Leibniz Institute
for high
performance
microelectronics

20. GI/ITG KuVS Fachgespräch Sensornetze (FGSN 2023)

4th of September, 2023

Proceedings

Edited by

Krzysztof Piotrowski, IHP

DOI: 10.26127/BTUOpen-6637

Table of Content

Introduction

Quo Vadis Wireless Sensor Networks?

There are many names we use to call them, like the (original) Wireless Sensor Networks, Wireless Sensor and Actuator Networks, Internet of Things, Cyber-Physical Systems, Cyber-Physical Systems of Systems, and some others. More or less visible, wireless sensor networks are already applied in many aspects of our lives and for different purposes.

During the 20 years of Fachgespräch Sensornetze (FGSN) we were able to observe the process of the birth and evolution of wireless sensor networks. What do they look like now, from that time perspective? Is there still room for research and improvements? Or are they maybe already that mature that everything has already been said? And what do they look like from the industry point of view? What is the future of sensor networks? Where are they heading?

These retrospective and perspective views are the central topic of the 20th edition of the Fachgespräch Sensornetze (FGSN 2023) held on the 4th of September 2023 at Hasso-Plattner-Institut as part of the NetSys 2023 conference in Potsdam. We were happy to meet again, to discuss these subjects within the scientific community. The aim of this series of Fachgespräch is to give scientists from academia and industry the opportunity for an informal exchange of ideas and to strengthen cooperation in this multidisciplinary research area. All around the sensor networks. From different perspectives.

Krzysztof Piotrowski - Conference Chair
IHP - Leibniz Institut für innovative Mikroelektronik
Frankfurt (Oder), Germany
piotrowski@ihp-microelectronics.com

Internet of Things and Real Internet of Things

Reinhardt Karnapke
Head of Pre-Development Perinet GmbH
Berlin, Germany
reinhardt.karnapke@perinet.io
and
Distributed Systems/Operating Systems Group
Brandenburg University of Technology Cottbus-Senftenberg
Cottbus, Germany
karnapke@b-tu.de

Karsten Walther
Managing Director Perinet GmbH
Berlin, Germany
karsten.walther@perinet.io

***Abstract*—Sensor networks have come a long way since their first inception 30 years ago. In this paper we look at the results achieved and the areas that still need research from an industry perspective.**

***Index Terms*—Sensor networks; Wireless sensor networks; Industrial control**

I. Introduction

With the inception of Smart Dust a quarter of a century ago came the vision of hundreds or thousands of tiny sensor nodes that could be easily deployed (e.g. by throwing them out of a flying plane), that would organize themselves into a network autonomously and deliver data and insights to the operators automatically, even pre-filtered, pre-processed and ready to use [1], [2], [5]. Pretty soon actuators entered the picture and the visions now included new forms of self organization and self-x properties with the most common actuator being a motor which enables mobility. Now, swarms of robots needed to be coordinated (e.g. [4]). All of this culminated in Cyber-Physical Systems and the Internet of Things.

However, by now many of the topics are either solved or rejected. For example, let us take a look at autonomous wireless multi-hop self-organization. While this topic is very interesting from a research perspective, it is rarely useful from an Industrial Internet of Things point of view. On the other hand, wireless multi-hop communication is, for some industrial monitoring applications, still very much a thing. However, the self-organization is tuned down quite a bit, compared to the visions from academic research. In other industrial settings, wireless communications is limited to a single hop or even omitted completely, returning to wired network. Still, even in the wired setting, a lot of results from academic research come to fruit, especially concerning the self organization. However, the focus is not always so much on the re-configuration of nodes as on the initial configuration. This is due to the fact that more and more sensors and actuators get added to factories, home automation and similar settings, and the number of available trained experts does not rise at the same speed. Also, experts are not cheap. Therefore, the devices should configure themselves automatically, with minimalistic human interaction.

From an industry perspective, robustness criteria, sometimes coupled with real-time requirements, leave us with these three distinct scenarios of interest:

1) Planned Wireless Multihop Networks
2) Planned Wireless Single Hop Networks
3) Self-Organizing Wired Networks

Even though the reliability and predictability of wires is usually preferred, there are some scenarios where drawing cables all the way is not possible. An example would be a long bridge on which vibration data is gathered. How huge amounts of data can be transmitted reliable, probably pre-processed, is still ongoing research. This represents scenario 1. Research focusses, apart from reliable data transmission, on the refinement of data on the sensor nodes. Often, AI is mentioned.

Scenario 2 can be seen in factories with moving machinery that should be monitored. Here, the data is transmitted wirelessly from the moving part of the machine to a stationary base station. Another example are monitoring elements that are added to already existing factories. They might work wired, but transmission between rooms is sometimes added wireless as drawing new cables would involve opening and resealing walls behind fire proof doors. Here, the classic sensor network research which focusses on data fusion, routing and similar is complete. There is, however, still research concerning the speed and reliability of data transmission, currently focussed on 6G networks.

Still, all wireless communication combined has less than 10% marketshare, cabling is prevalent.

Scenario 3 represents the research focus of Perinet. With all the research about self-organizing networks we have seen in the last decades, one would expect that this topic was already solved. However, when moving from academics to industry, you realize that there is a surprising amount of differences.

II. Internet of Things

The major focus point of the Internet of Things was, right from its inception, to include a lot of devices, if not all, into the Internet. A typical example was the smart fridge that would order new milk all by itself once it was almost depleted. For

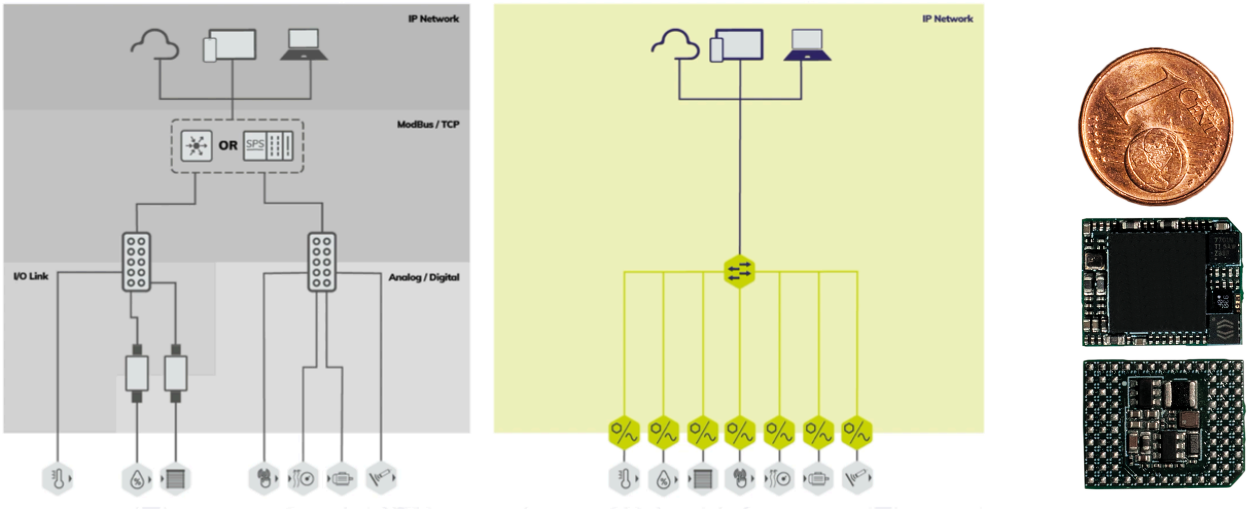


Fig. 1. Example of Current Setups (l), Future Setup (m), and a periCORE

such devices, commercial solutions exist. However, what is the state for 'normal' sensors and actuators?

One of the larger problems within the huge research community that has arisen around the topic of sensor networks in the 1990s and early 2000s was interoperability, or, rather, the lack of it. In 2005 the group of David Culler already called for a common abstraction within sensor networks [3], [6]. Their basic idea was that sensor networks, like the Internet, needed an hourglass architecture. Different protocols and hardware at the bottom, a common protocol (their proposal: SP, the sensornet protocol) in the middle, and lots of different applications above. This idea never took flight directly. However, with the Internet of Things it was promised that IP would be the common protocol. Is it, though?

When you look at typical vendors of one type if IoT devices, e.g. smart home, it pretty soon becomes apparent that even though it says IoT on the outside, what is inside is usually still a bunch of proprietary protocols. Interoperability between different vendors is basically not given, once you decide for one vendor you have to buy everything from that vendor.

III. Real Internet of Things

What we understand under the term Internet of Things differs. In our opinion this means that all devices in the network 'speak' IP. Figure 1 (left side) shows a typical setup in the Industrial IoT as it is now. There is some IP network within the company. However, once we go down to operations level, we usually find SPS, Modbus and similar. Many of the sensors are still analog, to address them you might even need to know to which output on which SPS they are connected when you try to sample values.

Our goal now was to enable IP connectivity up to the last sensor in order to be able to provide ease of deployment, better maintenance, and ease of use . When every sensor and actuator is accessible using IP, the network structure as depicted in the middle of the figure results, enabling easy access from anywhere within the network. On the right side of the figure you can see the periCORE chip we developed for this purpose. It can be integrated directly into sensors and actuators to make them accessible via IP.

The periCore it is also used within our periNODEs. PeriNODEs can be attached to existing (analog) sensors and actuators to give them the same capabilities described above. The main advantage of this approach is that existing sensors and actuators can continue to be used, making it highly sustainable.

IV. Challenges

While this might seem straightforward, there were numerous challenges we faced. These include but are not limited to

- User interfaces for non-Experts
- Device Discovery
- Name translation
- Semi-automatic setup
- Browsers not following specifications
- Security

One of the main problems when setting up and maintaining an Industrial Internet of Things deployment is actually people. In many cases you need highly trained experts for the deployment and for maintenance. These experts are expensive and in some cases not even available. Therefore, we needed to develop user interfaces that could be used by non-experts. One thing everyone is familiar with nowadays is the browser. Therefore, we decided to include a small web server on the periCORE which enables users to access their sensors and actuators directly from within their preferred browser.

For many of the issues we faced individual solutions were in existence but there was no combined approach. Table I shows the approaches we used to solve some of the problems we initially faced. For self-organization purposes we use IPv6 link local addressing and mdns. Routing of data is done using MQTT, which is already prevalent in many IIoT deployments. As security is an ever rising topic, we included mTLS and PKI. Json is used for interoperability.

TABLE I
PROBLEMS WE FACED AND THEIR SOLUTIONS

Problem	Solution
Self-organization	IPv6 LL / mdns
Data Routing	MQTT
Security	mTLS / PKI
Interoperability	Json

TABLE II
PROBLEMS WE DID NOT FACE AND THE REASONS

Problem	Implicit Solution
Data efficiency	100MBit to the sensor
Communication range	the network is our transparent tunnel
Energy Consumption	SPE Cable
Shared medium	SPE Cable
Antenna	SPE Cable

There are also some problems which are discussed in research which we did not have to face. For example, a lot of research has been focussed on reducing the number of transmitted bytes as much as possible. For us this is in most cases not relevant because we have a connection of 100MBit down to the sensor as we are using Single Pair Ethernet (SPE) cables. Table II shows further examples. Basically, the usage of SPE cables solved all of these problems we would have faced had we decided to use a wireless approach.

Still, we faced other challenges. For example, some browsers do not completely fulfill their specifications. Others check their internet connection at starting time. If this is an IPv6 connection, they go into IPv6 mode, otherwise they go into IPv4 mode. If they went into IPv4 mode, locally connected devices using IPv6 LL can not be discovered.

One part of future research focusses on scaling issues for the sensors as the number of sensor nodes is already much higher than the number of regular computers in the internet today.

Another part of future research is security. Even though we already have security measures in place, secure multicast is still an issue. Revocation lists are often used for which a replacement might be interesting. When working with Linux containers oder generally any virtual machine, secure private storage does basically not exist nowadays. When using security domains, a membership in multiple of those could be profitable.

