteine des IoT bestehen in der Regel aus kleinen eingebetteten Systemen mit stark eingeschränkten Ressourcen. Dies betrifft nicht nur die Rechenleistung und Speicherkapazität, sondern auch die Energieverfügbarkeit. Neben offenen Standards zur Kommunikation ist eine zuverlässige Energieversorgung der einzelnen Geräte eine wichtige Grundvoraussetzung für ein funktionierendes IoT. Energy Harvesting (EH) bietet als Alternative zu endlichen Batteriekapazitäten oder einer externen Energieversorgung eine Lösung, mit der Sensorknoten dauerhaft energieautark operieren können. Für EH lassen sich gängige Energiequellen wie beispielsweise Sonnen-, Wind- und Wasserkraft nutzen [1]. Auch alternative Ansätze wie Thermoelektrizität, menschliche Bewegungen [2], [3], Vibrationen oder Funkwellen [4] finden Verwendung. Trotz erheblicher Unterschiede zwischen den Eigenschaften der Energiequellen werden auf Software-Ebene des eingebetteten Systems allgemeine Energiesparmechanismen und eine möglichst intelligente Energieverwaltung benötigt, um die Leistungsfähigkeit der Anwendung im Rahmen der Energieverfügbarkeit zu maximieren.

Zur Unterstützung bei der Entwicklung von EH-Systemen und generischen Lösungen zur Energieverwaltung für das IoT-Betriebssystem RIOT [5], [6], wird in dieser Arbeit der Aufbau einer Testplattform vorgestellt. Dazu wird in Abschnitt II die Problemstellung genauer beschrieben. Anschließend wird in Abschnitt III von einer abstrakten Beschreibung des Gesamtsystems ausgehend, über die detailliertere Beschreibung der Einzelkomponenten zum vollständigen Aufbau eines Prototypen in Abschnitt IV hingeführt. Abschließend wird in Abschnitt V eine Zusammenfassung und ein Ausblick auf weiterführende Arbeiten gegeben.

II. Problemstellung und verwandte Arbeiten

Eine Schwierigkeit bei der Implementierung generischer Energieverwaltungsmechanismen auf Ebene des Betriebssystems besteht in den stark variierenden Konfigurationsmöglichkeiten der verschiedenen Zielplattformen. Beispiele hierfür sind verschiedene Low-Power Modi, MCU-interne Taktkonfigurationen oder Teilabschaltungen von internen und externen Peripheriebausteinen. Weiterhin entstehen zwischen diesen Ressourcen dynamische Abhängigkeiten, die sich zur Laufzeit verändern. Generische Komponenten für die Energieverwaltung auf Betriebssystemebene zu implementieren, welche gleichzeitig effektiv, effizient und zuverlässig sind, ist daher kein triviales Problem.

Der Umfang dieser Problematik lässt sich anhand des eingesetzten Betriebssystems einfach illustrieren. RIOT ist ein offenes IoT-Betriebssystem für ressourcenarme eingebettete Systeme. Es unterstützt verschiedene Architekturen wie z. B. AVR, MSP-430, ARM oder MIPS und bietet Entwicklern die Möglichkeit ihren Code ohne Anpassungen auf allen unterstützen Plattformen einzusetzen. RIOT stellt dazu eine umfangreiche Hardwareabstraktionsschicht bereit und erleichtert dem Entwickler durch bekannte Programmierparadigmen und Standardschnittstellen modulare Anwendungen zu erstellen. Die aktuelle Version von RIOT[1] unterstützt dabei über 80 verschiedene Zielplattformen. Zusätzlich zur großen Anzahl an Plattformen entstehen durch die kontinuierliche Weiterentwicklung Herausforderungen bezüglich Testmethoden und Verifikation der Energieverwaltung. RIOT wird von einer aktiven Community entwickelt und regelmäßig um neue Funktionen erweitert. Durch regelmäßige Änderungen sind automatisierte Tests zur Qualität und Funktionalität des Quellcodes unverzichtbar. Jede kleine Änderung einzeln auf allen relevanten Plattformen zu testen, wäre mit moderatem Aufwand nicht realisierbar. Ein Continuous Integration (CI) System schafft hierbei Abhilfe. Fehler im Code können damit erkannt und behoben werden, bevor sie in die Code-Basis übernommen, oder gar auf einem Produktivsystem eingesetzt werden. Der im IoT-Umfeld besonders wichtige Aspekt des Energieverbrauchs

[1]RIOT Release 2017.04 https://github.com/RIOT-OS/RIOT/tree/2017.04-branch

wird dabei häufig nicht ausreichend abgedeckt. Das CI System von RIOT bietet derzeit keine Möglichkeit, automatisiert die Auswirkungen von Software-Änderungen auf den Energieverbrauch zu Messen, um vorab vor starken Abweichungen zu warnen oder Energie-bezogene Optimierungen zu quantifizieren. Ein System das CI-Tests zum Energieverbrauch in Form von Unit-Tests durchführt stellen Woehrle et al. in ihrer Arbeit vor [7].

Abseits von Laborversuchen und Simulationen spielen für kabellose Sensorknoten Feldtests eine wichtige Rolle. Für EH-Systeme gilt dies im besonderen Maße, da deren Energieversorgung und weitere Systemparameter von dynamischen Umwelteinflüssen abhängen [8]. So kann eine falsche Annahme oder die Vernachlässigung eines Parameters in Laborversuchen zu Ergebnissen führen, die für einen realen Einsatz des Systems nicht belastbar sind.

III. Energy Harvesting Testplattform

Wir stellen eine Testplattform für EH-Systeme und anderen kabellose Sensorknoten vor, die Laborversuche und Feldtests zur Energieverwaltung im RIOT-Kontext vereinfachen soll. Dafür soll die Testplattform ohne externe Geräte Messdaten zum Energieverbrauch des Sensorknotens überwachen und aufzeichnen. Die gewonnenen Daten aus möglichst praxisnahen Tests, sollen anschließend zur realistischen Bewertung von verschiedenen Energieverwaltungsmechanismen in RIOT dienen. Die Plattform wird zu Gunsten der Flexibilität modular und unabhängig von verwendeten Evaluierungsboards gehalten. Abbildung 1 zeigt den schematischen Aufbau der Testplattform.

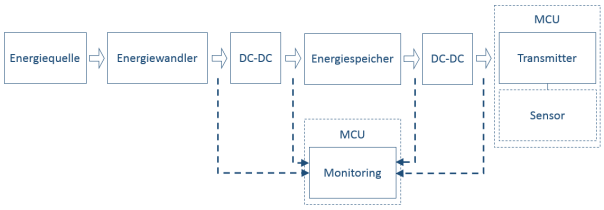


Abbildung 1: Schematischer Aufbau eines EH-Systems mit Monitoring Komponente

In der Testplattform wird eine Photovoltaik- (PV) Zelle verwendet. Diese lädt über ein Lademodul, basierend auf einem Maximum Power Point (MPP) Schaltregler einen Superkondensator. Aus diesem Superkondensator wird über einen DC-DC-Konverter ein Sensorknoten gespeist. Um den Energieverbrauch des Sensorknotens messen und die Messwerte speichern zu können, existiert eine Monitoring Komponente, die aus einem externen Messmodul und einem Datenspeicher in Form einer SD-Karte besteht. Das Messmodul kann je nach Anwendungsszenario in ein- oder mehrfacher Ausführung an einem separaten Mikrocontroller oder an dem Sensorknoten selbst angebracht werden. Sowohl für das Messmodul, als auch für die SD-Karte wird ein Treiber zur Verwendung in RIOT bereitgestellt. Nachfolgend werden die Eigenschaften der einzelnen Komponenten genauer erläutert.

A. Superkondensator

Als Energiespeicher für das EH-System wird ein Superkondensator verwendet. Superkondensatoren bieten gegenüber chemischen Batterien die Vorteile einer weitaus höheren Lebensdauer, eines einfachen Ladeverfahrens und einer hohen physikalischen Robustheit. Allerdings weisen sie eine wesentlich niedrigere Energiedichte und eine höhere Selbstentladung auf [9]. Zur Bestimmung des Ladezustandes lassen sich leichtgewichtige Modelle verwenden [10]. Durch die höhere Lebensdauer von Superkondensatoren sind längerfristige Einsätze im Bereich von Jahrzehnten realisierbar. Abbildung 2 zeigt den verwendeten Superkondensator. Er besitzt eine Kapazität von $100\,\mathrm{F}$ und eine Nennspannung von $2.7\,\mathrm{V}$. Nach Gleichung 1 kann damit eine Energiemenge von bis zu $364.5\,\mathrm{W\,s}$ gespeichert werden.

$$E_1 = \frac{1}{2} C \cdot U^2 \qquad (1)$$

$$E_2 = \frac{1}{2} C \cdot (U^2 - {U_{min}}^2) \qquad (2)$$

Abbildung 2: Superkondensator

Da die Spannung an einem Kondensator abfällt, sobald Energie daraus entnommen wird, kann nur ein Teil der Kapazität tatsächlich verwendet werden. Die tatsächlich verwertbare Kapazität eines voll geladenen Kondensators wird somit direkt durch die minimale Eingangsspannung des zweiten DC-DC-Konverters (in Abbildung 1 rechts) bestimmt. Gleichung 2 liefert die verwertbare Energie in Abhängigkeit von Nennspannung U und Minimalspannung U_{min} des Superkondensators.

Je nach Betriebsspannung und Energieverbrauch der verwendeten MCU und der Effizienz des eingesetzten Konverters kann der Wert für U_{min} stark variieren. Mit einem einfachen, handelsüblichen DC-DC-Konverter (ME2108A33) mit fester Ausgangsspannung von $3.3\,\mathrm{V}$ und einem großzügig dimensionierten Verbraucher lässt sich experimentell eine Untergrenze bestimmen. Bei einer durchschnittlichen Belastung von $40\,\mathrm{mA}$ durch ein Evaluierungsboard mit angeschlossenem Temperatursensor zur Messung und einem über SPI angebundenen LC-Display zur Anzeige, lag die Untergrenze für den erfolgreichen Betrieb des Systems bei knapp unter $1\,\mathrm{V}$. In diesem Beispiel liegt die verwertbare Energiemenge E_2 bei $314.5\,\mathrm{W\,s}$. Auf den Einsatz eines Konverters der geringere Eingangsspannungen erlaubt, wird vorerst verzichtet. Hierfür sind mehrere Gründe zu nennen. Bei weiter sinkender Spannung wird die Effizienz bei der Entnahme der Energie, als auch beim Wiederaufladen, zunehmend schlechter. Weiterhin verbleiben bei $1\,\mathrm{V}$ Ladespannung lediglich ca. $13.7\,\%$ der Energie im Kondensator, da sich die Spannung quadratisch auf die Energiemenge im Kondensator auswirkt.

B. Lademodul

Das Lademodul basiert auf dem Schaltregler LTC3105[2] von Lienar Technologies. Der hierzu angefertigte Prototyp bietet die Möglichkeit über ein Potentiometer die Eingangspannung auf den Maximum Power Point (MPP) der PV-Zelle einzustellen um diese im Leistungsmaximum zu betreiben. Die gewählte Bestückung der Platine erlaubt hierbei die Einstellung im Bereich unterhalb von 5 V. Ein weiteres Potentiometer erlaubt die Einstellung der Ausgangsspannung zwischen 2.24 V und 5.12 V. Abbildung 3 zeigt den Prototypen mit der Seite der Anschlussklemmen und den Potentiometern. Zusätzlich zum Eingang für die PV-Zelle (IN-, IN+) und dem Ausgang für den Superkondensator (OUT-, OUT+) verfügt das Modul über einen Eingang zum Aktivieren und Deaktivieren der Schaltung (SHDN) und einen Ausgang der signalisiert, ob die Ausgangsspannung im Rahmen der Toleranz auf die Zielspannung geregelt wird.

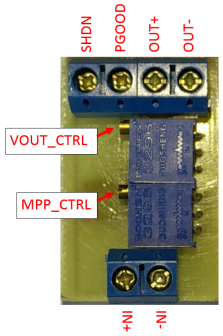


Abbildung 3: Prototyp des Ladereglers

C. Messmodul

Das Messmodul basiert auf dem Texas Instruments Baustein INA226[3]) und ist in der Lage, gleichzeitig die Spannung und den Strom (über den Spannungsabfall an einem Shunt-Messwiderstand) an einer angeschlossenen Last zu überwachen. Der Baustein verwendet zur Messung einen $\Delta\Sigma$-ADC mit 500 kHz Abtastrate. Das Modul wird über eine Inter-Integrated Circuit (I^2C) Schnittstelle an einen Mikrocontroller angebunden. Der Prototyp bietet die Möglichkeit, die I^2C-Adresse des Bausteins mit Jumper-Brücken einzustellen, und so mehrere baugleiche Module gleichzeitig an einem Bus zu betreiben. Der Messbereich ist über Schalter einstellbar, die zwischen den verschiedenen Messwiderständen auswählen. Durch den höchsten messbaren Spannungsabfall von 81.92 mV am Messwiderstand kann der Messbereich durch Auswahl eines geeigneten Widerstands dimensioniert werden. Bei Verwendung eines 750 mΩ Widerstands ergibt sich beispielsweise ein Messbereich bis ca. 109 mA bei einer Auflösung von rund 3.3 µA. Für einem Messwiderstand von 10 Ω ergibt sich ein Messbereich bis ca. 8 mA bei einer Auflösung von 0.25 µA. Weiterhin bietet das Modul Einstellungen für Konvertierungszeiten zwischen 140 µs und 8.2 ms und einer automatischen Mittelwertbildung für bis zu 1024 Werte. Für die Verwendung des Messmoduls unter dem Betriebssystem RIOT wird ein Low-Level-Treiber erstellt.

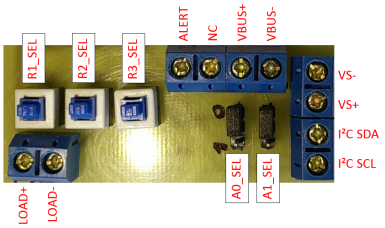


Abbildung 4: Prototyp des Messmoduls

D. Speicher für Messdaten

Ein Einsatzzweck der Testplattform ist es, unter möglichst authentischen Bedingungen Feldtests mit Sensorknoten durchführen zu können. Dazu ist eine Komponente für das Protokollieren der vom Messmodul geliefert Daten erforderlich. Aufgrund des eingeschränkten Volumens des internen Speichers einer klassischen MCU, wird hierzu ein externes Speichermedium benötigt. Ausgewählt werden hierfür SD-Karten. Der Standard für SD-Karten sieht vor, dass jede Karte neben dem nativen SD-Modus über einen SPI-Modus verwendet werden kann. Eine SPI Schnittstelle ist wiederum auf sehr vielen Mikrocontrollern als Peripheriebaustein verfügbar oder kann bei Verfügbarkeit der nötigen GPIO Pins unkompliziert in Software nachgestellt werden. Damit ist eine hohe Kompatibilität zu verschiedensten Mikrocontrollern sichergestellt.

Weiterhin besitzen SD-Karten bereits in ihrem kleinsten Formfaktor *microSD* bis zu 256 GB[4] Speichervolumen und bieten damit auch für Langzeitmessungen mit erhöhtem Datenaufkommen eine ausreichende Kapazität. Aufgrund der häufigen Verwendung von SD-Karten in Mobilgeräten wie Smartphones kann angenommen werden, dass der Energieverbrauch von SD-Karten auch Feldtests ohne externe Stromversorgung zulässt. Diese Annahme soll in weiteren Versuchen genauer überprüft werden, um die Grenzen des Systems detailliert aufzuzeigen. Ein weiterer Grund für die Verwendung von SD-Karten besteht in der einfachen Handhabung und der direkten Interoperabilität mit handelsüblichen Geräten. Für die Verwendung der SD-Karten wird ein Treiber für RIOT implementiert. Um die Verarbeitung der aufgezeichneten Daten weiter zu vereinfachen, wird außerdem eine Bibliothek für FAT Dateisysteme integriert. Damit können Messdaten vom Mikrocontroller z. B. direkt im CSV-Format abgelegt werden.

IV. Aufbau der gesamten Testplattform

Ein Prototyp der gesamten Testplattform ist in Abbildung 5 zu sehen. Die hier gezeigte Konfiguration besteht aus zwei

[2]http://cds.linear.com/docs/en/datasheet/3105fb.pdf
[3]http://www.ti.com/lit/ds/symlink/ina226.pdf

[4]https://www.sandisk.de/about/media-center/press-releases/2016/western-digital-launches-worlds-fastest-256gb-microsd-card-Broadens-sandisk-memory-card-portfolio-with-new-high-capacity-solutions

Teilsystemen. Das erste Teilsystem besteht aus einem Atmel SAM R21 Xplained Pro Mikrocontroller (A), welcher Strom- und Spannungsmessungen mit dem Messmodul (C) durchführt und die Ergebnisse auf einer Speicherkarte (B) ablegt. Die dedizierte Ausführung des Messsystems erlaubt eine externe Energieversorgung und damit eine weniger invasive Messung. Alternativ zur lokalen Speicherung auf der SD-Karte können die Messdaten per Funk übertragen werden, wenn die Bandbreite für das gewünschte Messintervall ausreicht und die zusätzliche Belegung des geteilten Funkmediums für den Versuch irrelevant ist.

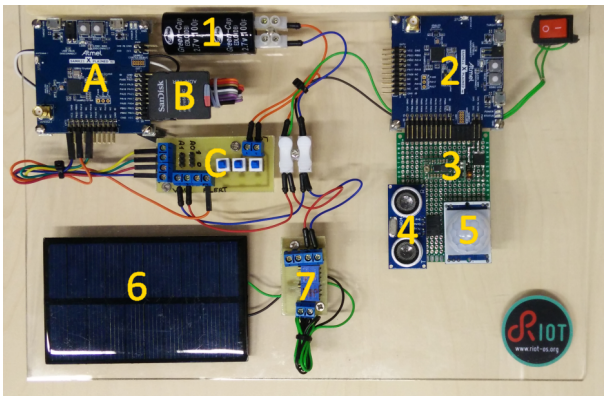


Abbildung 5: Testplattform mit dediziertem Messkonten (A-C) und EH-Präsenzsensor (1-7)

Das zweite Teilsystem ist das zu untersuchende EH-System. In diesem Aufbau erfüllt es den Zweck eines einfachen Präsenzsensors und setzt sich aus dem Superkondensator (1), einem Mikrocontroller (2), zwei Spannungsreglern (3) für MCU (3.3 V) und Sensorik (5 V), einem Ultraschallsensor (4), einem PIR (Pyroelektrisch Infrarot) Sensor (5) und einer PV-Zelle (6) mit angeschlossenem Lademodul (7) zusammen. Durch eine Bewegungsdetektion des PIR Sensors wird der EH-Knoten eingeschaltet. Anschließend werden mit dem Ultraschallsensor Distanzmessungen durchgeführt, deren Ergebnisse über die integrierte IEEE 802.15.4 Funkschnittstelle an eine Basisstation übertragen werden.

Erste Tests mit dem Gesamtaufbau zeigen auf, wo weitere Optimierungen nötig sind. Die physikalische Empfindlichkeit der Messbereichsschalter am Shunt-Messmodul legt die Verwendung von anderen Schaltern nahe. Die verbauten Schalter verändern bei Vibrationen oder Berührungen ihren Innenwiederstand wodurch die Messergebnisse beeinflusst werden und regelmäßige Kalibrierungen nötig werden. Zur Lösung soll auf unempfindlichere Schalter mit einem besonders niedrigen Innenwiederstand gewechselt werden.

Ein allgemeines Problem bezüglich der Feldtests besteht darin, dass viele Evaluierungsboards nicht, oder nur mit Modifikationen verwendet werden können. Auf dem verwendeten Board von Atmel befinden sich z. B. ein fest verbauter Baustein für Debugging-Anwendungen und LEDs welche nicht über die MCU abgeschaltet werden können. Dadurch liegt der Energiebedarf des Evaluierungsboards ohne Modifikationen teilweise um einen Faktor 1000 über dem Energieverbrauch der darauf verbauten MCU. Um unnötige Energieverbraucher zu eliminieren und damit das volle Potenzial der MCU-eigenen Energiesparmechanismen nutzbar zu machen, müssen daueraktivierte LEDs und Lötbrücken für den Debug-Chip entfernt werden. Auf anderen Plattformen können auch ineffiziente Spannungsregler oder weitere, nicht deaktivierbare Peripherie Probleme bereiten.

V. Zusammenfassung und nächste Schritte

In dieser Arbeit wurde der Aufbau einer modularen Testplattform für RIOT-basierte EH-Systeme gezeigt. Dazu wurden einzelne Module für diesen Anwendungszweck entworfen, in prototypische Hardware umgesetzt und mit einer Software-Unterstützung durch RIOT ausgestattet. Der vollständige Testaufbau vereinfacht bereits die Durchführung von Messreihen zum Energieverbrauch und zeigt erste Optimierungsmöglichkeiten auf.

Weiterführend sollen Versuche und Messungen an einzelnen Komponenten der Testplattform zeigen, wo die Grenzen des Aufbaus für Feldtests liegen und an welchen Stellen weiterer Optimierungsbedarf besteht. Dazu müssen die Einzelkomponenten der Testplattform hinsichtlich ihres Energiebedarfs genauer untersucht werden. So soll die Speicherkomponente insbesondere auf ihre Eignung für Feldtests ohne externe Energieversorgung geprüft werden. Dafür sollen sowohl Konfigurationen mit verschiedenen Kartenmodellen, als auch verschiedene Puffergrößen zum Akkumulieren von Schreibvorgängen untersucht werden. Außerdem soll der Overhead zum Ein- und Ausschalten der Karte ihrem Energieverbrauch im Sleep-Mode gegenübergestellt werden. Die Ergebnisse sollen in Abhängigkeit der zuvor genannten Systemparameter den Energiebedarf je gespeichertem Byte zeigen, um Vergleichswerte zur Funkübertragung der Daten zu liefern. Auf Betriebssystemebene besteht ein weiteres Arbeitspaket in der Implementierung und Erweiterung von Funktionen zur Steuerung von Low-Power Zuständen der MCU.

Literatur

[1] Faisal K. Shaikh. et al., “Energy harvesting in wireless sensor networks: A comprehensive review,” *Renewable and Sustainable Energy Reviews*, vol. 55, pp. 1041–1054, 2016.

[2] Longhan Xie. et al., “An In-shoe Harvester with Motion Magnification for Scavenging Energy from Human Foot Strike,” vol. 20, pp. 3264–3268, December 2015.

[3] Maria Gorlatova. et al., “Movers and Shakers: Kinetic Energy Harvesting for the Internet of Things,” vol. 33, pp. 1624–1639, January 2015.

[4] P. Chambe. et al., “Optimization of Energy Harvesting Systems for RFID Applications,” vol. 8, no. 7, pp. 1147–1150, 2014.

[5] Emmanuel Baccelli. et al., “RIOT OS: Towards an OS for the Internet of Things,” in *Proc. of the 32nd IEEE INFOCOM. Poster*, 2013.

[6] Peter Kietzmann. et al., “RIOT - das freundliche Echtzeitbetriebssystem für das IoT,” in *Internet der Dinge*, Berlin, Nov. 2016, pp. 43–52.

[7] Matthias Woehrle. et al., “Power monitoring and testing in wireless sensor network development,” in *WEWSN*, 2008.

[8] Bernd-C. Renner, *Sustained Operation of Sensor Nodes with Energy Harvesters and Supercapacitors*. BoD–Books on Demand, 2013.

[9] Sujesha Sudevalayam. et al., “Energy harvesting sensor nodes: Survey and implications,” vol. 13, pp. 443–461, 2011.

[10] Bernd-C. Renner and V. Turau, “State-of-charge assessment for supercap-powered sensor nodes: Keep it simple stupid!” in *International Conference on Networked Sensing (INSS)*, June 2012, pp. 1–6.

Kommunikation in heterogenen Sensornetzwerken mittels Cross-Plattform und Multiradio-Gateways

Paul Poppe, Danny Puhan, Sebastian Reinhold
IHP Microelectronics
Frankfurt (Oder), Germany
{ppoppe, puhan, sreinhol}@uni-potsdam.de

Max Frohberg, Mario Schölzel
IHP Microelectronics
Frankfurt (Oder), Germany
{frohberg, schoelzel}@ihp-microelectronics.com

***Zusammenfassung*—Änderungen an Aufbau und Zusammensetzung von Sensornetzwerken erzeugen einen erheblichen Aufwand bei der Portierung der eingesetzten Anwendungen und Kommunikationsprotokolle. Durch die Verlagerung des Netzwerkstacks in die Cross-Plattform als Middleware lässt sich dieser Aufwand mindern. In diesem Paper wird der Aufbau eines heterogenen Netzwerks mit einem Multiradioknoten als Gateway beschrieben. Anhand dieses Netzwerks wird der Implementierungsaufwand für zusätzliche Schichten am Beispiel zweier Medienzugriffsprotokolle untersucht und die Eignung der Cross-Platform für zeitkritische Kommunikationsprotokolle geprüft.**

I. Einleitung

Mit der zunehmenden Digitalisierung werden unter anderem Sensornetzwerke zur Erfassung von Messdaten eingesetzt. Die Erweiterung bestehender Systeme durch eine stetig wachsende Zahl an neuen Sensorknoten stellt dabei eine große Herausforderung dar. Veränderte Hardware, neue Kommunikationsprotokolle oder unterschiedliche Frequenzbänder können zur Entstehung von heterogenen Sensornetzen führen. Die Ausführung der gleichen Anwendung und ihre Kompatibilität zu alten Sensorknoten ist schwierig zu gewährleisten. Durch die Verwendung von mehreren Radiomodulen (Multiradio) und unterschiedlichen Kommunikationsprotokollen auf einem Sensorknoten, können Gateways die Kommunikation in heterogenen Sensornetzen ermöglichen und somit eine Brücke zwischen Teilsystemen bilden.

Heterogene Sensornetze können auch durch die Verwendung unterschiedlicher Betriebssysteme entstehen. Die in [1] vorgestellte *Cross-Plattform* (CP) zeigt einen Lösungsansatz für dieses Problem durch eine Abstraktion von Betriebssystemfunktionen auf.

Während Interoperabilität in *Low-Power Lossy Networks* zwischen IPv6-Implementierungen in verschiedenen Betriebsystemen grundsätzlich möglich ist, geht dies zu Lasten der Übertragungsleistung und des Energieverbrauchs [2]. Des Weiteren erreichen IoT-Lösungen mit IPv6 im Batteriebetrieb derzeit nur eine Laufzeit in der Größenordnung von Tagen [3]. Die in Sensornetzwerken angestrebten Laufzeiten von Jahren sind wegen des im Vergleich mit anderen Komponenten hohen Energiebedarfs des Radios derzeit nur mit *Duty Cycling* möglich. Bei bestehenden Low-Power-Wireless-Protokollen wurde dies in der Regel nicht berücksichtigt [3]. Auf dem Betriebssystem aufsetzend würde ein Netzwerkstack in der Middleware höchstmögliche Interoperabilität gewährleisten und könnte dennoch nach Belieben angepasst werden.

In diesem Paper wird die Multiradiounterstützung der CP vorgestellt, die es ermöglicht, heterogene Sensornetze unter Verwendung unterschiedlicher Betriebssysteme zu betreiben. Zudem werden die Grenzen der CP für den Einsatz von zeitschlitzbasierten Kommunikationsverfahren unter Verwendung von zwei einfachen MAC-Protokollen getestet.

II. Cross-Plattform und Netzwerkstack

Das in Abbildung 1 dargestellte Konzept der CP als Middleware sieht eine Kapselung von Betriebssystemfunktionen in OS-Wrappern vor, ohne das Betriebssystem dabei zu verändern. Einer Applikation wird der Zugriff auf diese Funktionen durch eine Schnittstelle (API) zur Verfügung gestellt. Ist ein Betriebsystem bereits portiert, entfällt der Portierungsaufwand für Anwendungen auf dessen unterstützte Plattformen. Die CP unterstützt die Betriebssysteme *Contiki* [4] und *langOS* [5], welche sich vor allem in der Anzahl der verwendbaren Funkmodule und in den Scheduler-Algorithmen unterscheiden. Vom Betriebssystem erzeugte Ereignisse werden in der CP in einem kooperativen Prozessmodell abgearbeitet.