V. CONCLUSION

In this paper we took a look at the usability of sensor network protocols from an industrial perspective. We discussed relevant scenarios and described which approaches from academia have found their way into industry, which have been rejected, and which still require some more research in our opinion.

REFERENCES

[1] I. Chatzigiannakis, S. Nikoletseas, and P. Spirakis, “Smart dust protocols for local detection and propagation,” in *POMC ’02: Proceedings of the second ACM international workshop on Principles of mobile computing*. New York, NY, USA: ACM Press, 2002, pp. 9–16.

[2] I. Chatzigiannakis, S. Nikoletseas, and P. G. Spirakis, “Efficient and robust protocols for local detection and propagation in smart dust networks,” *Mob. Netw. Appl.*, vol. 10, no. 1-2, pp. 133–149, 2005.

[3] D. Culler, P. Dutta, C. T. Ee, R. Fonseca, J. Hui, P. Levis, J. Polastre, S. Shenker, I. Stoica, G. Tolle, and J. Zhao, “Towards a sensor network architecture: Lowering the waistline,” in *Proceedings of Hot Topics in Operating Systems (HotOS ’05)*, 2005.

[4] D. Graff and R. Karnapke, “Cyber-Physical Systems—Exemplary applications and a distributed execution platform,” in *2016 IEEE International Conference on Emerging Technologies and Innovative Business Practices for the Transformation of Societies (EmergiTech) (EmergiTech2016)*, Mauritius, Aug. 2016, pp. 120–125.

[5] J. M. Kahn, R. H. Katz, and K. S. J. Pister, “Next century challenges: mobile networking for ”smart dust”,” in *MobiCom ’99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*. New York, NY, USA: ACM Press, 1999, pp. 271–278.

[6] J. Polastre, J. Hui, P. Levis, J. Zhao, D. Culler, S. Shenker, and I. Stoica, “A unifying link abstraction for wireless sensor networks,” in *SenSys ’05: Proceedings of the 3rd international conference on Embedded networked sensor systems*. New York, NY, USA: ACM Press, 2005, pp. 76–89.

TBLE: Time-Synchronized Routed Mesh Communication for BLE

Laura Harms[1,2], Olaf Landsiedel[1,2]
[1]*Kiel University, Germany*, [2]*Chalmers University of Technology, Sweden*
{lah, ol}@informatik.uni-kiel.de

***Abstract*—Bluetooth Low Energy is widely used for IoT applications, but it lacks a solution for routed and time-synchronized multi-hop mesh communication. Instead, we commonly see the use of IEEE 802.15.4 with its Time-Slotted Channel Hopping (TSCH) MAC layer. In this work, we introduce TBLE, a protocol that integrates TSCH into BLE, enabling time-synchronized routed mesh communication. Our experimental evaluation shows TBLE to achieve similar performance to TSCH on IEEE 802.15.4, with average latency reductions of up to 20%.**

***Index Terms*—IoT, WSN, TSCH, BLE, IEEE 802.15.4**

I. Introduction

Bluetooth Low Energy (BLE) has become the dominant standard for communication in low-power wireless networks, with an estimated 5 billion BLE devices set to be shipped in 2023 alone [4]. Its wide availability, low-cost and energy efficiency, make it the preferred choice for most smart devices. However, in industrial and smart home applications, another protocol, IEEE 802.15.4, is the prevalent communication standard. While BLE offers a less complex physical layer with cheaper radios, both protocols have limited range and rely on multi-hop networking for longer distance communication.

For IEEE 802.15.4, several protocols for mesh networking exist, with Time-Slotted Channel Hopping (TSCH) [1] being an established technique for routing-based communication in IEEE 802.15.4. In the realm of BLE, Bluetooth Mesh [3] is the standard mesh protocol, utilizing managed flooding over BLE advertisements. Mesh communication in BLE is limited to a single end-to-end communication at a time, using the entire network, and to our knowledge, there is no time-synchronized routing-based mesh communication protocol for BLE.

Previous works have explored the combination of TSCH and BLE either in terms of coexistence [5], [7] or using a single radio for both protocols and communicating TSCH control information using concurrent BLE transmissions [2]. Especially the latter work raises the question, why we still need IEEE 802.15.4 for TSCH, and why we rely on BLE only for communicating control information.

In this paper, which in parts summarizes our recently published paper [8], we propose TBLE, a protocol that combines the BLE physical layer with the TSCH MAC layer. By sending standard TSCH packets as part of time-synchronized BLE advertisements, TBLE enables routed mesh communication over BLE, effectively replacing the need for IEEE 802.15.4. This integration allows the use of well-established protocols, including real-time communication protocols, on top of BLE.

To demonstrate the feasibility and performance of TBLE, we design and implement a BLE driver for the nRF52840-DK [9] for Contiki-NG [10] and adapt it to transmit valid TSCH frames within BLE packets. Our evaluation shows that TBLE performs comparably to IEEE 802.15.4 TSCH, with the potential for lower average latencies of up to 20%. Additionally, due to the higher spectral efficiency of BLE compared to IEEE 802.15.4, TBLE enables more parallel routed communications, adding the potential to further reduce latency and increase throughput.

The contributions of this paper are as follows: (1) we present TBLE, a protocol bridging the gap for routed mesh communication in BLE by adding interoperability between TSCH and BLE, (2) we design and implement a BLE driver for the Nordic nRF52840-DK for Contiki-NG, and (3) we demonstrate TSCH over BLE, showcasing TBLE as a practical routed mesh protocol for BLE.

II. Design

Combining the MAC layer protocol Time-Slotted Channel Hopping (TSCH) and the physical layer protocol Bluetooth Low Energy (BLE) comes with a set of challenges. The goal of this work is to send TSCH frames as part of standard BLE advertisements. For other BLE devices, these appear like standard BLE packets, while devices running TBLE can recognize them and form a standard TSCH network using a BLE PHY instead of the IEEE 802.15.4 PHY for communication. To achieve this, we need to adjust the timings within a TSCH timeslot, create new hopping sequences for the higher number of available channels, and modify TSCH's time synchronization method to work with BLE radios. While these are several things that need adjustment, we can keep a major part of TSCH untouched: its payload size. IEEE 802.15.4 packets allow up to 127 data bytes, whereas BLE supports up to 258 bytes (cf. Fig. 1). Thus, we can send unmodified TSCH frames as part of BLE advertisements.

Timeslot Timing. TSCH timeslots are standardized to a length of 10 ms. Fig. 2 illustrates such a timeslot for a sender and a receiver. The timeslot length is mainly influenced by the frame and acknowledgment transmission times, which assume a radio data rate of 250 kbit/s, the data rate of IEEE 802.15.4. With BLE's different radio data rates of 125 kbit/s, 500 kbit/s, 1 Mbit/s, and 2 Mbit/s we need to adjust the slot length to the transmission times corresponding with these data rates,

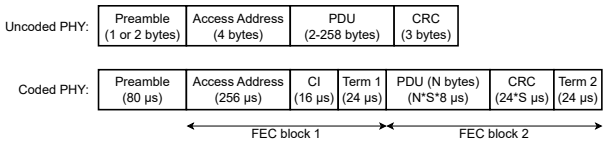


Fig. 1: BLE PHY packet formats for the uncoded PHY (1 Mbit/s and 2 Mbit/s) and the coded PHY (125 kbit/s and 500 kbit/s).

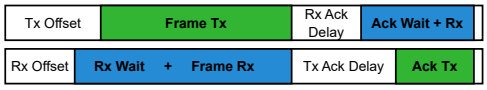


Fig. 2: Simplified timing within a TSCH timeslot.

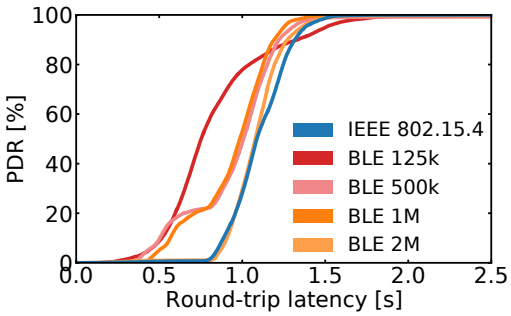


Fig. 3: Orchestra round-trip performance comparison for TBLE and IEEE 802.15.4 TSCH

resulting in slot lengths of 18.5 ms, 7.5 ms, 5 ms, and 4 ms for the four BLE modes, respectively.

BLE Adjustments. We want to send TSCH frames as BLE advertisements, and all devices not taking part in the TSCH network to easily discard these. For the uncoded BLE PHY (1 Mbit/s and 2 Mbit/s) we choose an application-specific access address and embed the TSCH frame as the packet's PDU (cf. Fig. 1). For the coded PHY (125 kbit/s and 500 kbit/s), the nRF52840 can only receive packets with the standard advertisement access address. Thus, we send standard advertisement packets as the PDU and use an undefined PDU type (0x90) to enable other devices to discard the packet.

TSCH Time-Synchronization. For time-synchronization, TSCH relies on the timestamping capabilities of the radio. In IEEE 802.15.4, TSCH uses the timestamp of the *start of frame delimiter (SFD)* for time-synchronization. As BLE doesn't have an SFD (cf. Fig. 1), we use the radio's functionality of timestamping the *end of address* for synchronization. As this timestamp is not as precise as the SFD timestamp in IEEE 802.15.4, we need to add radio specific offset compensations.

With these modifications of TSCH and BLE, we are able to form TSCH networks over BLE sending mostly standard-compliant BLE packets or BLE packets easily discardable by a device not running TBLE.

III. Evaluation

To evaluate the performance of TBLE, we use the standard benchmarking solution of Contiki-NG, Orchestra [6] in our 20-node testbed, spanning the top-most floor of a university building ($500m^2$) with offices and student rooms. For all four BLE modes, we are able to form a time-synchronized TSCH network. For evaluating the performance of TBLE, we compare the performance for each BLE PHY with the performance of TSCH using the IEEE 802.15.4 PHY. Fig. 3 shows that all BLE modes have a shorter average round-trip latency than IEEE 802.15.4. However, only BLE 500k and BLE 1M reach the same reliability as IEEE 802.15.4. While BLE 125k has the lowest latency for a significant number of packets, its latency increases drastically for nodes that are more than a single hop away due to a significantly longer slot length. BLE 2M experiences the most challenging environment as it has the lowest connectivity to neighbors in the formed TSCH network. Overall, BLE 500k and BLE 1M are viable alternatives to IEEE 802.15.4 achieving comparable reliability as IEEE 802.15.4 while decreasing average latency by 20%.

IV. Conclusion & Future Work

In this paper, we introduce TBLE, a protocol that combines the Time-Slotted Channel Hopping (TSCH) MAC layer with Bluetooth Low Energy (BLE) for time-synchronized routed mesh communication. TBLE closes the gap of routed mesh communication in BLE and has the potential to eliminate the need for IEEE 802.15.4. Our experimental evaluation demonstrates TBLE's comparable performance to IEEE 802.15.4 TSCH, with potential latency reductions of up to 20%.

In future work, we plan to conduct more comprehensive evaluations of TBLE to assess its performance in the presence of interference. Additionally, we plan to look at the implications of larger deployments and different environments on TBLE's performance. Furthermore, by exploring alternative protocols beyond Orchestra, we anticipate to be able to highlight the advantages of either of the four BLE modes.

References

[1] "IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer," IEEE, Tech. Rep., 2012. [Online]. Available: http://ieeexplore.ieee.org/document/6185525/

[2] M. Baddeley, A. Aijaz, U. Raza, A. Stanoev, Y. Jin, M. Schuß, C. A. Boano, and G. Oikonomou, "6TiSCH++ with Bluetooth 5 and Concurrent Transmissions," in *EWSN*, 2021.