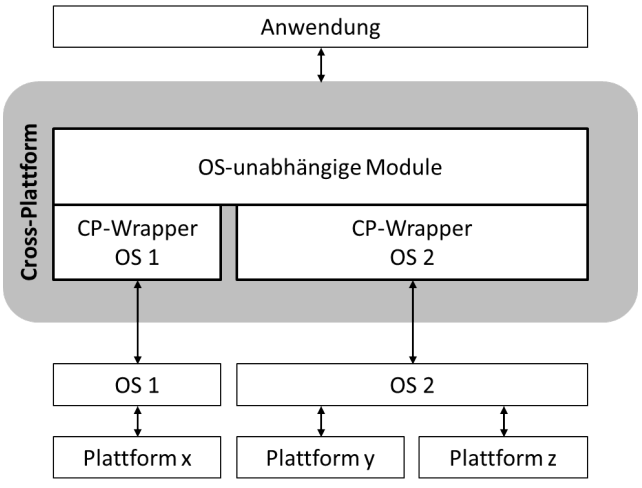


Abbildung 1. CP-Anwendungen sind in die Schichten Anwendung, CP, Betriebssystem und Hardware aufgeteilt.

Die CP-Schicht selbst ist dabei, wie in Abbildung 2 gezeigt, in Modulgruppen für Hardware-, Sensor-, Service- und

Netzwerkmodule unterteilt. Jedes Modul kann mehrfach instanziiert werden, dies erleichtert die Verwendung von Netzwerkmodulen auf einem Multiradiosystem.

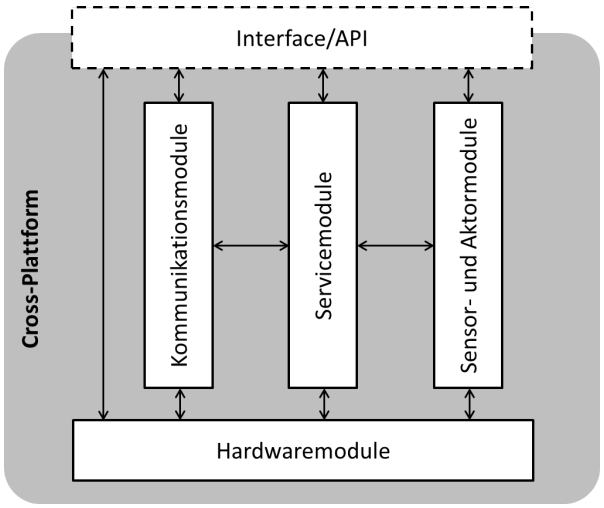


Abbildung 2. Interner Aufbau der CP. Die Anwendung greift über eine einheitliche API auf die in funktionale Gruppen aufgeteilten Module zu.

Der Aufbau des Netzwerkstacks ist vom *Open Systems Interconnection-Modell* abgeleitet. Die Module des Netzwerkstacks setzen gemeinsame Interfaces um, wodurch die Austauschbarkeit von Modulen innerhalb einer Netzwerkschicht oder Protokollfamilie ermöglicht wird. Als Beispiel für Medienzugriffsverfahren sind *Aloha* [6] und *Slotted Aloha* [7] in der CP implementiert.

III. Umsetzung und Evaluation

Um das Konzept der Multiradiokommunikation und die Eignung für zeitkritische Kommunikationsprotokolle in der CP zu evaluieren, wurden zwei Versuchsreihen mit unterschiedlichen Aufbauten durchgeführt.

A. Multiradio

Die Multiradiounterstützung der CP wurde mit der in Abbildung 3 dargestellten Testanwendung untersucht, in der zwei verschiedene Teilnetze durch ein Gateway miteinander verbunden wurden. In den Teilnetzen operieren, unter Verwendung der CP, der *FeuerWhere-Knoten* (A) mit langOS im 868 MHz Frequenzband und ein *TMote Sky-Knoten* (C) mit Contiki im 2,4 GHz Frequenzband. Der *FeuerWhere-Knoten* (B) fungierte als Gateway, da langOS die Verwendung von mehreren Radiomodulen unterstützt. Als Kommunikationsprotokoll zwischen (A) und (B) wurde Slotted Aloha eingesetzt, Knoten (B) und (C) kommunizierten unter Verwendung von Aloha. Die in der CP implementierte Testanwendung ließ ein Paket zwischen allen Knoten zirkulieren.

Die Implementierung des Programms für den Gateway-Knoten in der CP umfasst 54 Zeilen Quelltext. Das kompilierte Programm hat eine Gesamtgröße von 49270 Byte. Davon werden 266 Byte (0,5 %) für die Anwendung, 4536 Byte für Slotted Aloha (9,2 %) und 2226 Byte für Aloha (4,5 %) verwendet.

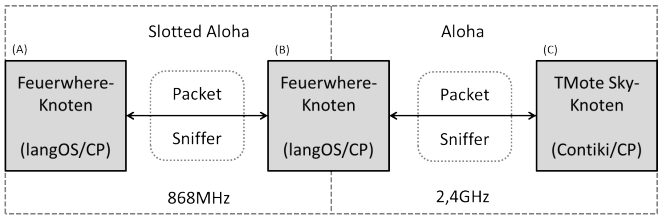


Abbildung 3. Testanwendung für ein heterogenes Netzwerk mit Multiradio-Gateway

B. Zeitschlitzbasierte Protokolle in der CP

Die Testanwendung zur Untersuchung des Durchsatzes auf Grundlage zeitschlitzbasierter Protokolle (TDMA) wurde exemplarisch durch das Kommunikationsprotokoll Slotted Aloha evaluiert. Die in Abbildung 4 dargestellten Versuchsreihen umfassten jeweils zwei Sensorknoten und einen Paket-Sniffer. Dabei kommunizierten, unter Verwendung der CP, zwei FeuerWhere-Knoten mit langOS im 2,4 GHz bzw. 868 MHz Frequenzband und ein TMote Sky-Knoten mit einem FeuerWhere-Knoten im 2,4 GHz Frequenzband.

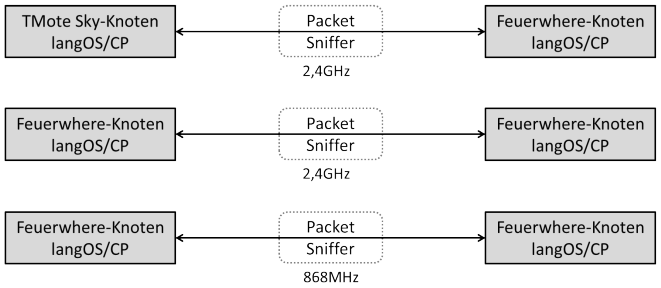


Abbildung 4. Testanwendung: TMote Sky- und FeuerWhere-Knoten mit langOS, im 2,4 GHz/868 MHz Frequenzband

Die Paketgröße bei 868 MHz beträgt insgesamt 98 Byte. Das Funkmodul wurde mit einer Datenrate von 37,5 kbit/s betrieben. Somit ergibt sich für Slotted Aloha eine rechnerische Zeitschlitzlänge von 20,43 ms und ein maximaler Durchsatz von 48,9 Paketen pro Sekunde. Wird Slotted Aloha auf einem 2,4 GHz Funkmodul mit einer Datenrate von 244 kbit/s verwendet, ergibt sich bei einer Gesamtpaketlänge von 99 Byte ein maximaler Durchsatz von 315 Paketen pro Sekunde. Die Resultate der tatsächlich ermittelten Zeitschlitzlängen durch die in der CP implementierten Kommunikationsprotokolle sind in Tabelle I angegeben.

Tabelle I
Durchsatz (Pakete pro Sekunde) in Bezug auf Zeitschlitzlänge in der CP

Zeitschlitzlänge (ms)	125	100	75	50	25
Durchsatz (p/s)	8	10	13	20	40

Bei 868 MHz wurde mit der Zeitschlitzlänge von 25 ms und 40 Paketen pro Sekunde, die theoretische minimale Zeitschlitzlänge und maximale Paketrate fast erreicht. Zudem konnte beobachtet werden, dass die minimale Zeitschlitzlänge,

bei Verwendung des TMote Sky-Knotens, auf 50 ms begrenzt war.

IV. Fazit

Es ist möglich mit der CP zeitschlitzbasierte Protokolle zu implementieren, deren Einsatz sollte dabei in Abhängigkeit vom Datendurchsatz und der verwendeten Frequenz abgewogen werden. Unter Anwendung verschiedener Medienzugriffsprotokolle ist die einfache Implementierung einer Multiradioanwendung in der CP möglich. Der modulare Netzwerkstack soll in Zukunft durch zusätzliche Protokolle auf verschiedenen Ebenen erweitert werden. Zudem sollen die Möglichkeiten der Datenakquise durch Sensor-, Netzwerk- und Servicemodule ausgebaut werden.

V. Danksagung

Diese Arbeit ist Teil des DIAMANT-Projekts am IHP und wurde durch das *Bundesministerium für Bildung und Forschung* unter der Referenznummer *03IPT601X* gefördert.

Literatur

[1] M. Frohberg, P. Poppe, N. Vetter, and M. Schölzel, "Cross-Plattform zur hardware- und betriebssystem-unabhängigen Implementierung von Anwendungen und Protokollen," *15. GI / ITG Fachgespräch Sensornetze*, 2016.

[2] J. Ko, J. Eriksson, N. Tsiftes, S. Dawson-Haggerty, A. Terzis, A. Dunkels, and D. Culler, "ContikiRPL and TinyRPL: Happy Together," in *Proceedings of the workshop on Extending the Internet to Low power and Lossy Networks (IP+SN 2011)*, (Chicago, IL, USA), 4 2011.

[3] A. Dunkels, J. Eriksson, and N. Tsiftes, "Low-power Interoperability for the IPv6-based Internet of Things," in *Proceedings of the 10th Scandinavian Workshop on Wireless Ad-hoc Networks (ADHOC 11)*, (Johannesberg Castle, Stockholm), 5 2011.

[4] A. Dunkels and O. Schmidt, "Protothreads: Lightweight, stackless threads in c," SICS technical report T2005:05, Swedish Institute of Computer Science, 2005.

[5] O. Stecklina, A. Krumholz, and S. Kornemann, "langOS - A Low Power Application-specific Configurable Operating System," *13. GI / ITG Fachgespräch Sensornetze*, pp. 9–12, 2014.

[6] N. Abramson, "The aloha system: Another alternative for computer communications," in *Proceedings of the November 17-19, 1970, Fall Joint Computer Conference*, AFIPS '70 (Fall), (New York, NY, USA), pp. 281–285, ACM, 1970.

[7] L. G. Roberts, "Aloha packet system with and without slots and capture," *SIGCOMM Comput. Commun. Rev.*, vol. 5, pp. 28–42, Apr. 1975.

About Deployment Limitations of LoRa Gateways

Albert Pötsch
Linz Center of Mechatronics GmbH
4040 Linz, Austria
Email: albert.poetsch@lcm.at

Florian Haslhofer
Institute for Communications Engineering
and RF-Systems
Johannes Kepler University, 4040 Linz, Austria

***Abstract*—In this work, we discuss the upcoming network stack LoRaWAN and its underlying modulation technology LoRa, which is an implementation of a LPWAN (Low Power Wide Area Network) and consider practical limitations that can occur when restricted Machine-to-Machine (M2M) data agreements for the connection of LoRa gateways to a cellular network are involved. We illustrate a LoRa-transceiver's contrary behaviour of energy consumption and range versus the achievable data rate and analyze the packet overhead introduced by the LoRa gateway to create awareness on the importance of the amount of generated backhaul network traffic.**

I. Motivation and Introduction

The Internet of Things (IoT) is an upcoming topic in our society. Daily used things such as washing machines, streetlamps or public dustbins in a city will be connected to the Internet and will send their status (e.g. completed washing process or full dustbin) into a cloud, indicating to other things (machines, robots etc.) or humans that some action is required. Therefore, wireless networks, which can serve a huge amount of clients, are required. In smart homes, the most common short range communication technologies for connecting things are WiFi, ZigBee or Bluetooth. In contrast, using these radio standards would not be appropriate in a smart city because of their short communication range or a too high energy consumption. For this area of application Low Power Wide Area Network (LPWAN) technologies such as LoRa [1], SigFox [2] and Weightless [3] have been developed. They all operate in unlicensed ISM bands at sub-GHz-frequencies typically at 433, 868 or 915 MHz.

LoRa, which stands for *Long Range*, is a long-range wireless communications system, promoted by the LoRa Alliance. **LoRaWAN** (a trademark of the LoRa Alliance) involves a protocol stack with the LoRa wireless technology as the physical layer. Within LoRaWAN a single sensor devices communicates over the air with (multiple) gateways which relay the data messages to a central network server as can seen in Fig. 1. While the LoRa modulation is proprietary, the LoRaWAN is an open standard being developed by the LoRa Alliance.

This work focus on the usability of LoRa networks that depend on gateways that are connected to the cloud using cellular networks as IP-backhaul-interface and which are operated with small and cost-effective Machine-to-Machine (M2M) rates for the network connection. Typical M2M contracts include data-volumes of less than 100 MB per month [4,5,6], so the amount of data-overhead produced by a LoRa-gateway can be a relevant issue, especially for large number of appliances when e.g. (municipal) service providers require thousands of SIM cards to operate those gateways.

[1] www.lora-alliance.org
[2] www.sigfox.com
[3] www.weightless.org

II. LoRa Basics

The LoRa LPWAN solution includes two major components. The first one, describing the physical layer, is the LoRa modulation, which is a CSS (Chirp Spread Spectrum) frequency modulated technology developed by company Semtech that uses wideband linear frequency modulated pulses whose frequency increases or decreases based on the encoded information. The main advantages of this approach are twofold: a substantial increase in receiver sensitivity due to the processing gain of the spread spectrum technique and a high tolerance to frequency misalignment between receiver and transmitter [7]. The second component is the LoRaWAN network protocol, described in more detail in chapter III.