[3] Bluetooth SIG, "Mesh Profile 1.0," 2017. [Online]. Available: https://www.bluetooth.com/specifications/specs/mesh-profile-1-0/

[4] ——, "2021 Market Update," 2021. [Online]. Available: https://www.bluetooth.com/wp-content/uploads/2021/01/2021-Bluetooth_Market_Update.pdf

[5] O. Carhacioglu, P. Zand, and M. Nabi, "Cooperative Coexistence of BLE and Time Slotted Channel Hopping Networks," in *IEEE PIMRC*, 2018.

[6] S. Duquennoy, B. A. Nahas, O. Landsiedel, and T. Watteyne, "Orchestra," in *ACM SenSys*, 2015.

[7] H. Hajizadeh, M. Nabi, M. Vermeulen, and K. Goossens, "Coexistence Analysis of Co-Located BLE and IEEE 802.15.4 TSCH Networks," *IEEE Sensors Journal*, vol. 21, no. 15, pp. 17 360–17 372, aug 2021.

[8] L. Harms and O. Landsiedel, "TSCH meets BLE: Routed Mesh Communication over BLE," in *IEEE DCOSS-IoT*, 2023.

[9] Nordic Semiconductor, "nRF52840 DK." [Online]. Available: https://www.nordicsemi.com/Products/Development-hardware/nrf52840-dk

[10] G. Oikonomou, S. Duquennoy, A. Elsts, J. Eriksson, Y. Tanaka, and N. Tsiftes, "The Contiki-NG open source operating system for next generation IoT devices," *SoftwareX*, vol. 18, p. 101089, jun 2022.

Towards Multi-hop BLE-Based Communication Using a Custom Routing Approach in Zephyr

Florian Jung and Silvia Krug
IMMS Institut für Mikroelektronik- und Mechatronik-Systeme gemeinnützige GmbH (IMMS GmbH)
Ilmenau, Germany
Email: florian.jung@imms.de

***Abstract*—Bluetooth Low Energy (BLE) is a wireless network protocol for low-power applications that is meant to be used for star networks, beacons or - when using the Bluetooth Mesh stack - mesh networks. Available hardware with low energy consumption enables various applications. However, single hop networks limit the usage of the corresponding nodes and multi-hop communication would be interesting. In this study, we present an approach how to enable this for otherwise beacon-based RuuviTag sensors using Zephyr. The idea is to use routing mechanisms instead of a full mesh to keep the energy efficient operation of the RuuviTag. In addition, we will present an initial performance evaluation and discuss further optimization potential.**

***Index Terms*—Internet of Things; BLE; Routing Protocol; Performance Evaluation**

I. Introduction

Bluetooth Low Energy (BLE) is a very popular communication standard because it is able to provide energy efficient operation of the connected devices. In most cases, the application consists of single-hop communication only. If larger distances are to be covered by sensors, this would require multiple base stations or a mobile agent to collect data, even if the sensors are deployed within communication range of each other. Such a scenario is typically addressed by multi-hop communication. The only standardized way to achieve this with BLE is the mesh standard [1]. Alternatives, such as 802.15.4 based OpenThread have a high energy consumption for the routers, which renders them infeasible for battery-powered operation. Hence, the goal of this study is to explore options to perceive the energy efficiency of BLE while enabling multi-hop communication. As an example, we use RuuviTags with several sensors and an energy efficient design, allowing the default BLE beacon firmware to run on them for 2-3 years based on a CR4570 coin cell battery. The goal is to achieve a minimum life time of 1 year for all devices in the network including battery powered routers.

II. State of the Art

BLE Mesh is the standardized option to realize multi-hop communication within the BLE specification. We decided not use BLE Mesh because it is comparably energy inefficient [2], [3] due to the overhead of managed flooding to forward messages [1]. This mechanism makes the network very stable but power draining despite optimization efforts as in [4]. Other approaches use IPv6 over BLE [5]. Which shows that BLE can outperform 802.15.4 in terms of energy consumption. Recent work has focused on multi-hop communication [6], [7] to show potential options. However, none of the IPv6 based approaches is currently available in Zephyr. Instead, we will define an algorithm for multi-hop Bluetooth Low Energy networking suitable for RuuviTags that can act as routers.

Before we define an algorithm for multi-hop Bluetooth Low Energy networking we want to elaborate where the issues with mesh networks like OpenThread come from. OpenThread is an open-source implementation of the Thread Protocol which is based on the IEEE 802.15.4 standard. Like BLE, it uses the same frequency band of $2.5\,\mathrm{GHz}$ but establishes a mesh-network with IPv6-support on top of the 802.15.4 MAC layer. Other than Bluetooth Mesh it uses explicit connections with routing tables instead of managed flooding. However, OpenThread is designed for mains-powered routers only. For further details refer to the specification at [8].

The issue towards energy efficiency with this approach is that the nodes responsible for routing the messages are in an always-on mode. One reason is to make sure that no messages from any neighbors are lost. Another reason is that they are also responsible for scanning for new devices in range. Besides that, they also have to ensure that all their connections to devices within range are still active to update the routing table in case of changes as fast as possible and signal it to all the other routing nodes in the whole network. This leads to a very high energy consumption rendering the routing nodes infeasible for battery-powered operation, as we will also evaluate in Section IV. As a result it is not possible to use OpenThread based multi-hop communication for outdoor or mobile scenarios.

III. Routing Algorithm Design

We chose Zephyr to implement the routing algorithm because the Nordic Semiconductor development team ports their boards themselves to this operating system. This results in very good support and performance for their chips, where the nRF52832 is the one the RuuviTag is based on.

To construct the algorithm, let us first recall the basic specification of BLE. In the default BLE stack, a device can have multiple roles: The most basic pair of roles is the *Broadcaster* sending so-called advertisements and the *Observer* receiving them in a connectionless manner.

The more advanced pair for use-cases where connection-based communication is needed consists of the *Peripheral* and the *Central*. The *Peripheral* acts as a server and cannot connect to other Peripherals. It advertises its service until it connects to a *Central*. Other connections are not allowed then. The *Central* acts as a client and can connect to multiple Peripherals but also not to other Centrals.

These latter pair of roles could be used to build the multi-hop communication if we do not want to implement connections by ourselves. However, there is no way to relay data for more than one hop as stated by [6]. To achieve more hops, a device needs to act as a Peripheral and Central concurrently which is allowed by the Bluetooth Core Specification 5.3 [9] and is implemented in the Bluetooth Low Energy stack of Zephyr. This is possible in Zephyr because the roles are instances of state machines in the link layer implementation. When both are used, there is one instance for each role [10].

With this possibility in mind, we can construct a tree layout for our architecture. The root node is our gateway device and only acts as a Central device, to be able to connect to many Peripherals at once. All other nodes are Peripherals and Centrals simultaneously. When building up the network, the gateway listens to advertisements from up to a maximum number of n_{max} nodes, trying to connect to the network. It will establish a connection with each node in range and continues with this process until there are n_{max} nodes connected. If a connection breaks, it will restart to try to connect to other nodes. Once connected, the other nodes will then start scanning for advertisements as Centrals and establish new connections the same way as the gateway did. At the same time, they will notify their measurements as Peripherals to their parent node which will in turn forward those messages to its parent node and so on. This way, a one directional message forwarding chain to the gateway is built up which roughly follows the ideas in [6] but was developed independently. An algorithm for the build-up process is shown in Algorithm 1.

Algorithm 1 Algorithm for build-up process in nodes

Require: $max\_child\_count > 0$
1: $child\_count \leftarrow 0$
2: start_advertising() {starts Peripheral state machine}
3: wait_for_connection()
4: {messages can be sent to parent now}
5: **while** *connected* **and** $child\_count < max\_child\_count$ **do**
6: start_scanning() {starts Central state machine}
7: wait_for_advertisements()
8: connect_to_found_devices()
9: $child\_count \leftarrow child\_count + new\_connections\_count$
10: **end while**

IV. Performance Evaluation

To evaluate the performance of our proposed solution, we first implemented it in Zephyr. Then we setup a comparison with OpenThread Routing nodes and measured the energy consumption of the nRF52840. For BLE, we use a RuuviTag first with the default Beacon firmware and afterwards with the new custom routing firmware. The one with the custom firmware is at first connected to a gateway and then we add a child device resulting in the setup visualized in Figure 1. While

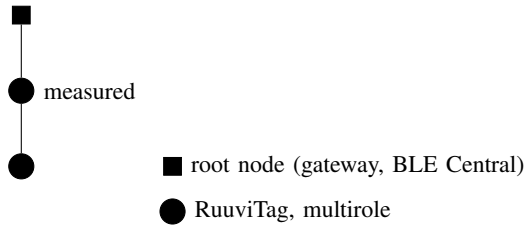


Fig. 1. Measurement setup for RuuviTag with custom routing firmware

it is connected to a child, the LED blinks once every minute to indicate the connection state. Both devices are configured to send their sensor data every minute. For OpenThread, we use a nRF52840 dongle that is configured as a router with similar sensors.

In Figures 2 and 3 the power traces for the RuuviTags are visualized.

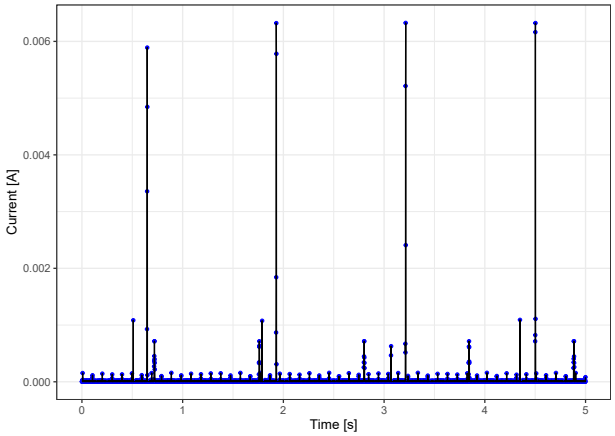


Fig. 2. Power Trace of RuuviTag with original firmware

The connection process of our custom firmware takes about 26 seconds and consumes about $4.7\,\mu\mathrm{Ah}$. The LED blinking consumes about $0.45\,\mu\mathrm{Ah}$ and the basic pattern consumes $0.18\,\mu\mathrm{Ah}$ per second. Therefore, the extrapolated capacity consumption is about $15.7\,\mathrm{mAh}$ per day which results in a lifetime of less than 64 days with the $1000\,\mathrm{mAh}$ of the built in battery of the RuuviTag. This sounds low, but when compared to the OpenThread router with a lifetime of about 6 days when powered with the same battery, this initial stage is already a massive improvement (≈ 10 times better) compared to OpenThread. In addition, we have measured the energy consumption of an OpenThread Sleepy End Device, the most battery efficient but very restricted device type of OpenThread. This results in an approximated lifetime of 79 days, which is not that much of an improvement compared to the BLE

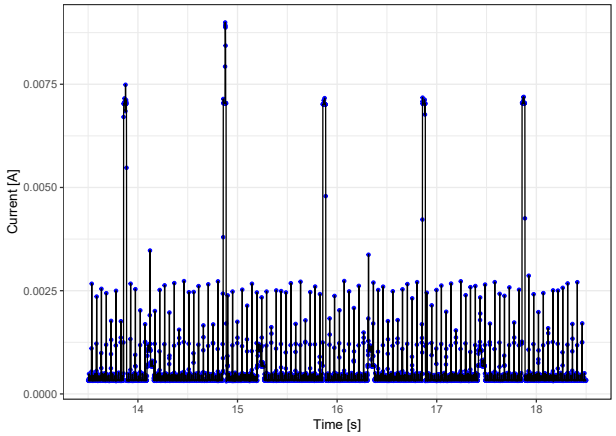


Fig. 3. Power Trace of RuuviTag with routing firmware

tree prototype. Table I lists our calculations based on the measurements.