In LoRa modulation the spreading of the spectrum is achieved by generating a chirp signal that continuously varies in frequency. The frequency bandwidth of this chirp is equivalent to the spectral bandwidth of the signal. The wanted data signal is chipped at a higher data rate and modulated onto the chirp signal. Several parameters are available for the customization of the LoRa modulation: Bandwidth (BW), Spreading Factor (SF) and Code Rate (CR). The SF describes how many bits per symbol are encoded. Its number ranges from SF6 (64 chips/symbol) to SF12 (4096 chips/symbol). Two signals with different spreading factors are orthogonal.

By properly setting SF and BW, the time on air T_{packet} (or packet duration) can be reduced or the range of the transmission can be optimized. According to the LoRa Modem Designer's Guide [7] T_{packet} consist of $T_{preamble}$ plus $T_{payload}$, both are based on the symbol duration T_{sym} given by (1).

$$T_{sym} = \frac{2^{SF}}{BW} \tag{1}$$

[4] www.embeddedworks.net/m2m-data.html
[5] www.m2m-mobil.de
[6] www.simfonymobile.com/global-mobile-connectivity
[7] www.semtech.com/images/datasheet/LoraDesignGuide_STD.pdf

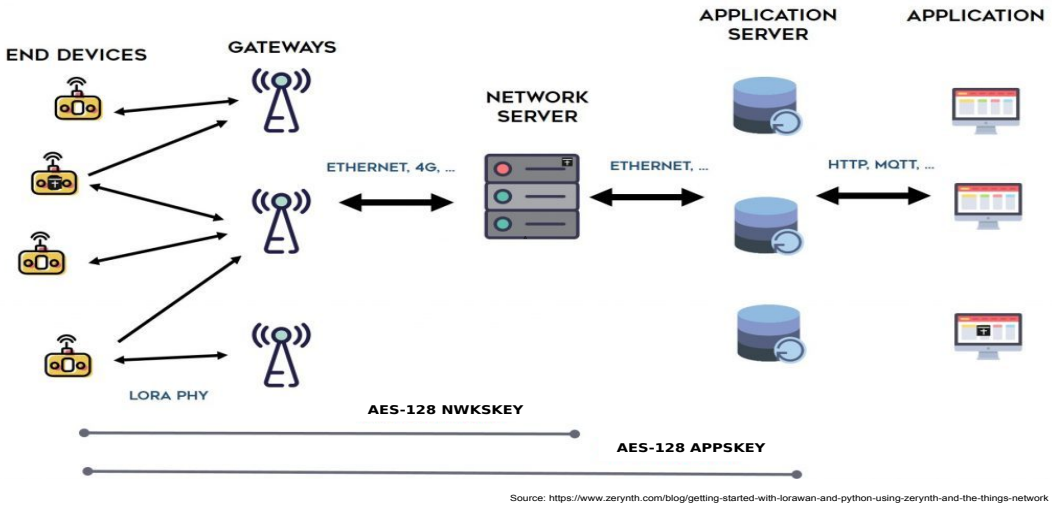


Fig. 1. Structure of a global LoRaWAN network

LoRa modulation includes a variable error correction scheme that improves the robustness of the transmitted signal at the expense of redundancy. Introducing the code rate CR, the relationship between the effective data bit rate R_b , BW and SF for LoRa modulation can be expressed using Equation (2), further details can be found in the LoRa Modulation Basics Application Note [8]. For a given BW of e.g. 125 kHz and CR set to $\frac{4}{5}$ datarates from 0.29 kbit/s (SF 12) up to 5.47 kbit/s (SF 7) can be obtained.

$$R_b = SF \frac{BW}{2^{SF}} CR \tag{2}$$

From (1) we see that the larger the SF and the smaller the BW is, the longer a transmission lasts. A longer transmission time obviously results in more consumed energy. But, as stated in the datasheet of Semtech's gateway-chip SX1301 [13], with longer transmission times also higher sensitivity of the receiver can be achieved, which results in a larger communication range. The data rates and sensitivities depend only on the modulation bandwidth and the spreading factor but not on the carrier frequency.

As a consequence, for the lowest energy consumption and a high data-rate the spreading factor should be as small as possible, whereas the bandwidth should be as big as possible. If the major concern is a long communication range between LoRa end-devices and the gateways, the data rate must be lowered to increase RF sensitivity of the receiver.

The public available documentation provided by SemTech does not reveal all the necessary information how the LoRa RF modulation format is setup, in fact the coding of a LoRa packet is closed source. Attempts to decode the LoRa RF modulation format are carried out by open communities and other companies [9,10].

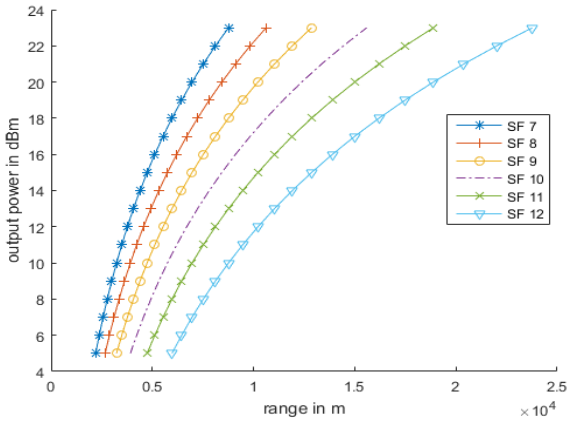


Fig. 2. Required output power for a required communication range of a LoRa link, depending on the spreading factor SF and assuming a path-loss exponent of 3 for urban area

For the communication range of LoRa different statements can be found reaching from multiple kilometers [11] to a maximum of 15 km [12]. To estimate the communication range d of LoRa we consider equation (3) for the path loss L, where f is the LoRa-frequency and c the speed of light. Assuming a path-loss exponent n (PLE) of 3 corresponds to signal-barriers in an urban area (the PLE can be set from 2 (free space) to 6 (high obstruction)). The link budget L_{Budget} for the transmission path is calculated by (4) with the receiver sensitivity $S_{RX}(SF, BW)$ and transmitted power P_{TX}.

$$L = 10 \log_{10} \left(\left(\frac{4\pi df}{c} \right)^2 d^n \right) \tag{3}$$

$$L_{Budget}(SF, BW) = P_{TX} - S_{RX}(SF, BW) \tag{4}$$

[8] www.semtech.com/images/datasheet/an1200.22.pdf
[9] https://revspace.nl/DecodingLora
[10] www.link-labs.com/blog/what-is-lora
[11] www.thethingsnetwork.org
[12] www.digikey.com/en/articles/techzone/2016/nov/lorawan-part-1-15-km-wireless-10-year-battery-life-iot

To obtain a theoretical estimate of the maximum possible communication range of a LoRa link depending on output power and spreading factor, we assumed there are no antenna gains to get a worst case result and set the path loss L equal to L_{Budget}. Now, if the path loss is equal to the link budget, then the receiver can just receive the signal. Values of $S_{RX}(SF, BW)$ are taken from the SX1301 LoRa concentrator/gateway transceiver datasheet [13].

According to Fig. 2 the theoretical maximum range can be achieved with SF 12 and highest transmission power (23 dBm). In an urban area with little obstructions, a gateway located 23 km away can - under ideal conditions - successfully receive the sent signal. This calculation confirms the range-information given by TheThingsNetwork [11] and DigiKey [12].

III. LoRaWAN

LoRa defines only the physical layer (PHY) (layer 1) of the Open Systems Interconnection (OSI) model. In contrast to the LoRa modulation LoRaWAN is an open LPWAN standard defining the medium access (MAC) and network (NWK) layer (OSI layer 2 and 3) and is maintained by the LoRa Alliance [1]. It is designed mainly for low data rate sensor networks, wherein sensors exchange packets with the network server. It also manages security, adaptive data rate (ADR) and de-duplicates the received data which is potentially received by multiple gateways.

A. Components of a LoRaWAN Network

Several components of the network are defined in the LoRaWAN specification and are required to form a LoRaWAN network: End-devices, gateways and the network server, as can seen in Fig. 1.

- **End-devices** are the low-power sensor nodes that communicate with gateways using LoRa-links.
- **Gateways** are the intermediate devices that forward packets coming from end-devices to a network server over an IP-based backhaul network allowing a higher throughput, such as Ethernet or cellular network standards. There can be multiple gateways in a LoRa deployment, and the same data packet can be received and forwarded by more than one gateway.
- The **Network server** is responsible for the management of the whole network. It knows about active nodes and when a new node (end-device) joins the network, it will ask the application-server if the node is allowed to join the network and if so, which settings to use for this node. It de-duplicates the received data which can possibly be received by multiple gateways, it authenticates this data (to make sure that these are no replay-attacks), it forwards this (encrypted) data to the application-server and it will ask the application-server if it should send anything back. Besides managing the data-flows, it also manages the state of the node through so called MAC-commands (e.g. to change the data-rate, channels etc.).

[13] www.semtech.com/images/datasheet/sx1301.pdf

Unlike traditional cellular networks, end-devices are not associated with a particular gateway in order to have access to the network. The gateways serve simply as a link layer relay and forward the packet received from the end-devices to the network server after adding information regarding the reception quality. Thus, an end-device is associated with a network server, which is responsible for detecting duplicate packets and choosing the appropriate gateway for sending back a reply (if any). Logically, gateways are transparent to the end-devices.

Three different LoRaWAN classes of end-devices are specified to address the various needs of applications:

- **Class A** end-devices can schedule an uplink transmission based on their own needs. This class of devices allows bi-directional communication, whereby each uplink transmission is followed by two short downlink receive windows. Downlink transmission from the server at any other time has to wait until the next uplink transmission occurs. Class A devices have the lowest power consumption, but also offer less flexibility on downlink transmissions.
- **Class B** end-devices open extra receive windows at scheduled times. A beacon from the gateway is required for synchronization, so that the network server is able to know when the end-device is listening.
- **Class C** end-devices have almost continuous receive windows and therefore have maximum power consumption.

LoRaWAN does not enable device-to-device communications, therefore packets can only be transmitted from an end-device to the network server, or vice-versa. Device-to-device communication can only be established by way of two gateway transmissions.

Since LoRa Class A end-devices uses the ALOHA protocol, there is no collision avoiding mechanism like Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) in WiFi. However, two signals can be received at the same time by a gateway under the condition that one signal is 6 dB stronger than the other one.

B. LoRaWAN software solutions

Several software solutions implement a LoRaWAN network stack, one example is the open-source *LoRa Server project* provided by *www.brocaar.com*. This implementation splits the architecture in three main components: The Gateway bridge, the LoRa (Network-)server and the Application-server as can be seen in Fig. 3. All these systems are required for a complete setup of a LoRa-network. The services can run on distributed computers as well as on the same machine, since its a pure software solution.

The Gateway Bridge is responsible for the communication with a gateway. It translates the UDP protocol from the gateway to a JSON message and publishes it to the MQTT implementation at the server. The entire data traffic has to go through the bridge. The application-server is the graphical user interface for configuring a gateway, creating new applications or editing the authentication mode for a node.

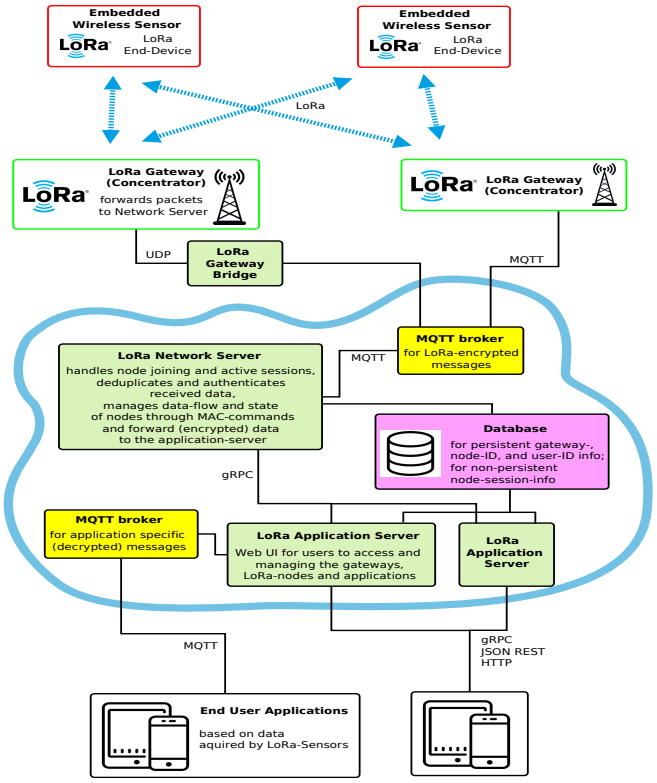


Fig. 3. LoRaWAN system architecture of the open source *LoRa Server project* provided by www.brocaar.com

C. Global IoT networks

Several communities strive for providing a world-wide solution for a common network for "things" by using LoRaWAN-enabled networks. Examples are companies like *Inmarsat* (Britain), *KPN* (Netherlands) or the open and crowd-sourced community-driven *TheThingsNetwork* [11]. A setup including a public server and several gateways all over the world makes it possible to build up an extensive global network. The basic idea is that the community or company provides the server and cloud services, whereas the gateways come from private people. Thereby a LoRa node from third parties can be easily attached (with credentials) to the 'big community'. It is also possible to run all services on the same device as the gateway does, to form an enclosed system, which is usually used for private LoRa networks with only one gateway.

IV. LoRaWAN packet overhead

For the reception of LoRa-modulated packets, a special concentrator chip, e.g. SX1301[13], is used. This chip can demodulate 8 LoRa signals, with different spreading factors, in parallel. The end devices (nodes) are connected in a star topology to the gateway. One gateway can handle thousands of LoRa nodes, though with increasing number of nodes the amount of data-collision will also rise as shown in [1]. The main task of the gateway (also called concentrator) is to convert data received by LoRa into a UDP packet. Therefore a JSON string, including the payload and other information about the gateway as well as the connection-quality will be

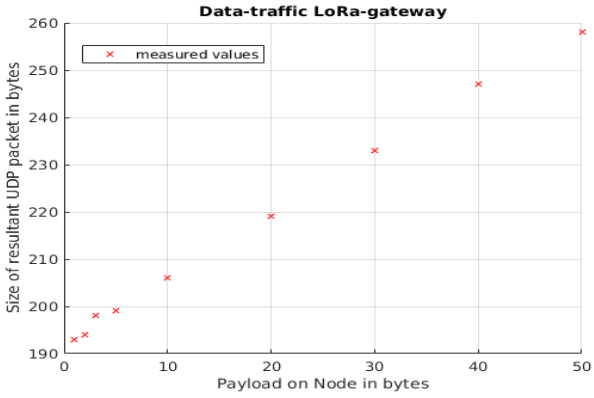


Fig. 4. Evaluation of data-overhead: UDP packet size generated by LoRa-gateway vs. LoRa-payload sent by end-device.

generated, introducing a significant data overhead. This string is transferred over a high-data-rate network, such as Ethernet, WiFi or Cellular. So the low-power concept of LoRaWAN ends at the gateway. In case of a cellular backhaul network for the gateways, the generated data volume can be highly relevant here, because common M2M contracts often include data volumes of less than 100 MB per month or are even billed per megabyte [4,5,6]. For IoT service providers operating a large amount of gateways it is essential to keep the transmitted data and in consequence also costs at a minimum.

From Fig. 4 we can see that the generated data-overhead of the gateway amounts to about 190 bytes per message.In case of sending a 10 byte payload every 5 minutes from a single sensor node, a data volume of about 1.7 MB per month is needed (status updates and heartbeats from the gateway are not included yet!). With the before mentioned data limit of 100 MB, a maximum of only 50 nodes per gateway can be served. This proves that the low-energy and low resources concept of LoRa includes only the connection between end-devices (sensor-nodes) and gateways but not the connection to the IP-based backhaul network.

V. Conclusion

In this paper we summarized the basics of the LoRa modulation and the open standard LoRaWAN that defines the communication protocol and system architecture for a Low Power Wide Area Network. By analyzing the packet overhead generated by the LoRa gateway we have shown that LoRaWAN realizes the low energy and low resources concept only between end-device and gateways. After the LoRa transmission to the gateway, there is no economical use of data.

Acknowledgment

This work has been supported by the Linz Center of Mechatronics in the framework of the Austrian COMET-K2 programme.

References

[1] O. Georgiou and U. Raza, "Low Power Wide Area Network Analysis: Can LoRa Scale?" *IEEE Wireless Communications Letters*, vol. 6, no. 2, pp. 162–165, April 2017.

Evolution of an Acoustic Modem for μAUVs

Jan Heitmann, Lucas Bublitz, Timo Kortbrae and Christian Renner
Research Group smartPORT
Hamburg University of Technology
{jheitmann, christian.renner}@tuhh.de

***Abstract*—The use of small autonomous underwater vehicles as mobile sensor networks is topic of many recent research projects. The swarm deployment of those robots enables a variety of flexible and low-cost use cases, such as port monitoring or marine research. For swarm interaction, an underwater communication device is needed. For this purpose, we developed an acoustic modem, which fits into miniature autonomous underwater vehicles, is cheap compared to the costs of a μAUV and offers reliable communication of up to $50\,\mathrm{m}$. During research with this modem, we discovered inadequacies, like low computing and insufficient transmission power. This paper presents an evolution of the modem, still beeing modular and low-cost, but having more computing power and an extended transmission range.**

I. Motivation

In recent years, many sensor network applications have been transferred into marine and especially into port areas. As an example, knowledge on the spatial extension of toxic pollution is important for the authorities to initiate countermeasures. Research projects like the *Clockwork Ocean* [1] measure finely-grained temperature gradients in sea water. Using a static sensor network for this purpose is inflexible, whilst using a single large mobile sensing device lowers coverage. Using inexpensive and very small autonomous underwater vehicles (μAUVs), however, offers large coverage and low cost at the same time. Recently, many small and cheap μAUVs have been developed, e.g. MONSUN [2], [3] and HippoCampus [4].

To enable swarm applications of these robots, robust underwater communication is needed. Communication devices already exist, but are either too large, too expensive or consume too much energy for this purpose. Devices that fit the needs are mostly closed source, and therefore do not allow the use in research on e.g. novel communication protocols or algorithms.

To face these shortcomings, we have developed a small, low-power and inexpensive underwater acoustic modem [5]. It was designed for integration in MONSUN and allows reliable communication at small distances of up to $50\,\mathrm{m}$. Unfortunately, it has several inadequacies: (a) Due to its form factor it cannot be easily integrated into other, possibly smaller, μAUVs like HippoCampus. (b) Its microcontroller is outmoded and slow, only allowing the software to run with some major deficiencies. Lacking computing power, sample rate and with that also the usable frequency band is limited. (c) The transmitter of the existing version only has a small transmission power, limiting the communication range in undisturbed conditions to roughly $50\,\mathrm{m}$ — or less, if the signal is disturbed by shallow water, reflecting surfaces or the growth of underwater plants. (d) Voltage regulators with a high quiescent current are used, resulting in low power-efficiency of $60\,\%$. (e) Due to incremental improvements and fixes, loose cables are used and signals are routed across the whole PCB, both negatively influencing the signal quality.

Taking all the downsides together, we concluded that a complete redesign of the modem was needed, which we present in this paper. In contrast to the existing version, our evolution is small enough to fit into many μAUVs, e.g. MONSUN, which is available for us due to a joint project with the UniversitÃd't zu LÃijbeck. We maintain the existing modular design by identifying three main components and integrating each of them on one PCB each: mainboard, receiver and transmitter. To increase the computing power, allowing higher sample rates and the implementation of advanced algorithms, e.g. for medium access, we decided to use a new and modern, yet low-power processor family.

The output voltage of the transmitter is also increased to twice of the old value. This enables communicating within a higher range allowing the inspection of a whole port area by a big swarm of robots. Lastly, the efficiency of the modem is improved by integrating power down modes and more efficient voltage regulators. The details of the new evolution of this modem are presented in this paper.

II. System Design

In this section, the main design decisions for the new version are presented for every independent part of the modem.

A. Fundamentals

As the modular design of the existing modem was useful for quick and easy changes and improvements, the revised version should still be modular, yet reducing the number of parts and connectors. Regarding this, we decided to split up the modem into three main parts: mainboard, receiver and transmitter. The mainboard holds the main processor, its power supply and communication interfaces to the computing unit of the robot. The receiver consists of amplifiers, an analog filter circuitry and an analog-digital converter. The transmitter contains a digital-analog-converter, a reconstruction filter and a final amplifier. All three parts of the modem are designed on a circular PCB with a diameter of $52\,\mathrm{mm}$. The circular shape allows an easy integration into MONSUN and many other μAUVs, as most of them are circular shaped, too. Our modem is still a low-cost device, the priced bill of material

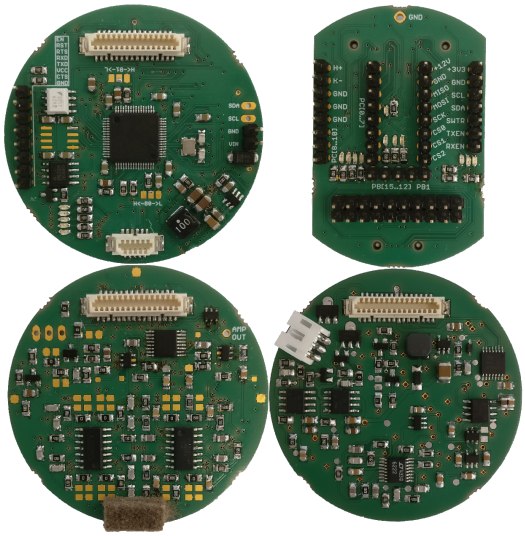


Fig. 1: The new evolution of our underwater modem. From upper left to lower right: mainboard, debug board, receiver, transmitter. The diameter of the boards is 52 mm. All boards stacked together have a height of around 30 mm.

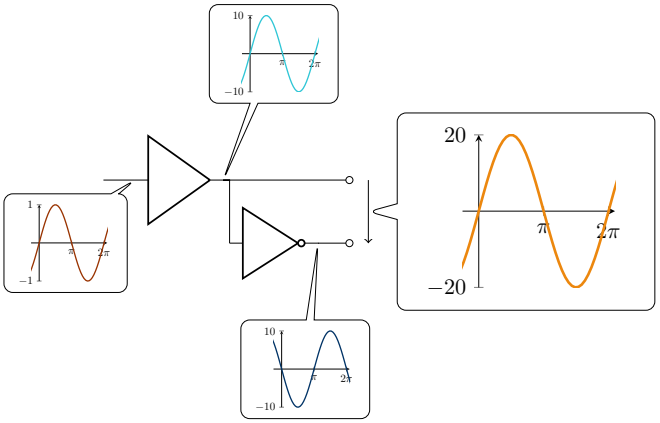


Fig. 2: Architectural overview of a bridge amplifier, with an exemplary gain at the amplifier of 20 dB.

(excluding hydrophone) is € 150 each, if at least ten pieces are built.

B. Mainboard

The existing version of the mainboard was powered by two voltage regulators with a high quiescent current of up to 7 mA each, resulting in an efficiency of only 60 %. The AVR32 processor with its maximum frequency of 66 MHz is at its performance limit. Furthermore, the analog-digital (ADC) and digital-analog-converters (DAC) were integrated on the mainboard leading to long analog signal paths.

The new version of the modem mainboard is based on an STM32F446RE [6] microcontroller. It has a Cortex M4 core with a maximum clock frequency of 180 MHz. With its flash memory of 512 kB, it provides sufficient storage for the software-based implementation of the modem. We chose the 64-pin version of the microcontroller, as it is easily solderable by hand, still providing all needed peripherals like I^2C, SPI, UART and enough GPIOs. For improved accuracy, neither the onboard DAC nor ADC is used. A TPS54202 step-down converter at 3.3 V is used as power supply. It has a low quiescent current of 45 µA. To protect the modem from voltage spikes possibly generated by the μAUV, all external signals are electrically isolated by MOCD207 optocouplers and ADUM1202 isolating ICs. In addition to isolation, the ADUM1202 also provides level translation, enabling a flexible use of the modem in μAUVs with different voltage levels.

C. Transmitter

The existing transmitter had an amplitude of 15 V at the output and the transmission range was limited to roughly 50 m. As the output power is quadratically correlated to the output voltage, we had to increase the amplitude of the output to achieve a higher transmission range. To increase the amplitude, a negative voltage power supply is needed, capable of generating the maximum peak power of the transmitter. With a maximum frequency of 75 kHz and a nominal hydrophone capacitance of 6 nF, a power of 4.5 W is needed for an amplitude of 40 V. These power supplies are either expensive, cannot deliver the needed current or have a large footprint. A bridge amplifier solves this problem. Only a positive voltage has to be generated which is fed into an inverter. Placing the load between the non-inverted output and the inverted output gives twice the voltage generated, resulting in a higher transmission power. The architecture of such a bridge amplifier is shown in Fig. 2. Our bridge amplifier is built with two OPA55x [7] supplied by a voltage of up to 40 V, limiting the voltage amplitude to around 70 V as the amplifier do not allow a rail-to-rail output. For low-power applications or short ranges the transmission power can be adjusted. An LT6222 quadruple amplifier is used for two purposes: Three amplifiers are used as a 6th-order reconstruction filter, low-passing the signal generated by the DAC. The remaining amplifier is used as a pre-amplifier, generating the input signal to the power stage. The amplification factor can be adjusted using a digital potentiometer.

D. Receiver

The receiving part of the existing modem worked well in real world tests, so the basic architecture was kept for this second modem revision. The receiver still consists of multiple stages: a pre-amplifier, an 8th-order high- and lowpass-filter and a power amplifier. To adjust the gain, a multiplexer was used to switch an array of resistors on or off. This approach occupied nine GPIO pins of the microcontroller and the single steps of adjustment were coarse. Approaching these two drawbacks, a digital controllable potentiometer, the AD5142A, is used to adjust the gain in the new version. It is controllable via the I^2C bus, using only two GPIO pins. These pins can also be used to communicate with other I^2C devices, for example to adjust the gain of the transmitting part, too. The AD5142A has

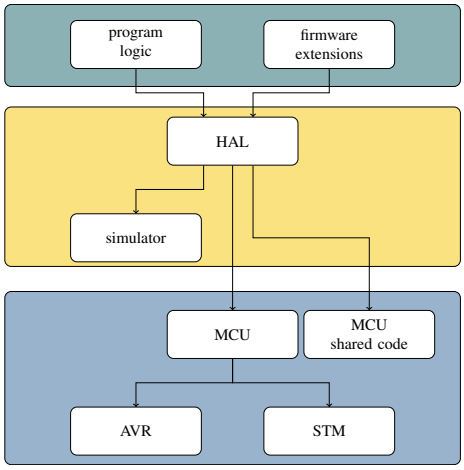


Fig. 3: Architectural overview of the modem software

a resolution of 8 bits, allowing the gain to be adjusted from a minimum value of 46 dB up to a maximum value of 104 dB. It is adjustable in steps of at least 2 dB over the complete range.

E. Software

The software for the modem contains many hardware specific parts, like interfacing with the peripherals. No separation between functionality and hardware existed, so the same software used for the AVR could not be used with the STM microcontroller. Since the already existing modems should still be used for student projects or field tests, a software compatibility with the revised version was needed. The hardware-dependent parts have to be encapsulated from the hardware-independent parts. To achieve this goal, all hardware specific functions have been extracted from the original software into a hardware-abstraction-layer (HAL). Its architecture is illustrated in Fig. 3. The HAL provides functions like sending and receiving data over UART or I^2C and is transparent to the user. Timers and communication with ADC and DAC are also abstracted, allowing to easily change hardware parts without changing the program logic. In addition to that, there is also a simulator, which allows to run the program logic on any computer without the need for having the actual hardware available.

III. Evaluation

This section evaluates the performance of the newly designed parts of the modem and compares them to the existing version regarding frequency gain, computing power and energy efficiency.

A. Mainboard

To compare the performance between the AVR microcontroller and the STM microcontroller, a firmware version with the same functionality has been flashed to both of them. A GPIO is pulled low when the processor is idle. At the beginning of every interrupt, the same GPIO pin is pulled high again, indicating that the processor is active. Using this method, the reception of an arbitrary 16 B packet has been recorded. The AVR is idle at 47.2 % of the complete time of reception. The STM improves this value and is idle at 60.6 % of the reception. The time gained can be used to activate a sleep mode or to use a higher sampling rate. At a first glance the improvement looks not very remarkable. During a packet reception some tasks have to be done once and in hard real-time, like decoding the packets' header or preamble detection. This took 142 µs on the AVR and now only takes 38 µs on the STM processor.