TABLE I
TABLE SHOWING THE EXTRAPOLATED RESULTS

Node	mAh per 24h	lifetime in days	...years
RuuviTag BLE beacon	0.6912	1446	3.964
RuuviTag BLE multihop	15.6816	63	0.175
OT Router	161.4	6	0.017
OT SED	12.6	79	0.217

It should be noted, that the measured scenario is not that realistic and a bigger network will result in more power consumption because of more messages sent, especially for the devices near the root node. We plan to analyze this in more detail in the future.

V. Room for Optimizations

So far, we only implemented a prototype for proof-of-concept. This leaves room for many possible optimization ideas that we want to discuss in this section. To increase the energy efficiency, the messages could be synchronized, enabling a longer idle time for each device which in turn will reduce the energy consumption. This is similar to ideas presented in [4]. Since all nodes except the gateway have no knowledge about the network in the current version, there should be a way to control which node in range should initiate the connection. This could be possible by forwarding the received advertisement to the gateway that then decides which node should connect to the new device. This requires a simple routing table in each node.

In order to try to optimize the width and depth of the tree, parent nodes could decide if the child should try to connect to other nodes or not. This could be achieved using a writable characteristic that, when set to `true`, activates the otherwise per default always disabled Central mode.

Another improvement could be made by setting up a connection on top of the connectionless Observer-Broadcaster-model from BLE, where the only messages sent are advertisements as in [7]. When choosing this method, advertisements including the hop distance to the gateway should be broadcasted from parent nodes allowing possible new children to choose the parent closest to the gateway. Thus, building the routing table based on this information and only the parent would forward corresponding data. This reduces the broadcast of downstream advertisements in the prototype. As a result, less messages are sent but the risk of bottlenecks increases. Those bottlenecks could be prohibited by restricting the amount of nodes of a subtree, as mentioned before. This option should be as close as possible to the original energy consumption of the RuuviTag. We therefore want to explore this further.

When a link breaks the current way to handle this is to disconnect the complete subtree and restart the build-up process from there. When using the Observer-Broadcaster-model, this could be optimized by just letting the orphaned nodes connect to other devices in reach.

VI. Conclusion

In this paper, we presented a proof-of-concept to implement a multi-hop network on top of the Bluetooth Low Energy protocol by using the Peripheral and Central Roles concurrently in Zephyr. The observed measurements show the potential of this approach to enable battery powered routers. However, we need to further refine the protocol in the future to unlock the potential and achieve a node lifetime closer to 1-2 years.

Acknowledgment

This work is part of the thurAI research project which is funded by the German Land of Thüringen via the Thüringer Aufbaubank under the reference 2021 FGI 0008.

References

[1] *Bluetooth Mesh Networking - An Introduction for Developers*, Bluetooth SIG, Inc., 12 2020, v1.0.1.
[2] S. M. Darroudi and C. Gomez, "Experimental Evaluation of 6BLEMesh: IPv6-Based BLE Mesh Networks," *Sensors*, vol. 20, no. 16, p. 4623, 2020.
[3] N. Paulino, L. M. Pessoa, A. Branquinho, R. Almeida, and I. Ferreira, "Optimizing Packet Reception Rates for Low Duty-Cycle BLE Relay Nodes," *IEEE Sensors Journal*, vol. 22, no. 13, pp. 13 753–13 762, 2022.
[4] D. Hortelano, T. Olivares, and M. C. Ruiz, "Reducing the energy consumption of the friendship mechanism in Bluetooth mesh," *Computer Networks*, vol. 195, p. 108172, 2021.
[5] M. Spörk, C. A. Boano, M. Zimmerling, and K. Römer, "Bleach: Exploiting the full potential of ipv6 over ble in constrained embedded iot devices," in *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*, 2017, pp. 1–14.
[6] H. Petersen, T. C. Schmidt, and M. Wählisch, "Mind the Gap: Multi-hop IPv6 over BLE in the IoT," in *Proceedings of the 17th International Conference on emerging Networking EXperiments and Technologies*, 2021, pp. 382–396.
[7] H. Petersen, J. Brodbeck, T. C. Schmidt, and M. Wählisch, "IPv6 over Bluetooth Advertisements: An alternative approach to IP over BLE," *arXiv preprint arXiv:2210.06236*, 2022.
[8] Thread 1.3.0 Specification. Thread Group. [Online]. Available: https://www.threadgroup.org/ThreadSpec
[9] *Bluetooth Core Specification*, Bluetooth SIG, Inc., 07 2021, v5.3.
[10] Bluetooth overview. Zephyr Project. [Online]. Available: https://docs.zephyrproject.org/latest/connectivity/bluetooth/overview.html

BLE Periodic Advertising as an Alternative to BLE Mesh and Scatternets for Multihop Communication

Bennet Kaluza
Brandenburg University of Technology Cottbus–Senftenberg, Germany
{bennet.kaluza}@b-tu.de

***Abstract*—Waldwächter 5G is a project about preventing damages to forests, for example by fire. For this case, sensors connected in a Bluetooth Low Energy (BLE) multi-hop environment are deployed to detect abnormalities. BLE is a widely spread wireless communication technology used for low power devices, such as smart home appliances or sensors. This paper presents two approaches for BLE multi-hop communication, BLE Mesh as a flooding based network primarily used for non-power-limited devices and BLE scatternets as a lose collective term for a lot of interconnected smaller networks. It then introduces the idea of a third possible option, a network topology based on a BLE 5.0 feature called Periodic Advertising (PA). The idea of a approach to PA meshes is explained and the advantages presented.**

***Index Terms*—BLE, WSN, PAwR, multihop communication, scatternets**

I. Introduction

With climate change causing increasingly extreme and unpredictable weather patterns, forest fires are becoming more of a problem than they already were. Waldwächter 5G as a whole aims for an earlier detection to prevent as much damage as possible. Part of that is a widespread sensor network on the ground measuring temperature, humidity and gas resistance. Sensor networks are always bound to power constraints, meaning a communication protocol has to balance transmission quality and power consumption.

As such, the Waldwächter sensors are to be connected via a multi-hop network, making the aforementioned balance ever harder to find.

Several communication technologies could be used in this scenario, but after a evaluation Bluetooth Low Energy (BLE) was chosen for the purpose of this project. BLE is a widely used wireless communication technology focused on reducing power consumption while still maintaining a similar range to classic Bluetooth, making it a perfect fit for sensor networks.

The communication patterns needed are similar to classic sensor networks and consist of the following:

We have few 5G gateways and many sensor nodes providing BLE Long Range functionality, using multihop communication for further range. We need to be able to flood messages from the gateway to every node from time to time, forward data from any node to the nearest gateway and get data from any node to all nodes.

The remainder of the paper will explain how BLE Periodic Advertising can be used to achieve this sort of network. It starts with an introduction into BLE and its most important features, especially Periodic Advertising (Section II). It is followed by an explanation on how Periodic Advertising works and what possibilities it provides (Section III). The next section presents a PA-based concept for a sensor network (Section IV) followed by a conclusion (Section VI.

II. Bluetooth Low Energy

Bluetooth is a protocol stack from the physical to the application layer. Version 4.0 introduced the Low Energy protocol (BLE), which is designed for very low-power operation and a simpler rapid link build-up compared to the incompatible "classic"Bluetooth (BR/EDR). The core specification [1] covers the BLE controller, which is responsible for the physical and MAC layer, the host to controller interfaces, and the BLE host, which is responsible for the representation of services.

BLE divides the 2.4 GHz ISM band into 40 channels of 2 MHz bandwidth. The coexistence with other users is mainly achieved through a pseudo-random frequency hopping across 37 channels. Three channels are dedicated to advertisement broadcasts for service discovery.

Two of the most important features of BLE are Scanning and Advertising.

Advertising is a broadcast, which can for example contain data about services provided by the advertising device, application data or specific manufacturer data. The advertising interval triggers broadcasting of the advertisement message. In each interval the message is sent to each advertising channel once. To reduce collisions between advertisers, a random delay up to 10 ms is prepended. Next to legacy advertisements, which are small and sent in one piece on advertising channels, BLE 5.0 introduces Extended Advertising, which is made up of two parts: The first part is a message on the advertising channel announcing the second part which is then sent via a data channels. This enables bigger advertisement messages without blocking the advertisement channels.

Scanning works with a scan interval, which defines when to switch advertising channels, and a scan window to define how long to listen for advertisements. In response to a received advertisement, the scanning device can request additional information (active scanning) or try to establish a connection. The upper bound for the neighbor discovery latency depends on the combination of advertising interval, scan interval, and scan window [2]. For example, the scan window can be chosen twice as large as the advertising interval. In [3] the impact on data reception of BLE advertisements was studied in a large sensor network.

Energy efficiency is achieved through time-slotted channel hopping: The *advertising interval* triggers broadcasting of the advertisement message. In each interval the message is sent to each advertising channel once. To reduce collisions between advertisers, a random delay up to 10 ms is prepended. The *scan interval* switches between advertising channels and the *scan window* defines how long to listen for advertisements. In response to a received advertisement, the scanning device can request additional information (active scanning) or try to establish a connection. The upper bound for the neighbor discovery latency depends on the combination of advertising interval, scan interval, and scan window [2]. For example, the scan window can be chosen twice as large as the advertising interval. In [3] the impact on data reception of BLE advertisements was studied in a large sensor network.

III. Periodic Advertising and its extensions

As mentioned before, longer scanning windows provide a better chance at receiving but also increase the power consumption. Periodic Advertising aims to provide a better solution if data is to be sent repeatedly in fixed intervals. *Periodic Advertising* is a BLE 5.0 feature, which based on extended advertisements allows data to be sent in fixed intervals. It works as a connectionless multicast one-way transmission for small payloads. After an initial advertisement containing an extended advertising indication (AUX_ADV_IND), sender and receiver will synchronize via AUX_ADV_IND (contains auxiliary info about the interval and starting point of the PA) followed by a periodic AUX_SYNC_IND, which represents an actual PA event and data transmission. A AUX_SYNC_IND can also be followed by additional data (AUX_CHAIN_IND) in one interval. The periodic advertisement payload can be changed between intervals. Of these messages, only the first AUX_ADV_IND is sent on the primary advertising channels and has to be manually scanned for (bigger scanning windows). Every other packet is sent on secondary advertising channels and at specific points of time, so that the receiver knows exactly when to wake up.

In the BLE 5.4 specification, *Periodic Advertising with Response* (PAwR) was added. This changed quite a bit about what Periodic Advertising is able to achieve.

First of all, events are now split into several periodic PAwR Subevents. Receiving devices can now synchronize to specific subevents, receiving only the data they need from the PA, further reducing the time they need to be awake. Additionally each subevent has several response slots on which a receiver can respond to the advertising device. Managing response slots and which subevent to subscribe to is a application level issue.

Overall, this enables a connectionless 1-to-M bidirectional communication. The amount of advertisers a receiving device can synchronize to is the same as the maximum number of possible active connections specified by the BLE controller.

IV. Concept

Based on PAwR and its ability to create 1-to-M bidirectional communication, it seems easy to create a concept for a network structure. For our network fulfilling the communication patterns mentioned in the introduction, we let the 5G gateway advertise periodically and each non-connected sensor node is scanning. As soon as a sensor node is synchronized with the gateway, it will start advertising its own PA for the next wave of non-connected sensor nodes. Every node tries to synchronize to two nodes which are closer to the gateway then themselves which creates several paths for messages to be sent.

The resulting mesh-like structure enables a power efficient way to get data from any node to any other node, possibly via the gateway. We also can propagate data from the gateway via the Periodic Advertising to every single sensor or gather sensor data via PAwR response slots. The PAwR subevents and corresponding response slots also provide the option to easily classify messages as alarms, data collection or telemetry. For example we could reserve response slot 1 for alarms and response slot #2 for collected data.

This mesh-like network structure is easy to create and manage, power efficient and enables bidirectional data transmission.

V. comparison with existing approaches

Existing approaches can largely by assigned to two categories with widely different requirements and use cases. *BLE Mesh* [4] is a networking standard for connection-less many-to-many communication based on BLE technology and specified by the Bluetooth SIG. Message transmission operates on managed flooding, which describes a network topology built on broadcasting instead of one-to-one transmissions. Every node receives the broadcasted message of every other node in direct radio range.
This approach has two notable problems by design. Flooding requires a lot of power to continuously scan for messages to enable a decent packet delivery ratio. While this is unproblematic for non-power-limited devices (e.g. smart home appliances), it is ill-fitted for mostly battery powered networks intended for long running use cases. Moreover, flooding can create congestions leading to increased packet collision probability [5] and thus higher power consumption. The second approach are *BLE Scatternets* which are a lose grouping of connection-based network topologies with the sole similarity, that they are constructed from several smaller networks (piconets) connected to each other via BLE.

Because BLE Mesh solves a different problem than PA-based network would, we can only compare them superficially. Both are connectionless and work with advertisements, but every BLE Mesh node needs to be awake permanently compared to PA's synchronized and extremely short wake time. This very basic topology premise of BLE Mesh makes it easier to use and manage, if you can affort the very high power consumption. As long as the amount of mesh members is lower than the connection limit, a PA-based network could create the same topology with a fraction of the power draw and message quantity, albeit with a slightly higher setup effort.

Scatternets would be kind of similar to PA-based network in that every scatternet is a subset of PA-based networks. Both have the same limit of synchronous data channels, but that

means missing PA's biggest advantage: It is asynchronous and thus way more flexible. Each device can send data to as many devices as are synchronized to it, with PAwR even providing the option to make all these transmission bidirectional, creating options for optimization or completely new approaches to power-constrained BLE networking.

VI. Conclusion

This paper presents Periodic Advertising, and more importantly Periodic Advertising with Response, as an alternative to existing approaches to BLE networking for multihop communication. BLE Periodic Advertising with Response enable asymmetric bidirectional connection-like relations, while being more flexible. This allows for significant energy savings for sensors-to-gateway scenarios. Because it hast different constraints and goals compared to BLE scatternets and BLE Mesh, it also paves the way for new mesh communication patterns.

As of yet, PAwR is recent enough, that use cases and implementations are still being worked on, but with the points presented in this paper, it might become a very important BLE feature.

References

[1] C. S. W. Group, "Bluetooth core specification," Jan. 2023. [Online]. Available: https://www.bluetooth.com/de/specifications/specs/core-specification-5-4/

[2] P. H. Kindt, M. Saur, M. Balszun, and S. Chakraborty, "Neighbor discovery latency in ble-like protocols," *IEEE Transactions on Mobile Computing*, vol. 17, no. 3, pp. 617–631, 2018.

[3] T. Zachariah, N. Jackson, B. Ghena, and P. Dutta, "Reliable: Towards reliable communication via bluetooth low energy advertisement networks," in *Proceedings of the 2022 International Conference On Embedded Wireless Systems and Networks*, ser. EWSN '22. New York, NY, USA: Association for Computing Machinery, 2023, p. 96–107.

[4] M. Woolley, "Bluetooth mesh networking – an introduction for developers," Dec. 2020. [Online]. Available: https://www.bluetooth.com/wp-content/uploads/2019/03/Mesh-Technology-Overview.pdf

[5] R. Rondón, A. Mahmood, S. Grimaldi, and M. Gidlund, "Understanding the performance of bluetooth mesh: Reliability, delay, and scalability analysis," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 2089–2101, 2020.

Secure Multi-hop Telemetry Broadcasts for UAV Swarm Communication

Randolf Rotta[1], Pavlo Mykytyn[1,2]
[1]Brandenburg University of Technology, Cottbus, Germany
[2]IHP - Leibniz Institute for High-Performance Microelectronics, Frankfurt (Oder), Germany

***Abstract*—Unmanned Aerial Vehicles (UAVs) are evolving as adaptable platforms for a wide range of applications such as precise inspections, emergency response, and remote sensing. Autonomous UAV swarms require efficient and stable communication during deployment for a successful mission execution. For instance, the periodic exchange of telemetry data between all swarm members provides the foundation for formation flight and collision avoidance. However, due to the mobility of the vehicles and instability of wireless transmissions, maintaining a secure and reliable all-to-all communication remains challenging. This paper investigates encrypted and authenticated multi-hop broadcast communication based on the transmission of custom IEEE 802.11 Wi-Fi data frames.**

***Index Terms*—Unmanned Aerial Vehicles, Multi-hop Networks, Vehicular Networks, Swarm Flight**

I. Introduction

To share data with a ground control station (GCS), other UAVs in a swarm, or other centralized infrastructure, UAVs rely on wireless communication. UAVs communicate with the base station to receive commands and transmit sensor data. Such point-to-point communication between a single vehicle and a GCS is straightforward.

However, when considering a swarm with multiple vehicles and ground stations, enabling efficient and secure communication becomes increasingly challenging. The communication security is critical because the swarm relies on it for cooperation, formation forming, or collision avoidance. However, wireless communications is particularly vulnerable to intentional and unintentional interference, jamming, interception, eavesdropping, and enables cyber-attacks targeting data privacy and integrity [1]. Furthermore, the dynamic nature of UAV networks with nodes continuously moving and re-establishing the connection between one another makes dependable communication links much more difficult to maintain. The dynamic and mobile nature of UAVs, combined with the limited range of wireless communication, necessitates the use of multi-hop communication techniques.

In this paper we focus on security and reliability aspects of all-to-all multi-hop broadcast communication between UAV swarm members and the GCS. Low-cost microprocessors with a few megabytes of RAM and integrated Wi-Fi radio open up the opportunity for new experiments with mesh protocols. One example is the Espressif ESP32 series, which allows broadcasting and receiving custom Wi-Fi frames without the need to associate to any Access Point. Securing these transmissions is left to be the protocol implementation.

This research has been funded partly by the Federal Ministry of Education and Research of Germany under grant numbers 16ES1131 and 16ES1128K. The authors are responsible for the content of this publication.

II. Related work

Mesh communication protocols such as B.A.T.M.A.N and Babel are often considered as a basis for UAV swarms [2]. Their design assumes that security mechanisms are handled on higher layers. Because of this assumption, mesh protocol layers are open to a variety of cyber-attacks. The authors in [3] mention that in multi-hop UAV swarm communication, active RF jamming and eavesdropping are among the most common cyber-attacks. Additionally, UAV swarms using the Robot Operating System (ROS 1) are vulnerable to a variety of cyber-attacks. ROS 2, on the other hand, has eliminated some of the issues mentioned above by introducing authentication and encryption based on a public key infrastructure.

Babel [4] proposes two optional mechanisms based on shared keys or Datagram Transport Layer Security (DTLS). SecBATMAN [5] proposes a security extension but lacks a dynamic key exchange. Studies like [6] argue, that dynamic key management schemes are necessary in order to secure these mesh protocols. Beyond mesh protocols that try to optimize forwarding routes through the network, Synchronous Flooding (SF) provides a much simpler alternative, c.f. [7]. These protocols synchronize the forwarding of broadcasts such that a n-hop broadcast needs just n consecutive time slots independent of the actual number of participating nodes. Unfortunately, existing implementations focus on Bluetooth Low Energy and IEEE 802.15.4 radios, which limits the available throughput. To our best knowledge no implementation based on IEEE 802.11 exists.

III. Multi-hop Telemetry Broadcasts

Point-to-point communication refers to a direct communication link established between two devices, such as an individual UAV and the GCS. Multi-point communication in UAV swarms connects a single GCS to multiple UAVs and multi-point meshes add multi-hop routing between GCS and UAVs. While this link to the ground provides essential connectivity, it is insufficient to facilitate seamless data exchange within the swarm. Mesh protocols like B.A.T.M.A.N and Babel focus on providing multi-hop point-to-point communication between arbitrary network nodes. This would be perfect, for example, for

two vehicles that cooperate on a task. However, cooperation between multiple vehicles requires to broadcast at least their position telemetry frequently enough to all the other UAVs. Flooding the mesh network with each vehicle's telemetry would work, but is inefficient because it re-broadcasts more often than necessary. This reduces the available throughput and increases the risk of colliding transmissions. Therefore, efficient multi-hop broadcast mechanisms are needed. We propose to revisit the flooding of route discovery messages in proactive mesh protocols like B.A.T.M.A.N and Babel. They use flooding to learn the best path to each possible destination node. Only the best next hop towards each destination node is stored, which is repeated at the next node until the destination node is reached. This approach provides an advantage of adjusting the path, while messages are already traveling. The next hops essentially form a collection tree toward each destination node. Our approach aims at inverting these trees into broadcast trees originating from that node. Thus, together with each outgoing message the node broadcasts its next hop table. The receiver of that message can then figure out its position in each broadcast tree and avoid unnecessary transmissions. This information allows each node to selectively forward pending messages from the queue based on the neighbor's needs. Typically, telemetry messages in the MAVLink protocol are much smaller than 256 bytes. Thus, during forwarding, multiple messages from different sources can be aggregated into a single IEEE 802.11 frame. A similar pattern can be achieved with ROS2-based communication. The underlying data distribution service allows to configure forwarding of published messages, for example, to multicast IP addresses. It can also be configured to receive such multicast messages. However, the Real-time Publish-Subscribe Protocol (RTPS) that is used between the ROS2 nodes is much more complex than MAVLink. Instead of maintaining a separate broadcast tree rooted at each node, a single spanning tree could be sufficient [8]. A broadcast message is re-broadcasted only in case if it was received via a neighbor in the spanning tree, it is not a leaf node, and the message was not re-broadcasted before. This approach combined with the loop avoidance techniques of the Babel protocol [4] should generate even more effective results.

IV. Security in UAV communication

UAV swarm communication is susceptible to RF jamming, Man-in-the-Middle (MITM), Eavesdropping, Traffic Analysis (TA), and Replay attacks. Eavesdropping and TA are passive cyber-attacks and require additional hardware in order to detect them. However, by integrating data encryption mechanisms, the effects of these cyber-attacks can be mitigated. The effects of a MITM attack, can be mitigated by integrating data authentication mechanisms. The effects of a Replay attack can be mitigated by integrating a timestamp to deem old and repeated messages invalid. Our approach is based on providing a secure and authenticated communication for all of the UAV swarm members. As a key exchange protocol we plan to use the Elliptic-curve Diffie–Hellman (ECDH) adapted to be used for multiple parties. Once all of the parties have generated their private and public keys and calculated a common shared secret (session key), we will authenticate the message using Hash-based Message Authentication Code (HMAC) based on Secure Hashing Algorithm (SHA-2) by generating a hash of the message together with the session key and appending the first 16 bytes of the hash to the end of the message, thus providing authentication of the contents of the broadcasted message. To encrypt the broadcasted message we will utilize the Advanced Encryption Standard (AES) with the 128-bit key. Each sent message will also include a timestamp to protect against Replay attacks and deem old messages invalid. By using this approach, the UAV swarm members can establish a common shared secret through ECDH key exchange, enabling secure and authenticated communication within the UAV swarm.

V. Conclusions

This paper has presented an idea for multi-hop telemetry broadcasts communication within a UAV swarm, specifically designed to enable fast, efficient, and secure mesh communication for mission execution and collision avoidance purposes. The proposed approach incorporates all-to-all broadcasts using flooding and message relaying. By leveraging the ECDH group key exchange protocol, drones establish a shared secret, ensuring secure communication channels within the mesh network. The use of AES-128 encryption guarantees the confidentiality of telemetry broadcasts, protecting sensitive information from unauthorized access. To ensure message integrity and authenticity, each broadcast message includes a HMAC-256 signature with a timestamp. This signature provides a reliable means to verify the origin and integrity of the message, preventing tampering or spoofing attempts.