The calculation of the maximum sample rate achievable by both microcontrollers is hard because of the non-deterministic occurrence of certain events and the time overhead caused by them. Therefore the maximum sample rate has been determined by repeatedly increasing until the system was not able to reliably receive packets. With the STM sample rates can be twice as high compared to the AVR.

The power consumption of the new mainboard has been analyzed by measuring the voltage across a 3.3 Ω shunt. The supply voltage was 7.4 V, which is the nominal voltage of the used 2-cell lipo battery. For testing purposes a 16 B long packet has been received, and the whole process of reception has been recorded. If no sleep modes are used, the mainboard and its power supply consumes around 210 mW. Entering the sleep mode while the sample queue is empty, results in an average power consumption of 144 mW. The previous mainboard has a power consumption of 280 mW.

B. Transmitter

To evaluate the output power of the transmitter, a sine waveform has been applied to the input of the amplification circuit. The frequency was swept in steps of 400 Hz from 400 Hz to 100 kHz and in steps of 10 kHz from 100 kHz to 500 kHz. The input amplitude was 1.5 V with an offset of 1.5 V, using almost the full range of the used DAC8830 at 3.3 V supply voltage. Figure 4 shows the output of up to 32 dB V. The increase of transmission power inevitably led to a higher power dissipation in the operational amplifiers. Due to the small size of the PCB and the ICs, the power amplifiers heated up to over 70 °C after a couple of seconds sending. This is still in the allowed temperature range stated in the datasheet, but led to glitches in the amplified signal. All our tests were done in free air, so the temperature rise will likely be more significant after integration into an µAUV. Changing the package of the amplifier or adding heatsinks can help to solve this problem.

C. Receiver

The power consumption of the receiving circuit has been measured in the same way as the mainboard with the same supply voltage of 7.4 V. We used a mainboard PCB without a mounted microcontroller allowing it to include efficiency of the voltage regulator in the measurement without measuring the microcontroller's current. The receiver consumes 10 mW when it is disabled. If all components of the receiving circuit are turned on, it consumes 120 mW in total. In addition to the power consumption, noise of the filter stages has also been

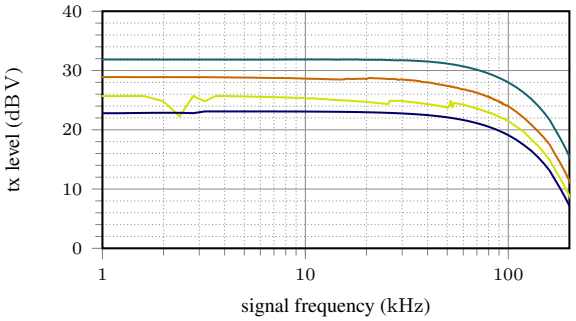


Fig. 4: Output transmission level at different gains

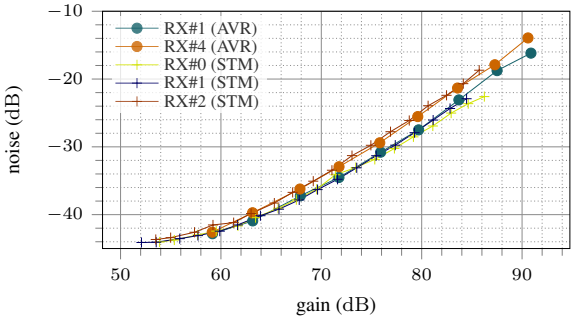


Fig. 5: Recorded noise at different gain settings of the receiver.

investigated and is shown in Fig. 5. As the hardware is almost the same as on the existing version of the receiver, the noise level has not changed significantly.

IV. Related Work

Several other devices have been developed during the last years. Most of the commercially available products are too expensive for integration into small and cheap AUVs. The Evologics S2C M HS [8] costs around €8000. The modems produced by Linkquest offer a high communication range of up to 1000 m with a simultaenously high data rate of up to 7000 bit/s. Beeing more than three times heavier than the HippoCampus, they are not suitable for an integration in μAUVs of that size. Another manufacturer is Teledyne Benthos, offering multiple devices with data rates of up to 15 kbit/s. Unfortunately, they are too big for an integration to μAUVs. In addition to size and cost issues, none of the commercial devices allows the access to lower layer components, making their use impractical when it comes to research of modulation schemes or MAC protocols.

In contrast to the commercially available off-the-shelf modems, multiple devices have been developed by different research groups around the world. The acoustic modem presented in [9] by Nowsheen et al. is a software-defined radio, offering a high data rate. However the demodulator is only available in MATLAB, making it unsuitable for integrating into real world robots. The WHOI micro modem [10] offers a low energy consumption in receive modem and a large transmission power. The data rate is up to 5400 bit/s, which would be sufficient for communication between swarm members. Unfortunately, the price is in the range of the commercial devices. In [11] a comprehensive comparison of several different acoustic underwater modems can be found. We are currently not aware of any modem that is as small as our modem is, simultaneously offering an adequate data rate and a price in the three-digit range.

V. Conclusion

In this paper we presented a revised version of our small and low-power underwater acoustic modem. In contrast to the existing version, the new version is much smaller, allowing a flexible installation in many μAUVs, such as MONSUN or HippoCampus. The microcontroller provides more computing power, making it either possible to increase the sampling rate or to activate sleep modes during active waiting times. Using sleep modes is not useful at all in the existing version due to the high quiescent current of the voltage regulators. This problem is now solved by using low-power switching supply sources. The modem software was completely restructured, making it easy to include or change new hardware parts in the future. Next, we will further analyze the issues we found during the building process. As stated in Section III the heat dissipation has to be further investigated. Furthermore, outdoor tests with the modem integrated in robots are on our agenda.

Acknowledgment

This work has been partially supported by the German Federal Ministry of Education and Research (BMBF) under grant number 13N14153.

References

[1] Helmholtz-Zentrum Geesthacht Centre for Materials and Coastal Research GmbH, “Clockwork Ocean.” [Online]. Available: http://www.uhrwerk-ozean.de/downloads/brochure_clockworkocean_overview.pdf

[2] B. Meyer, K. Ehlers, C. Isokeit, and E. Maehle, “The Development of the Modular Hard- and Software Architecture of the Autonomous Underwater Vehicle MONSUN,” in *ISR/Robotik 2014; 41st International Symposium on Robotics*, June 2014, pp. 1–6.

[3] C. Osterloh, T. Pionteck, and E. Maehle, “MONSUN II: A Small and Inexpensive AUV for Underwater Swarms,” in *ROBOTIK 2012; 7th German Conference on Robotics*, May 2012, pp. 1–6.

[4] A. Hackbarth, E. Kreuzer, and E. Solowjow, “HippoCampus: A Micro Underwater Vehicle for Swarm Applications,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[5] C. Renner and A. J. Golkowski, “Acoustic Modem for Micro AUVs: Design and Practical Evaluation,” in *Proceedings of the 11th ACM International Conference on Underwater Networks & Systems*, 2016.

[6] *STM32F446xC/E*, STMicroelectronics, 09 2016, "Rev. 6". [Online]. Available: http://www.st.com/resource/en/datasheet/stm32f446re.pdf

[7] *OPA55X High-Voltage, High-Current Operational Amplifiers*, Texas Instruments Incorporated, 1 2016, "Rev. B". [Online]. Available: http://www.ti.com/lit/ds/symlink/opa551.pdf

[8] Evologics GmbH, “Underwater acoustic modems.” [Online]. Available: https://www.evologics.de/en/products/acoustics/index.html

[9] N. Nowsheen, C. Benson, and M. Frater, “A High Data-Rate, Software-Defined Underwater Acoustic Modem,” in *Proceedings of the MTS/IEEE Oceans Conference*, Sydney, Australia, May 2010.

[10] L. Freitag, M. Grund, S. Singh, J. Partan, P. Koski, and K. Ball, “The WHOI Micro-Modem: An Acoustic Communications and Navigation System for Multiple Platforms,” in *Proceedings of OCEANS 2005 MTS/IEEE*.

[11] S. Sendra, J. Lloret, J. M. Jimenez, and L. Parra, “Underwater Acoustic Modems,” *IEEE Sensors*, vol. 16, no. 11, pp. 4063–4071, 2016.

Challenges for Sensor Network Based Outdoor Positioning in Forests - A Case Study

Silvia Krug and Jochen Seitz
Communication Networks Group, Technische Universität Ilmenau, Germany
{silvia.krug,jochen.seitz}@tu-ilmenau.de

***Abstract*—Wireless Sensor Networks are a well-known option to provide localization services in indoor environments. Besides that, they have been used to monitor environmental conditions or wildlife outdoors. However, there are several challenges to precisely locate moving objects in outdoor environments even if GPS is used as a sensor. Especially in forests, the tree canopy can partially block the satellite signals, leading to rather poor accuracy. In this paper, we present insights from a measurement campaign based on mobile sensor nodes equipped with GPS receivers. For better comparison, a set of predefined locations is used to evaluate the accuracy. The results show strong fluctuations in the localization quality.**

***Index Terms*—Wireless Sensor Networks; Performance Evaluation; Localization; Outdoor Scenarios.**

I. Introduction

Exact and precise localization of various objects or things in outdoor scenarios remains challenging even if satellite-based localization systems such as the Global Positioning System (GPS) are used. This is especially true, if single GPS receivers are used. The typical accuracy of such systems is within 10 m [1]. However, this typically holds for unobstructed line of sight conditions between the receiver and the satellites. For navigation purposes, this accuracy is sufficient as the 2D positions can be mapped to paths on the map and thus provide better localization results. If such a map is not available and the signal gets blocked partially by vegetation as it is the case in forests, the accuracy will decrease further.

Smart or precision agriculture and forestry are however two important use cases that require exact positioning. The same is true if mobile users or hand-carried equipment is to be tracked. One example for such a scenario is the surveillance of rescue dogs that are supposed to cover a search area and find missing persons. In this case, the documentation of full coverage requires the dogs to be equipped with positioning sensors that are ideally able to exchange their information amongst each other and the devices carried by dog handlers as well. This covers the documentation requirements and provides an ability to intervene in the field, if certain areas are not covered by the dogs. A similar case is the tracking of equipment parts that get deployed and have to be relocated later on. Again, the positioning information has to be gathered and send to a receiver that can be near by or further away. Therefore, a small battery-powered system that is able to define the location of the object it is attached to as well as to send this location to a defined destination is required. Due to the battery-powered device, expensive multi-sensor systems to obtain the positions are infeasible.

In this paper, we therefore present insights on the performance of GPS-based localization from a measurement campaign in an example forest area. The captured GPS data is part of a year round measurement campaign targeting the influence of vegetation on the radio communication in different forest areas.

II. Measurement Campaign Concept

The evaluation of the accuracy of GPS tracks is the side product of another measurement campaign that was performed in 2016 to evaluate the influence of outdoor conditions on WiFi communications [2]. The complete campaign consists of 19 measurement sets each located within a subset of 5 different areas. In each area, individual performance measurements were performed at a various number of measurement points with a distance of approximately 15 m between each other. The number of points varies, because for each set the maximum communication range is obtained, which is affected by the external conditions.

Fig. 1. Raspberry Pi based sensor node with Adafruit GPS module

We used a Raspberry Pi board that was extended with suitable WiFi USB dongles for the communication performance evaluation and a custom of the shelf GPS receiver module from Adafruit[1] for tracking the measurement locations. We

[1]https://learn.adafruit.com/adafruit-ultimate-gps/overview

used the recent version 3 of the Adafruit GPS module, which provides an accuracy of about 3 m according to the data sheet. This hardware is combined with an external power-pack that is usually used to charge smartphones and a waterproof casing that allows to fix all components securely and thus provide some shock protection. Figure 1 shows the node with open case.

In addition to the actual communication-related measurement, each node in the experiment was configured to log GPS positions once per second throughout the measurement set. Thus, the collected data represents the paths from and to the measurement area as well as the measurement points at which the nodes did not move for the duration of a measurement (90 s in this case). The log contains raw *gpsd* data logged via a shell script on the Raspberry Pi.

Since the goal of the original measurements was to gather insights on possible network performance variations under changing outdoor conditions, the measurement points were selected on a map and marked in the field. This ensures that the same points are used for subsequent measurements and provide an ideal base for an evaluation of the GPS accuracy since the hiking paths leading to the measurement areas and the measurement points themselves are known. The initial positions are obtained via an open access geo-information system provided by the Free State of Thuringia[2]. Figure 2 shows a screenshot of the target coordinates from one area, that will be used in the following.

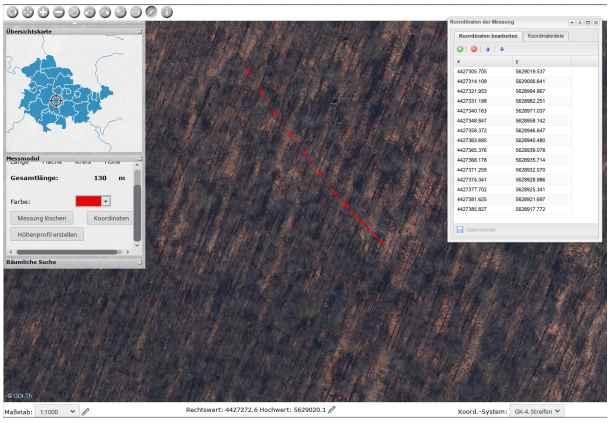


Fig. 2. Screenshot from the geo-spatial information system with target positions of the measurement area

The systems provides distance measurements as well as elevation information based on the geographic data. Besides that, the screenshot shows the vegetation in winter of the selected area, which mainly consists of old beech trees with thick undergrowth. Figure 3 shows the same positions plotted separately via R on a map of the area.

III. Results and Discussion

Figure 4 shows a visualization of the captured GPX tracks of two measurements (yellow and green) and in red the target

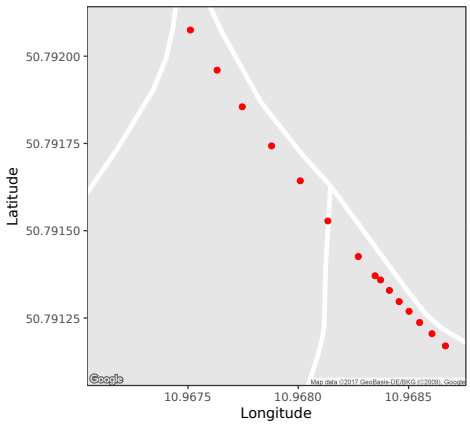


Fig. 3. Target positions of the measurement area in R

positions as a track in Viking[3]. The yellow track was obtained in April and the green one in November under similar weather conditions, except that the leaves of the trees were starting to develop in April and were already off in November. Further tracks of other measurement sets are also available, but the selected sets represent similar conditions and yet the most significant changes in the GPS accuracy, while the on-ground communication performance was similar in both cases. The other traces show a performance in between these extremes.

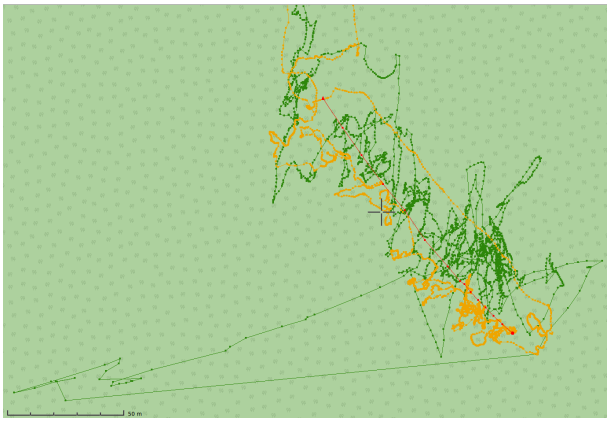
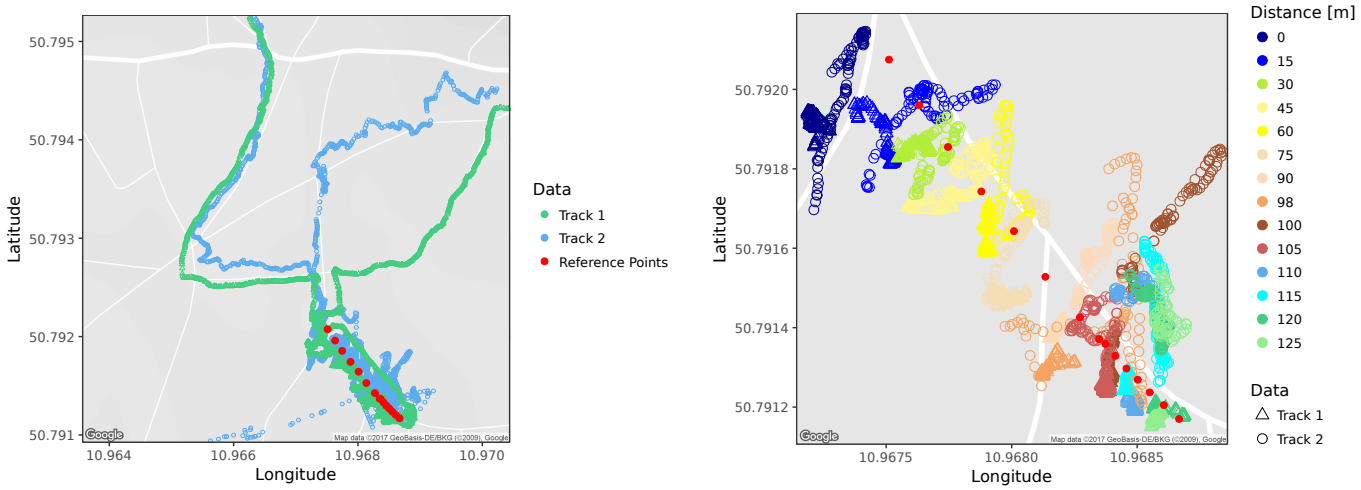


Fig. 4. Screenshot of two example GPX tracks in Viking

To compare the accuracy of the captured positions, we post processed the data in R. Figure 5 shows the corresponding positions for the measurement points and the accuracy when moving. Figure 5a includes a section in-between the measurement areas, where the nodes (pedestrians in this case) were moving at hiking speed along hiking paths, while Figure 5b shows the communication measurement area where the nodes only move in-between the different points and stay at a fixed predefined location otherwise. The movement in-between each measurement point has been filtered out in this analysis based on the timestamps of the performance measurements and those of the GPS positions.

[2]http://www.geoproxy.geoportal-th.de/geoclient/start_geoproxy.jsp

[3]https://sourceforge.net/projects/viking/



(a) Moving

(b) Fixed Positions

Fig. 5. Comparison of accuracy from two example GPX tracks in R

When comparing the tracks to the target positions or the actual hiking paths on the map, the first track 1 shows some deviations but still is close to the expected location. In contrast to this, the second track shows a worst case performance where the position approximation is rather poor. When closely examining these results, several outliers and maxima can be detected for both cases (e.g. in Figure 4). In the fixed position case, the maximum distance of deviations to the left and right of the target position is 200 m between points of the same fixed measurement points when the receiver is not moving. However, this can be seen as outliers because the average distance is around 30 m which is above the typical accuracy of ± 10 m that is assumed as maximum accuracy for single receiver setups [1].

Interestingly, a variation in the accuracy of a single track can be observed when evaluating the mobile scenarios in Figure 5a. On the left, both tracks stick quite close to the hiking path with a maximum distance of approximately 18 m. This accuracy remains stable until entering a side path to the right (on the left of Figure 5a). Afterwards, the deviation becomes bigger for with a certain offset to the top (blue) for track 2 while it becomes smaller for track 1, which stays more or less exactly on the path. While the green track returns to an initial accuracy of approximately 18 m later on, the blue one does not. It remains quite unstable with an average distance of 27 m and a maximum of about 60 m. The effect of changing accuracy could result from the increased coverage by vegetation (leaves and branches in this case) on smaller hiking paths that lead to increased shadowing and multipath propagation effects.

These results show that a single GPS receiver is not able to provide reliable positioning within forests. In case of movements along predefined paths, this could be mitigated to some extend by the mapping to the corresponding paths on a map by the navigation software [3]. If the movements are off-track in rough terrain as for example during search and rescue missions or other forestry related tasks, this option does not help.

Differential GPS with a special kind of receiver that sends its position information to a predefined anchor node, is another option to manage inaccuracies caused by atmospheric or orbital conditions. However, a battery-powered sensor node might not be able to perform the required communication for each update and the energy drain is quite high, if differential GPS is used. Besides that, common correction services are mostly distributed along coast lines to enable precise localization of sea vessels and thus might not be reachable in mountainous regions. Dual modules would also be an option if the energy budget of the node allows their integration. Further evaluations are needed to clarify this. Additional receiver modules are similar to the previous option and again the size and energy constraints of the finial system cause additional challenges if non-vehicular nodes are targeted.

Similarly, inertial sensors could be used to calculate relative positions based on a known initial position [4]. This relative position could be used to correct the GPS position. However, these sensors themselves are sensitive to errors and due to common integration procedures, these errors accumulate quite fast. Besides that, the continuous integration requires high computational effort that might not be available on resource-constraint sensor nodes. First experiments with a cheap inertial sensor and filtering failed on the Raspberry Pi version B due to the computation effort for the filtering and error accumulation. Other sensors such as compasses might however help to provide a fast and efficient filtering of impossible positions while moving (online).

Another option such as visual navigation (e.g. [5]) is also not applicable, because the required camera system is usually not possible for single sensor nodes. The same is true for anchor based localization systems that require some fixed infrastructure at known positions in the target area. Such

systems are infeasible for outdoor use, as the infrastructure will most likely not be available. These systems however do not use GPS but rather information on the received signal strength to acquire the positions of nodes (e.g. [6]). A system without the need for fine-grained infrastructure is presented in [7], where the authors propose to employ LoRa anchors for the positioning. However, blocking terrain effects especially in rough terrain with a changing landscape make this challenging.

These challenges require a different approach to provide accurate positioning under outdoor conditions that is aware of the constraints of the sensor platform. In [8], the authors show an interesting approach that allows multiple simple receivers to cooperate and thus determine relative positions with high accuracy. However, there are no modules available currently using that technique. The idea to provide accurate relative positions between individual nodes is actually a promising solution also for energy constraint devices, especially if few local nodes, e.g. at the edge of the forest, can be used as anchors in addition. At the same time, the ongoing communication of the sensor nodes could be exploited to piggy-back the required information exchange with the reference anchor nodes. In our search and rescue scenario, such a concept would allow cooperation leading to redundant positioning information shared among the individual nodes and at the same time allow the nodes to get reference positions from non-moving ”server” nodes.

IV. Conclusion and Future Work

In this paper, we presented insights on pure GPS accuracy of sensor nodes equipped with single GPS receivers in forest areas. The results show a strong deviation even if the environmental conditions are similar. Besides the 2D accuracy, this is even worse for the elevation data. Sensor nodes have been equipped with GPS or inertial sensors before. However, if the sensors are supposed to be deployed under the tree canopy and are potentially mobile, different solutions are required especially with respect to the energy consumption. Whether anchor nodes are possible in such a scenario will highly depend on the envisioned use case.

Based on the current results, we plan to further analyze the available data also with respect to sensor networks in precision forestry and the tracking of unmanned aerial vehicles, where precise 3D positions are required in addition. To achieve this position estimation, we plan to evaluate possible techniques for relative positioning in combination with few anchor nodes.

V. Acknowledgment

The authors thank Stefan Golisch and Christoph Gerhardt, who evaluated the inertial sensor performance on the Raspberry Pi during their student project.

References

[1] Garmin Ltd. (2017) What is GPS? [Online]. Available: http://www8.garmin.com/aboutGPS/

[2] S. Krug and J. Seitz, “Impact of Environmental Conditions on WiFi-based Outdoor Mesh Networks,” in *15. GI/ITG KuVS Fachgespräch Sensornetze (FGSN 2016)*, ser. Technical Reports, Augsburg, Germany, Sep. 2016, pp. 1–4.

[3] H. Aly and M. Youssef, “Dejavu: An Accurate Energy-Efficient Outdoor Localization System,” in *21st International Conference on Advances in Geographic Information Systems (SIGSPATIAL)*, 2013, pp. 154–163.

[4] W. Seo, S. Hwang, J. Park, and J.-M. Lee, “Precise outdoor localization with a GPS–INS integration system,” *Robotica*, vol. 31, pp. 371–379, 2013.

[5] M. Sharifi, X. Chen, and C. G. Pretty, “Experimental Study on Using Visual Odometry for Navigation in Outdoor GPS-Denied Environments,” in *12th International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, 2016, pp. 1–5.

[6] N. E. Agroudy, N. Joram, and F. Ellinger, “Low Power RSSI Outdoor Localization System,” in *12th Conference on Ph.D. Research in Microelectronics and Electronics (PRIME)*, 2016, pp. 1–4.

[7] S. Kim and J. Ko, “Low-complexity Outdoor Localization for Long-range, Low-power Radios,” in *14th Annual International Conference on Mobile Systems, Applications, and Services Computation (MobiSys)*, 2016, p. 44.

[8] W. Hedgecock, M. Maroti, A. Ledeczi, P. Volgyesi, and R. Banalagay, “Accurate Real-Time Relative Localization Using Single-Frequency GPS,” in *12th ACM Conference on Embedded Network Sensor Systems (SenSys)*, 2014, pp. 206–220.

Using Wireless Sensor Networks for Object Location and Monitoring

Frank Senf*, Silvia Krug*† and Tino Hutschenreuther*
*IMMS Institut für Mikroelektronik- und Mechatronik-Systeme gemeinnützige GmbH
Ehrenbergstraße 27, Ilmenau, Germany
{Frank.Senf, Tino.Hutschenreuther}@imms.de
†Communication Networks Group, Technische Universität Ilmenau, Germany
silvia.krug@tu-ilmenau.de

***Abstract*—The use of Wireless Sensor Networks (WSN) for monitoring objects is growing in nearly every field of daily life. In this paper we present ideas on how to use WSNs to monitor the relative location and the state of objects that are arranged in a linear order. Special focus lies on low-latency communication and energy efficiency. We demonstrate how to apply these ideas for surveillance of a cargo train. This paper is based on results of the *fast-realtime* collaborative project.**

I. Motivation

Novel trends such as the Internet of Things and cloud-based, centralized data processing results in the need to collect data from all kind of objects. This requires to equip such objects with sensors and to communicate the sensor data towards a processing unit, which can be challenging in specific areas. Adaption to exceptional environmental conditions, self-administrating networks and extended service intervals are among the key features that have to be considered.

As an example, in contrast to today's passenger trains an electrical connection between the wagons of a cargo train is still avoided. This is mainly because such connections are not suited to be used in rough environments with respect to reliability and maintenance. An option to allow monitoring various parameters of a freight car is to equip it with sensors and form a wireless network of sensor nodes. Besides the collection of sensor data, the fact of having a sensor node at each wagon allows to retrieve information regarding the ordering and the integrity of a cargo train.

This approach is not specific to cargo trains but can be easily adapted to monitor other systems that arrange objects in a linear topology.

II. Use Case: Monitoring a Cargo Train

As the use case that defines the requirements for the WSN we have selected to monitor a cargo train. Monitoring in this context covers several tasks as

- detecting the wagons that belong to the train
- determining the order of the wagons
- detecting loss of integrity (wagon loss)
- validating operating parameters, such as temperature

There will be no manual configuration in case a wagon is chained to a specific train. The nodes have to build an ad-hoc network autonomously. Each node has to join the network of the train it belongs to.

It is expected that the radio propagation in a cargo train environment is very limited. Explorative measurements on freight cars of a typical length of 10 meters have shown that a wireless node will only see 1 - 2 neighbor nodes at most. This complies with the simulation results presented in [6]. We force a linear topology of the network during automatic network configuration. Details on this will be explained in section V.

The radio nodes of the WSN are battery-powered. To achieve a lifetime of six years that is required to be compliant with cargo train regulations and certification, special considerations have to be taken into account to create an energy-efficient system. Among others, the use of a dedicated sensor for geo-positioning of a network node is to be avoided.

A cargo train can have more than 70 wagons and thus will create a deep linear network. For low-latency event indication a fast transport of messages is a main focus. All estimations and calculations are based on defining an upper limit of 100 wagons. Section VI discusses details on the communication latency in linear topologies.