References

[1] K.-Y. Tsao, T. Girdler, and V. G. Vassilakis, "A survey of cyber security threats and solutions for uav communications and flying ad-hoc networks," *Ad Hoc Networks*, vol. 133, p. 102894, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570870522000853

[2] A. Guillen-Perez, A.-M. Montoya, J.-C. Sanchez-Aarnoutse, and M.-D. Cano, "A comparative performance evaluation of routing protocols for flying ad-hoc networks in real conditions," *Applied Sciences*, vol. 11, no. 10, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/10/4363

[3] M. A. Lopez, M. Baddeley, W. T. Lunardi, A. Pandey, and J.-P. Giacalone, "Towards secure wireless mesh networks for uav swarm connectivity: Current threats, research, and opportunities," in *2021 17th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2021, pp. 319–326.

[4] J. Chroboczek and D. Schinazi, "The Babel Routing Protocol," RFC 8966, Jan. 2021. [Online]. Available: https://www.rfc-editor.org/info/rfc8966

[5] P. Racz, A. Lunn, and J. Paatero, "A security extension for ad-hoc routing protocols," in *7th Workshop on Wireless and Mobile Ad-Hoc Networks (WMAN 2013)*, 2013.

[6] M. Sbeiti and C. Wietfeld, "One stone two birds: On the security and routing in wireless mesh networks," in *2014 IEEE Wireless Communications and Networking Conference (WCNC)*, 2014, pp. 2486–2491.

[7] M. Zimmerling, L. Mottola, and S. Santini, "Synchronous transmissions in low-power wireless: A survey of communication protocols and network services," *ACM Comput. Surv.*, vol. 53, no. 6, dec 2020. [Online]. Available: https://doi.org/10.1145/3410159

[8] A. Jüttner and Á. Magi, "Tree based broadcast in ad hoc networks," *Mobile Networks and Applications*, vol. 10, pp. 753–762, 2005.

Impact of EU Regulations on Multi-Hop Wireless Sensor Networks

Jakub Maj, Krzysztof Piotrowski
IHP - Leibniz Institut für innovative Mikroelektronik
Frankfurt (Oder), Germany
{maj, piotrowski}@ihp-microelectronics.com

***Abstract*—In some cases, the use of Single-Hop technologies does not provide the required network coverage in Wireless Sensor Networks (WSNs), and additional solutions are required, such as Multi-Hop communication. When implementing solutions based on WSNs, the impact of all radio spectrum regulations should be taken into account to ensure equal band usage for all users while at the same time ensuring the reliable operation of the WSN. The regulations may vary by region or country. This article focuses on the European region and the 868–868.6 MHz band.**

***Index Terms*—WSN, Wireless Sensor Network, Multi-Hop, Duty Cycle, EU Regulations**

I. Introduction

Wireless Sensor Networks (WSNs) are solutions used for monitoring various variables in different environments. Wireless sensor nodes are easier and faster to deploy than conventional wired monitoring systems because every collected data is transmitted wirelessly, and wireless sensor nodes are often battery-powered or use energy harvesting. Nowadays, WSNs are applied in many fields, such as smart cities, smart grids, and environmental monitoring.

WSNs often use unlicensed bands, and must thus share access to radio spectrum with other systems. The large number of WSNs operating at the same time in the same bands forced the implementation of regulations to prevent continuous transmission or the transmission of harmful interference and ensure fair access to the spectrum for all WSNs. Regulations are region-, country-, and band-dependent. This paper focuses on the impact of EU (European Union) regulations in the 868–868.6 MHz band [1], [2].

II. Related Works

In [3], the authors gave a detailed overview of EU Regulations in the 863–870 MHz bands in the context of Low-Power Wide Area Network (LPWAN) technologies. The authors described available frequency bands, duty cycles, transmission power limits, polite spectrum access, EU regulatory institutions, and the recent evolution of regulations. In the second part, the authors describe how several of the most popular LPWANs cope with EU regulations, before finally moving on to describe future work. In [4], the authors present a MAC protocol that uses the special frame to set up the path and transmit data packets in a single duty cycle with the use of Multi-Hop transmission. The authors compare the average packet latency of their MAC, analyzed in a probabilistic manner, with simulation in Network Simulator 2 (NS-2). On the basis of analyses and simulations, the optimal duty cycle was mathematically calculated, minimizing energy consumption, while maintaining the assumed delay.

III. SmartRiver

A. Environmental Monitoring System

SmartRiver [5], as a WSN-based system for environmental monitoring working in the European region in the 868-868,6 MHz band, is the starting point for considering the impact of the duty cycle on Multi-Hop WSN. The nodes (End Nodes) used in the SmartRiver project are responsible for the monitoring of hydrologic, weather, and air quality parameters. Hydrologic nodes are deployed on flooding walls, flooding fields, rivers, and water tanks. In contrast, weather and air quality nodes are deployed on lanterns in the cities of Słubice and Frankfurt (Oder), as well as in the surrounding suburban areas. This solution causes the measurement area to spread over a large area, approximately 64 square kilometers. The distribution of nodes over such a large area causes a problem with measured data transfer to the sink because of the distance between nodes and the sink. An additional difficulty is the urban environment and the hilly terrain.

B. Multi-Hop transmission

The solution for large distances between nodes was to deploy additional repeaters (Relay Nodes) between nodes and sink (Gateway) and implement a Multi-Hop transmission method. The Multi-Hop transmission allows relaying data packets through other nodes or repeaters, thereby increasing network coverage. In addition to Multi-Hop transmission, packets are transmitted using a Simple Link Long Range physical layer encoding technique, that exchanges data rate for the receiver's sensitivity, which extends transmission distance between single nodes and reduces the repeater's deployment density. In the SmartRiver project, Simple Link Long Range is set to work in the 868–868.6 MHz band. In the MAC (Medium Access Control) layer, CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) was implemented to reduce the chance of packet interference between neighboring nodes. The WSN deployed in the SmartRiver project operates in a tree architecture. Nodes can build a network autonomously

and find paths in a tree, using the Asynchronous BFS (Breadth First Search) [6] algorithm with Discovery and Join packets.

IV. EU regulations on harmonization of the radio spectrum

The EU regulates both transmission power limits and duty cycle limits. Transmission power limits are given as an Effective Radiated Power (ERP), which is the total power in watts that would have to be radiated by a half-wave dipole to give the same radiation intensity as the actual antenna, at the same distance and in the direction of the antenna's strongest beam. The duty cycle *DC* is the ratio of the total transmission time $\sum T_{OnAir}$ to observation time T_{Obs} (usually one hour), as shown in (1).

$$DC = \frac{\sum T_{OnAir}}{T_{Obs}} \tag{1}$$

The duty cycle is measured within a single band, which means that simultaneous transmission on multiple bands increases the duty cycle limits for a single device. The regulations also do not impose how the transmission is to be distributed in time. The available time can be exploited at the beginning of observation time, or individual transmissions can be staggered. Duty cycle limits can be loosened by using techniques to access the spectrum and mitigate interference with similar performance to those adopted under Directive 2014/53/EU [7]. Those techniques determine whether the channel is free (Listen Before Talk, LBT) and avoid transmission on already occupied channels (Adaptive Frequency Agility, AFA). Those techniques extend the maximum transmission time to 100 s per observation time per 200 kHz of the spectrum. The band chosen to be used in the SmartRiver project is band number 48. The chosen band allows transmission for all kinds of radio devices and has a limitation of 1% for a duty cycle and 25 mW ERP for the transmission power limit.

V. Impact of EU regulations on Multi-Hop Transmission

The proposed Multi-Hop solution described in section III can have a problem meeting the EU regulation in the 868–868.6 MHz band. This is due to several factors, such as a large network with lots of devices, Multi-Hop transmission, and a low data rate physical layer.

Relay Nodes, besides relaying, can also transmit their own data packets, thus causing a lot of packets to be transmitted through the network. The large number of packets transmitted in the network combined with Multi-Hop transmission causes the nodes closest to the sink to become bottlenecks because they need to relay the largest amount of data. Because of the duty cycle restrictions, those nodes become even more bottlenecked, because they cannot transmit packets infinitely. The described situation, combined with a low data rate physical layer, may lead to exceeding the duty cycle and blocking the transmission, and consequently to data loss due to overflow of buffers storing packets on nodes.

Apart from data loss, waiting for the node to transmit will cause long delays in transmitting end-to-end (e2e) packets. Transmission delays have a particular impact on the network-building phase, as well as data packet transmissions at critical moments where data must be sent to the sink immediately.

Maximum transmission power regulations also affect the operation of Multi-Hop networks, because lower transmission power reduces the transmission range between single nodes, which necessitates the deployment of more nodes, which leads to more traffic in the network, and consequently to transmission delays or loss of packets.

VI. Conclusions

As described in section V, EU regulations can have a significant impact on WSNs using Multi-Hop transmission. Both the duty cycle limits and transmission power limits affect those solutions. Future research will focus on finding the most effective way to measure and control the duty cycle. At the moment, two concepts of duty cycle measurement and control are being considered, including one based on the use of the LBT technique in combination with AFA. In addition, the impact of EU regulations on the effectiveness of WSNs based on Multi-Hop transmission will be examined in practice.

Acknowledgment

This work was supported by the European Regional Development Fund within the BB-PL INTERREG V A 2014-2020 Programme, "reducing barriers - using the common strengths", project SmartRiver, grant number 85029892. The funding institution had no role in the design of the study, the collection, analyses, or interpretation of data, in the writing of the manuscript, or in the decision to publish the results.

References

[1] "COMMISSION IMPLEMENTING DECISION (EU) 2017/1483 of 8 August 2017 amending Decision 2006/771/EC on harmonisation of the radio spectrum for use by short-range devices and repealing Decision 2006/804/EC," Official Journal of the European Union, August 2017.

[2] "CEPT/ECC, ERC Recommendation 70-03 Relating to the use of short range devices (SRD).", Subsequent amendments 10 June 2022, editorial update 17 February 2023, 1997.

[3] M. Saelens, J. Hoebeke, A. Shahid, and E. De Poorter, "Impact of EU duty cycle and transmission power limitations for sub-GHz LPWAN SRDs: an overview and future challenges," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, Sep. 2019.

[4] K. -T. Cho and S. Bahk, "Duty cycle optimization for a multi hop transmission method in wireless sensor networks," *IEEE Communications Letters*, vol. 14, no. 3, pp. 269-271, March 2010.

[5] E. Michta, P. Powroznik, R. Rybski, R. Szulim, K. Piotrowski, U. Kolodziejczyk and J. Kostecki, "Flood embankments monitoring system in on-line mode," *Measurement Systems in Theory and in Practice*, ed. by R. Rybski, Zielona Gora, Institute of Metrology, Electronics and Computer Science, University of Zielona Gora, 2020, pp. 139-158.

[6] J. Aspnes, "Distributed breadth-first search," *Notes on Theory of Distributed Systems*, arXiv:2001.04235v3 [cs.DC], 2020, pp. 27-28. [Online]. Available: https://arxiv.org/abs/2001.04235v3.

[7] "Directive 2014/53/EU of the European Parliament and of the Council of 16 April 2014 on the harmonisation of the laws of the Member States relating to the making available on the market of radio equipment and repealing Directive 1999/5/EC Text with EEA relevance," Official Journal of the European Union L153/52, April 2014.