A centralized component will act as the gateway between the WSN and the internet. Collected data will be forwarded to a cloud-based visualization and archiving service.

Beside detecting the wagon order and checking the integrity of the train the WSN has to monitor operating parameters of each wagon. Some parameters are common to all variants of wagons (such as battery voltage) where others are specific to a single variant (such as the temperature of a refrigerator wagon). We will not discuss this in detail in this paper but list this for comprehensiveness.

III. Related Work

Already about ten years ago a dissertation at the Technical University of Braunschweig introduced a concept for a radio-based monitoring system of cargo trains [1]. In 2015, the Robert Bosch GmbH presented a system for condition monitoring and data networking in rail freight transport [2]. In contrast to the approach presented here, these concepts follow a more centralized approach and, among others, use separate sensors (GPS) to determine the position of the cargo train wagons. The monitoring system that is described in this paper

strongly focuses on low-latency communication and energy efficiency.

IV. Network Architecture

To implement the use case as described in Section II we decided to use the 6LoWPAN technology as the network stack for the WSN. The physical layer (IEEE 802.15.4) is enhanced by the 6TiSCH medium access control. To create a first demonstrator we selected the OpenWSN platform [7] as one of the freely available implementations of these technologies (see VII). OpenWSN implements the routing of datagrams according to the RPL standard.

A. 6LoWPAN

IPv**6** over **Lo**w-Power **W**ireless **P**ersonal **A**rea **N**etwork (6LoWPAN) defines an efficient transport of IPv6 packets in small link layer frames. In contrast to other network architectures the native IP communication allows 6LoWPANs to be connected to IPv6 networks without the need of complex routers or gateways with corresponding translations.

B. 6TiSCH

IPv**6** over the **Ti**me**s**lotted **C**hannel **H**opping combines a set of protocols for configuration and operation of a time-slotted medium access in IPv6 networks. Time-scheduled access to the physical layer enables energy-efficient operation of the network node because it allows a node to enter a low-power mode during inactive time slots without the risk of missing transmissions. Using different radio channels for medium access (channel hopping) lowers the risk of collision and thus increases robustness and throughput of large networks.

C. RPL

The IPv6 **R**outing **P**rotocol for Low-Power and **L**ossy Networks (RPL) describes mechanisms to route datagrams within **L**ow-Power and **L**ossy **N**etworks (LLNs). The nodes (routers) in LLNs typically are constrained regarding processing power, memory and energy consumption and are interconnected by lossy links. Beside simple point-to-point traffic, LLNs often require point-to-multipoint and multipoint-to point traffic. RPL was specified according to these requirements. Although it might not be well-suited for deep linear topologies, we will use it for now as it is the only routing implementation that is available for the OpenWSN platform.

V. Topology Monitoring

A key functionality to implement topology monitoring for a WSN is the ability to continuously determine the positions of the nodes within the network. For energy-efficiency this needs to be realized without adding specific sensors (GPS or similar) but by using information that result from the network communication itself.

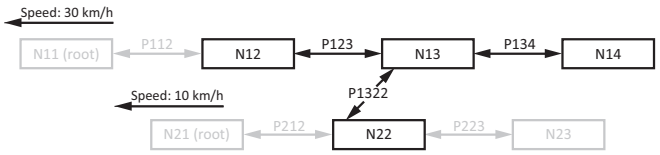
During network operation, the nodes receive datagrams from other nodes within the radio range. Each node adds the source addresses of these datagrams together with weighting parameters for the communication path to a table of known neighbor nodes. This table acts as the base for topology monitoring, which is implemented in a cloud-based service.

Forwarding all neighbor information of all nodes to the cloud service for processing will cause a very high network load, especially for the huge number of nodes that is expected for the use case. Therefore, it is required to implement at least a part of the topology monitoring decentralized at the network nodes. For this, each node needs to be able to autonomously find its position in the network.

As part of the WSN configuration, a node selects the neighbor node that it uses to forward directed datagrams (parent node). This selection has to consider the quality of the communication path to ensure a reliable data transport. By knowing the parent nodes of all nodes in the WSN, a centralized component is able to create a directed graph of the network. Thus, in our implementation, we decided to only propagate changes in the parent node selection to the monitoring service to reduce the network load.

A. Node Ordering

In our use case, the selection of parent nodes has to create a linear topology and only has to include nodes that belong to the same cargo train. This is achieved by adapting the implementation of the Objective Function (OF) of the RPL module. In the first implementation, the received signal strength indication (RSSI) is used to rate the data path to the neighbor nodes. Depending on the arrangement of the nodes, there will be 1 .. N neighbor nodes with the same data path quality. Figure 1 shows a part of a typical topology.



Communication Path Weighting:

	P123	P134	P1322
RSSI	-30 dB	-30 dB	-30 dB
Distance	10 m	10 m	10 m
Angle	0 °	180 °	70 °

Fig. 1. Topology Example.

If there are only nodes of one cargo train in the radio range, then all nodes but the first and the last one will detect two neighbor nodes with equal data path quality. In the topology example node N13 will detect the nodes N12 and N14 as parent candidates. To make the parent selection unique, the number of hops between the neighbor node and the root of the network is taken as a further rating parameter. Only such nodes that are closer to the root node are taken into account. Thus, node N13 will select node N12 as its parent. Doing so, a linear topology will form out, starting at the root node and sequentially adding the nodes in the order they are arranged.

The more complex situation of having nodes of another cargo train in radio range requires further considerations regarding parent selection. In the example node N13 will detect node N22 as another possible parent. Section V-C discusses possible solutions to overcome this.

B. Integrity Loss

Once a linear topology has formed out, another task of the monitoring system is to detect a loss of integrity, e.g. losing one or more wagons. For this, the nodes have to check its (virtual) connection periodically and have to communicate a disconnect event to the monitoring service if a change is detected. This has to be done by the parent node of the connection as the child node may fall out of radio range in case of a disconnection from the train. Here again, in a first implementation the RSSI is used to detect a loss of connection. This needs to be replaced by a better suited indicator, probably by one of the alternatives discussed in section V-C.

C. RSSI Alternatives

Because of the large number of possible confounding factors for the RSSI value when deploying the nodes on a cargo train it is expected that the RSSI value is not suited for reliable data path rating and detection of integrity loss in the field.

We are evaluating a system that allows to determine the distance between two radio nodes and the relative angle of a target node based on the radio signal itself (ranging). At the time this paper was written, the evaluation is still ongoing but first results are promising. Although the radio signal propagation in a train environment is very limited, test measurements have shown that a reliable monitoring of the position of the next wagon is possible. Knowing the distance and the angle of a target node will allow to select the nearest node towards the head of the train as the parent node. An increasing distance between local and target node will be an indicator for an integrity loss.

Another replacement for RSSI-based decisions is to measure and compare the speed the wagons are moving with. The process of forming out the train topology will start once the train starts moving only, due to energy saving requirements. Local and parent node continuously inform each other about the current speed. Only target nodes that are moving with the same speed are trusted as parent candidates. This will avoid to include nodes of another train. After the topology has formed out, a difference in the speed of parent and local node will indicate a loss of connection.

As a drawback, both alternatives require adding a further sensor to each wagon and will cause additional radio communication, which will increase power consumption.

VI. Challenges for Large Linear Topologies

When defining the monitoring system as described above, we were faced with some very specific challenges resulting from the fact of creating a large linear topology. These will be addressed in the following sections.

A. Datagram Latency

Datagrams that are generated by a specific node of the WSN traverses the network hop by hop from the originator to the target. At each hop, the datagram is delayed as the node has to wait for the time slot it has allocated for packet transmission to the parent node. The delay depends on the number of time slots (slotframe length), the duration of a single slot and the slot ordering (schedule). Table I lists the delays introduced by a single hop for a slot length of 10 milliseconds depending on the slotframe length.

TABLE I
Hop delay depending on slotframe length

Slotframe Length	Best Case	Typical	Worst Case
11	10 ms	50 ms	110 ms
101	10 ms	500 ms	1100 ms

When a datagram traverses a time-slotted network the delays introduced by the hops accumulate and result in the overall datagram latency. In [3] the authors introduce an optimized scheduling scheme that helps to minimize the latency, especially in linear topologies. In Table II the datagram latency of default and optimized schedules is compared for various numbers of hops. The latency calculation is based on a slot duration of 10 milliseconds and a sloframe length of 11 slots.

TABLE II
Typical Latency Depending on Schedule

Number of Hops	Default Schedule	Optimized Schedule
1	10 ms	10 ms
10	500 ms	100 ms
100	5000 ms	1000 ms

B. Repeating Nodes

Beside handling the datagrams generated locally, each node has to forward the datagrams it receives from its child nodes. This causes higher load (and thus energy consumption) for such nodes that are located closer to the root node. When assuming that during the maintenance interval each node will occupy each position in the topology for the same amount of time the average energy consumption can be assumed as equal for all nodes. Nevertheless, when defining the communication protocol an overall datagram sending scheme should be defined that avoids the above effect at least partially.

C. Downward Datagrams

The optimized scheduling shown in [3] only covers the case where a datagram is sent towards the root node of the network. Datagrams sent in the opposite direction (downwards) by default share a single slot in the slotframe. This results in a constant latency added for each slot.

Furthermore, the routing of downward datagrams can have a negative impact on latency. Datagrams that travel towards the root node are routed automatically by forwarding it to

the selected parent node. In contrast, the routing information for a datagram that is sent from the root node to a specific node of the network has to be stored either in the datagram headers or each node has to maintain routing information. Both methods will occupy significant resources. When including routing information in the datagram header the header space will grow depending on the number of hops the packet has to travel. This will reduce the payload space of the datagram and introduce datagram fragmentation for large topologies which will further increase the latency. If the communication protocol requires directed downward datagrams then this needs to be considered. The use of ”directed” multicast datagrams will be an alternative if an efficient mechanism for delivery of such packets in radio networks is found.

VII. Demonstrator: OpenWSN on OpenMote

For evaluation and visualization of the concept for the monitoring system we have created a demonstrator with five radio nodes that form the WSN. A host PC with a specific access point to the WSN collects the datagrams that are generated by the radio nodes and runs the monitoring service.

As the network stack implementation that runs on the nodes we chose the OpenWSN platform [7]. OpenWSN provides an open-source implementation of a complete protocol stack, including 6LoWPAN, 6TiSCH and RPL, and is already ported to a wide range of hardware platforms. Compared to alternative network stacks like TinyOS or Contiki the OpenWSN stack is a more lightweigth implementation that is compliant to the most recent releases of the protocol specifications.

The OpenWSN platform is extended by application modules that implement specific tasks of the monitoring system. The communication between the nodes and the monitoring service is based on the UDP protocol.

For the network nodes, we chose the OpenMote hardware module [8] which is based on the Texas Instruments CC2538 System on Chip (SoC). The powerful microcontroller, an on-chip radio transceiver and extended power-saving features make it a well-suited platform for evaluation purposes. During the initial test phase the nodes are powered by a standard battery with a capacity of 8500 mAh. The selection of a suitable battery for the cargo train use case will be possible after the overall communication scheme is defined and the optimization of the sleep phases of the nodes is finished.

VIII. Conclusion

In this paper we presented a system concept for localizing and monitoring objects based on a network of radio nodes. As key factors for successful realization of the concept we identified low-latency event communication and energy-efficient operation. For the challenges that result from these factors we presented possible solutions.

The implementation and validation of the concept in conjunction with a further decrease of the communication latency will be continued in the context of the *fast-realtime* collaborative project.

Acknowledgement

The work that is presented in this paper is supported by Projektträger Jülich, Forschungszentrum Jülich GmbH (PtJ), based on funding by the Federal Ministry of Education and Research (BMBF, funding number 03ZZ0504J) as part of the *Twenty20 Cooperation for Innovation* initiative (https://de.fast-zwanzig20.de/).

The authors would like to thank Michael Rink, Marco Götze and Marian Sauer from IMMS GmbH for the excellent cooperation while working on the *fast-realtime* project.

References

[1] Thorben Kupke, *Funkbasierte energieautarke Kommunikation für Eisenbahngüterzüge*, Dissertation an der Fakultät für Maschinenbau der Technischen Universität Carolo-Wilhelmina Braunschweig, 2007

[2] Bosch Media Service, *Bosch bringt Güterzüge ins Internet*, http://www.bosch-presse.de/pressportal/de/de/bosch-bringt-gueterzuege-ins-internet-42941.html, 2015

[3] Tengfei Chang, Thomas Watteyne, Qin Wang, Xavier Vilajosana, *LLSF: Low Latency Scheduling Function for 6TiSCH Networks* 2016 International Conference on Distributed Computing in Sensor Systems

[4] Dragan Vasiljević, Gordana Gardašević, *Performance Evaluation of OpenWSN Operating System on Open Mote Platform for Industrial IoT Applications*, 2016

[5] V. Sempere-Payá, J. Silvestre-Blanes, D. Todolí, M. Valls, S. Santonja *Signal Propagation in Metal-Rich Railway Environment*, 2016

[6] Bojana Nikolić, Bojan Dimitrijević, Tino Hutschenreuther and Hannes Töpfer, *Signal Propagation in Metal-Rich Railway Environment*, International Symposium on Theoretical Electrical Engineering (ISTET), Ilmenau, 2016

[7] *The OpenWSN project*, https://openwsn.atlassian.net/wiki/

[8] OpenMote Technologies, *OpenMote - Open Hardware for the Internet of Things*, http://www.openmote.com/

Posters & Demos

1. Albert Pötsch: *Indoor Air Quality Monitoring with LoRa-enabled Sensor Nodes*
2. Andreas Reinhardt: *TUCap: A Sensing System to Capture and Process Appliance Power Consumption in Smart Spaces*
3. Michel Rottleuthner, Thomas C. Schmidt: *Eine Testplattform für Energy Harvesting mit RIOT*
4. Thorben Schüthe, Hauke Pape, Karl-Ragmar Riemschneider and Klaus Jünemann: *Simulation und Auswertung von Permanentmagneten fur magnetoresistive Sensor-Arrays*
5. Tim Tiedemann, Christian Backe, Thomas Vögele and Peter Conradi: *Automotive Ad Hoc Sensor Networks in the Project SADA: Concept and Current State*