Design of a Supporting Protocol for a Broadcast Protocol in Wireless Multi Hop Networks

Kai Kientopf, Jonas Rebbelmund, Mesut Güneş
Communication and Networked Systems (ComSys)
Faculty of Computer Science
Otto-von-Guericke University Magdeburg
Universitätsplatz 2, 39106 Magdeburg, Germany

***Abstract*—This paper introduces NABB Network Discovery (NABB-ND), a supporting protocol for the neighborhood analysis based broadcasting (NABB) protocol in Wireless Multi Hop Networks (WMHNs). NABB-ND addresses challenges in real-world scenarios, including the handling of asymmetric links. It uses heartbeat messages to determine the 2-hop neighborhood topology and to identify asymmetric links. A reduction in network traffic is achieved by combining NABB with NABB-ND. NABB-ND provides valuable enhancements to NABB, improving its applicability in practical WMHN deployments.**

***Index Terms*—Broadcast, WMHN, WSN**

I. Introduction

We developed a protocol named neighborhood analysis based broadcasting (NABB) for broadcasting in Wireless Multi Hop Network (WMHN) [1]. The focus of the protocol is the use of local neighborhood information to reduce the network traffic of broadcasts while maintaining a good network coverage. Locally, on one node of the WMHN, the protocol uses the topology information of the 2-hop-neighborhood (2HN) and the metadata of messages received from the nodes of the 1-hop-neighborhood (1HN). Based on that information the protocol calculates a probability and make a probabilistic decision if the broadcast message should be repeated by the node.

The protocol was evaluated with simulations using the OMNeT++ and INET framework. Within the simulation we assumed an ideal network (without packet loss) and a WiFi stack, but without interference or obstacles. Though, we get very comparable results. In real-world scenarios, especially with low cost hardware, like it is common in IoT devices, has challenges that were not addressed in the protocol design itself.

Excluding the collection of the neighborhood information was a design decision, to simplify the protocol and may take advantage of information that the lower layer might provide depending on the network stack. The protocol also assumes symmetrical connections or rather leave the exclusion of asymmetric connections to the same mechanism that collects the neighborhood information. Furthermore, packet loss also effects the protocol's calculations.

In this paper we describe our approaches to address the problems the NABB protocol would face in real-world WMHNs. In Section II we present related work. Section III describes the design of the supporting protocol as a solution to the problems discussed in this section. Finally, we conclude this paper and propose future work on the topic in Section IV.

II. Related Work

Three studies were conducted based on NABB, which address different challenges the protocol encounters in real-world WMHNs.

The results of the first study leads to the conclusion that most network traffic is avoided because NABB can recognize that all nodes in the 1HN already got the broadcast message. Therefore, Wirth changed the probabilistic nature of NABB to a timeout based approach: timeout based NABB (tNABB) [2]. With these changes he maintained a low network traffic, while providing a high network coverage. The approach was also evaluated in simulations.

Rebbelmund implemented NABB for the GNRC network stack (GNRC) of RIOT OS (RIOT) and evaluated the protocol in the Magdeburg Internet of Things Lab (MIoT-Lab) with IEEE 802.15.4s hardware [3]. He developed several strategies to counter the challenges described in Section I. Some approaches of the work we also incorporate in NABB Network Discovery (NABB-ND) protocol in section III.

Moosdorf implemented NABB for WiFi under Linux [4]. He also implement strategies to avoid problems of NABB in real-world WMHNs. The main focus of his work is the combination of NABB with Network Coding (NC).

III. Supporting Protocol

In this section we address our solutions for the problems from section I with an supporting protocol: NABB Network Discovery (NABB-ND). A possible structure of NABB-ND is depicted in Figure 1.

A. Neighborhood Information

To determine the topology of the 2HN a node x broadcasts a message locally (to the 1HN) with a list of all known neighbors from its 1HN. Therefor we use NABB-ND and call this message a Heartbeat (HB). If any node y in the 1HN does not know about node x, y will also send out its HB. To avoid collisions, there should be a random timeout before each HB. To cope with changing network topologies, all nodes should periodically send out HBs to inform their neighbors about their existence. If node x does not receive the expected HB

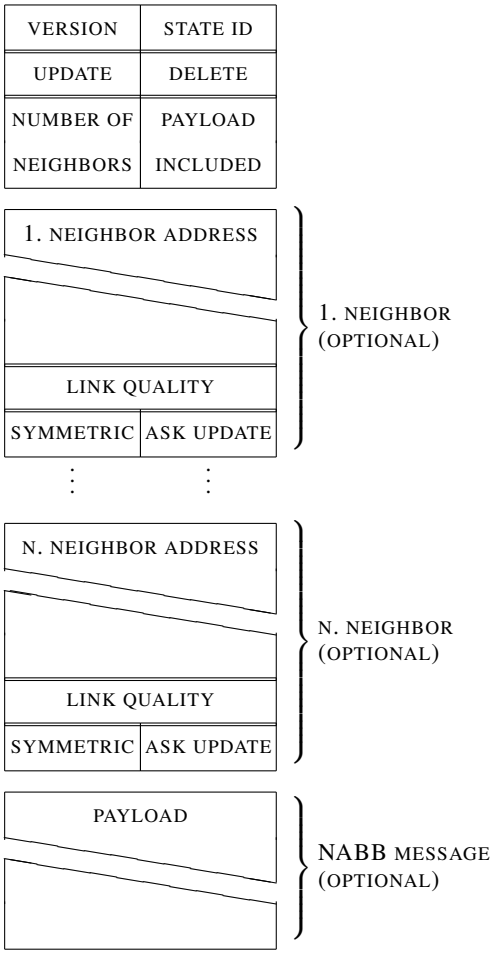


Figure 1: Example for the Heartbeat (HB) structure of the NABB Network Discovery (NABB-ND) protocol. The size of the fields are not proportional.

from y within the period it deletes y from its list of 1HN and sends an HB. In the case the HB from y has just been lost due to packet loss, y will be informed through the HB from x and should send out its own HB again. A node should also send out an HB, when it detects a new node within its 1HN. From the HB messages, each node can construct a graph of its 2HN.

B. Asymmetric Connections

NABB-ND can also be used to identify asymmetric links within the 1HN. A node x sending out an HB, can expect to be included in any HB it receives, if the connections are symmetric. To deal with asymmetric connections, the list with the 1HN should be extended with a boolean value indicating whether the connection is symmetric. If x receives a HB from y that includes x in the 1HN list, then x should mark the connection to y as symmetric. The HB from a new node z that does not include x should result in an asymmetric link mark in the 1HN list of node x. In such a case, x should send out its HB once. If the connection between x and z is symmetric, node z includes x in its 1HN list and sends out an HB again. With the additional information, a directed graph of the 2HN can be constructed and asymmetric links can simply be ignored by NABB or NABB can be redesigned for asymmetric links.

C. Link Quality

Another field can be added to the list of 1HN: LINK QUALITY. Different metrics can be used for this field, e.g. the Received Signal Strength Indicator (RSSI) or the Packet Delivery Ratio (PDR). To handle packet loss, NABB can simply ignore links under/over a certain threshold. Alternatively NABB can also be modified to use the additional information. A node should just update its link quality if the changes exceed a certain value to avoid unnecessary traffic.

D. Overhead

One goal of NABB is reduced network traffic. The additional overhead from NABB-ND is necessary to ensure the function of NABB. Stacking NABB on top of NABB-ND can reduce the overhead. By combining the periodical HB with a regular NABB message, the number of messages get reduced and with this also the overhead of underlying protocol headers that need airtime.

To further reduce traffic, NABB-ND can include an increasing STATE ID number. If there is no change for a node x, the HB can include the last STATE ID number without the whole 1HN, to indicate that there are no changes. If a node y is new in the 1HN or missed the old HB with the 1HN information, it needs a way to request the 1HN information. An ASK UPDATE flag at the node x in the HB of node y can signal the node x to send a complete HB.

With the UPDATE flag in the header the overhead can be further reduced. The idea is that the the last HB with the UPDATE flag disabled is used as the base information i and all subsequent HBs with UPDATE or DELETE flag enabled and increasing STATE ID update i. To allow nodes to be deleted from the list, the DELETE flag results in a removal of all neighbors listed in i. It is important that there are no gaps in the STATE ID numbers, otherwise a node must ask for an update. Updating information should only be performed if periodical HB are send out.

IV. Summary

We designed the NABB-ND supporting protocol for the NABB protocol. NABB-ND provides the 2HN information to the NABB protocol, which uses it as the basis for broadcasting decisions. NABB-ND also addresses several wireless communication issues that were not directly addressed in NABB. A future step is modify of NABB to make use of asymmetric links and take advantage from link quality metrics.

References

[1] Kai Kientopf and Mesut Guenes. Analyze the 2-hop-neighborhood for efficient broadcasting in wireless multi hop networks. In *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*. IEEE, December 2017. doi:10.1109/glocom.2017.8253951.

[2] Dominic Wirth. Effiziente Verteilung von Broadcast-Nachrichten mit Hilfe dynamischer Timeouts, 2018.

[3] Jonas Rebbelmund. Experimentelle Evaluation des NABB-Protokolls im MIoT-Lab, 2021.

[4] Agostino Moosdorf. Efficient broadcasting in wireless multi hop networks, 2023.

Investigating the Effects of Precipitation on the Reliability of Lossy LoRaWAN Links

Daniel Szafranski
Department of Informatics
Clausthal University of Technology, Germany
daniel.szafranski@tu-clausthal.de

Andreas Reinhardt
Department of Informatics
Clausthal University of Technology, Germany
reinhardt@ieee.org

***Abstract*—The effects of climate change on the planet have become more and more noticeable in the last decades. An increase in extreme weather events, such as flooding and droughts, can be observed all around the world. The usage of monitoring systems can help to understand, analyze, and ultimately predict such weather extremes, e.g., by reliably monitoring important indicators like precipitation, river levels, and soil moisture. Since these parameters often exhibit strong variations, even within small geographical boundaries and short time intervals, monitoring must be done at a high spatial and temporal resolution. This poses a significant challenge, because an accurate monitoring requires sensors to be deployed in a large area, which might furthermore be inaccessible or hard to reach, mandating the use of wireless devices. Wireless link reliability can be additionally aggravated by harsh weather events like heavy rainfall, even though monitoring is particularly crucial during such extreme weather events. The relation between the prevailing weather conditions and the corresponding link quality has only been researched to a limited extent so far. We thus analyze long-term link quality data from a Long Range Wide Area Network (LoRaWAN)-based sensor network in the German Harz mountain range, in order to assess the relation between precipitation events and the reliability of links. Our evaluation indicates that lossy links indeed suffer from precipitation, and packet loss is indeed greater during rain events.**

***Index Terms*—LoRa, LoRaWAN, Weather events, Reliability**

I. Introduction

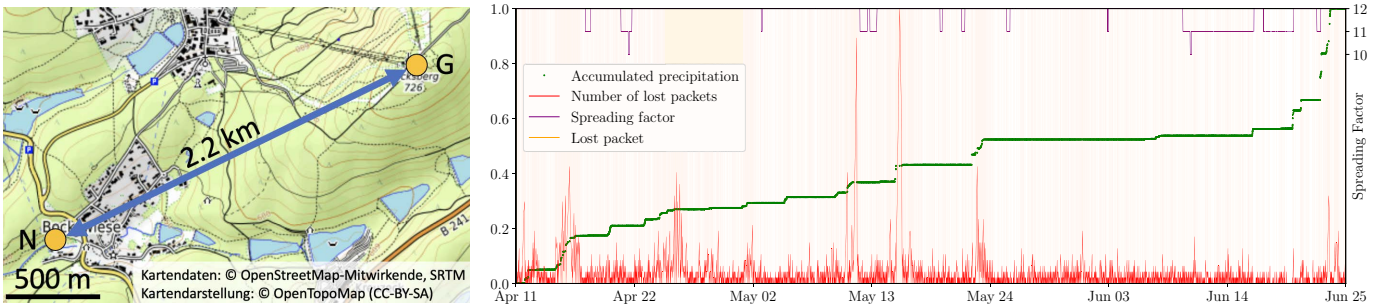
Climate change is known to be the cause of many weather extremes, like heatwaves and torrential rain [1]. One example for such an extreme precipitation event is the German *Ahrtal flood* that happened in July 2021. Within 24 hours, rainfall accumulated up to 150 liters per square meter, whereas the long-term average rainfall for the same region is less than 70 liters per square meter for the entire month of July [2]. The extreme precipitation resulted in one of Germany's most severe catastrophes of the last decades. More than 180 people died during the flood and the financial damages added up to nearly 30bn Euro [2]. Related works indicate that such extreme events may accumulate in future due to climate change [1, 3].

The research project *EXDIMUM* (Extreme Weather Management with Digital Multiscale Methods) aims to understand and analyze weather extremes by collecting and modeling data on multiple scales. One very crucial data source under consideration in the project are terrestrial data, as they can provide information about important hydrological parameters and indicators, like precipitation, river levels, and soil moisture. Due to their nature, these parameters can vary significantly over even small areas and short periods of time. For correct hydrological modeling, it is crucial to capture these local variations reliably. In order to do so, it is necessary for the measurement system to achieve a high spatial and temporal resolution. This is challenging in multiple regards. First, to achieve the desired high spatial resolution, sensors need to be deployed over a wide area, which is often also hard to reach. Secondly, the sensor nodes potentially need to transmit their measurements over long distances in the range of several kilometers, calling for wireless networking. Furthermore, the functionality of all nodes must be ensured for a long period of time with the limited energy capabilities from batteries. And lastly, data captured under extreme weather events is of particular high importance, as it contains valuable information for hydrological analysis and modeling. Therefore, the measurement system must be able to operate and collect data reliably even under extreme weather conditions.

Even though the latter requirement is especially relevant, the impact of precipitation on wireless links in sensor networks has not seen extensive consideration in previous research (cf. Section II). We thus analyze data that was captured over several months from a LoRaWAN-based sensor network, introduced in Section III, which is operated in the Harz mountains in Germany. In order to evaluate the reliability of wireless sensor nodes under harsh environmental conditions, we specifically focus on lossy links (i.e., links that can both increase and decrease their packet reception rates in response to ambient conditions), in order to assess whether they are negatively impacted by precipitation events (cf. Section IV).

II. Related work

Since Long Range (LoRa) nodes are often deployed outdoors and thus prone to changing weather conditions, several related works have investigated the effects of weather on the link quality in LoRa networks. Most of the works agree that weather conditions have an impact on the link quality. However, the impact of individual weather characteristics is controversially discussed in recent literature. On the one hand, a correlation between temperature and link has been described in multiple works, and there is consensus that higher temperatures decrease the Received Signal Strength Indication (RSSI) and Packet Reception Rate (PRR) [4–6]. On the other



(a) Topography of the LoRaWAN deployment.

(b) Precipitation data received from the node during April 11 - June 25, 2023.

Fig. 1. A topographic map of the analyzed LoRaWAN deployment is shown in subfigure (a). The node is marked with N, while the gateway is marked with G. Subfigure (b) shows the received accumulated precipitation data from Node N as well as the number of lost packets on a normalized scale. The Spreading Factor is also shown and varies between 10 and 12.

hand, no unequivocal statement has been determined for the impact of precipitation. The authors in [7] evaluated the RSSI for LoRa transmissions under tropical heavy rain of up to 180 mm/h and their results showed no impact on the RSSI or PRR. In contrast to this, [8] evaluates the RSSI for LoRa transmissions on an offshore sea farm. The results clearly show a correlation between precipitation and the RSSI. Furthermore, even a differentiation in RSSI between light, medium, and heavy rain is visible in their measurements. These results were also confirmed by [9], where Wang et al. conducted experiments on the performance of LoRa on a campus and found that even light rain reduced the PRR significantly by almost 20 %. Another interesting study was carried out by Ameloot et al. in [10]. The authors determined that the antenna placement is a crucial point in the correlation of precipitation and RSSI. Two different antenna locations, outdoor and indoor were compared and only the outdoor antenna was suffering from dips in RSSI during rain while the indoor antenna did not show any significant variations. The authors assume that the humidity on the outdoor antenna changed its behavior, resulting in an antenna mismatch and RSSI drops.

Overall, related works indicate that weather indeed has an impact on the link quality of LoRa networks and should be considered. The specific impact on any individual link is, however, highly specific to the deployment scenario. Previously weak links seem to be particularly prone to adverse weather conditions, since the RSSI may drop more easily below the receiver sensitivity threshold, resulting in packet losses. We therefore analyze and investigate the effects of weather conditions on the LoRaWAN deployment in the Harz mountains in order to assess its reliability under different weather conditions.

III. Monitoring System

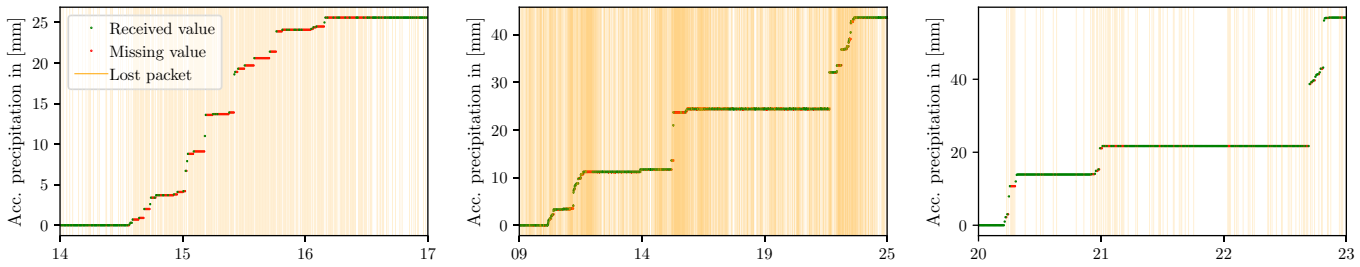
The analyzed LoRaWAN deployment is located in the Upper Harz in Germany. For an in-depth analysis we decided to focus on a node that is placed at the edge of the reception range and thus potentially particularly vulnerable to adverse weather events. The node (N) and gateway (G) locations are marked in the topographic map in Fig. 1a. The gateway is located on top of a mountain, approximately 720 m above sea level and the node is placed in the valley, approximately 540 m above sea level. The distance in between is 2.2 km and there is no line-of-sight path. A *MultiTech Conduit* [11] gateway and an *ELSYS ELT-2* [12] node were used. The node transmits data every 10 minutes using the LoRa transmission parameters as shown in Table I and uses an Adaptive Data Rate (ADR) feature, which allows for dynamic adjustments of the Spreading Factor (SF).

TABLE I
Used LoRa transmission parameters.

Frequency	Spreading Factor	Bandwidth	Coding Rate
867.1 - 868.5 Mhz	10 - 12	125 kHz	4/5

IV. Evaluation

We evaluated the data collected during the timespan of 74 days, between April 11 and June 24, 2023. During this period, 6228 samples were received and 4479 were lost, resulting in a mean PRR of 58 %. The RSSI ranged from -100 to -119 dbm, averaging at -116 dbm. The accumulated precipitation values and number of lost packets are shown in Fig. 1b. For the sake of visual clarity, all values have been normalized to a scale from 0 to 1. Additionally, vertical lines are used to mark all sample points that have been lost. Thus, their intensity gives an impression of the accumulation of consecutive packet losses. Furthermore, the SF is also shown and varies between 10 and 12. The diagram clearly shows two things. First, as mentioned earlier, the PRR is comparatively small, resulting in packet losses across the entire time period. Secondly, more packet losses occur especially during rain events (as indicated by a rise of the *precipitation* line). A similar behavior can be observed with the SF, as it stays at the maximum value of 12 during rainy periods with high packet losses and reduces during periods of no rain and minor packet loss. This is expected, since ADR is specifically designed to adapt the SF based on the link quality. In order to get a more detailed impression of the correlation between precipitation and link quality, we analyzed several rain events with different characteristics and high packet loss rates. The results are shown in Fig. 2 and confirm our previous observation: The



(a) April 14-17, 2023. (b) May 9-25, 2023. (c) June 20-23, 2023.

Fig. 2. A detailed view on the received precipitation data as well as missing samples for three different time periods with particular high packet loss rates.

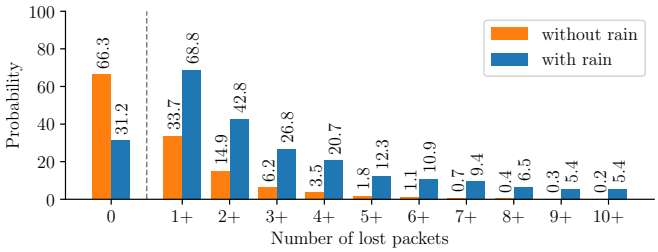


Fig. 3. Probability for the loss of different-length sequences of packets.

number of lost packets increases during rain events, as can be observed as an increase of the density of missing sample points. To quantify the results, we analyzed the probability for packet losses for times with and without rain events. Therefore, we assumed a rain event if an increase in the accumulated precipitation was measured between two received samples. The results are presented in Fig. 3 and clearly show that the probability for packet losses is increased during rain events. While there is a chance of 66 % that no packet is lost when there is no rain, it decreases to 31 % in case of rain. The probability for at least one missing sample is 34 % without rain and increases up to 69 % during rain. For larger numbers of packet losses, the probability decreases exponentially for both (rain and no rain), however, the probability stays consistently higher for samples collected during rain.

V. Conclusion and Outlook

In this paper we investigated the impact of rain on the link quality of a LoRaWAN-based sensor network. By way of an analysis of data captured during several months in the Harz mountains in Germany, we have correlated rainfall with the ensuing packet losses. In our evaluation, we have focused on a node with particular weak RSSI and PRR, given that such nodes are especially susceptible to even minor deteriorations of the link quality. Our results, contrary to some related works, indicate that precipitation indeed has a significant impact on the link quality of lossy LoRaWAN nodes. We observed a decrease in the probability for no packet losses from 66 % down to 31 % in case of rain. Furthermore, the probability for losses of subsequently transmitted packets is consistently higher during rain. For our future work, we plan to investigate the observed correlations in more detail and validate our first results on a larger dataset and with more nodes. In the future, Wireless Sensor Networks (WSNs) could benefit from considering weather conditions and dynamically adapting to them. Potential adaptions may include changes of transmissions parameters or alternative routing approaches. Also, the prediction of lossy links (due to bad weather conditions) may offer interesting research ideas, especially in regions where adverse weather conditions occur regularly.

Acknowledgment

This work was supported by the German Federal Ministry of Education and Research (BMBF) within the scope of the project EXDIMUM.

References

[1] D. Coumou and S. Rahmstorf, “A decade of weather extremes,” *Nature climate change*, vol. 2, no. 7, pp. 491–496, 2012.
[2] Bundesministerium des Inneren und für Heimat, Bundesmisterium der Finanzen, “Bericht zur hochwasserkatastrophe 2021: Katastrophenhilfe, wiederaufbau und evaluierungsprozesse,” 2022.
[3] C. B. Field, V. Barros, T. F. Stocker, and Q. Dahe, *Managing the risks of extreme events and disasters to advance climate change adaptation: special report of the intergovernmental panel on climate change*. Cambridge University Press, 2012.
[4] M. Cattani, C. A. Boano, and K. Römer, “An experimental evaluation of the reliability of lora long-range low-power wireless communication,” *Journal of Sensor and Actuator Networks*, vol. 6, no. 2, p. 7, 2017.
[5] C. A. Boano, M. Cattani, and K. Römer, “Impact of temperature variations on the reliability of lora,” in *Proc. 7th Int. Conf. Sensor Netw.*, 2018, pp. 39–50.
[6] N. Jeftenić, M. Simić, and Z. Stamenković, “Impact of environmental parameters on snr and rss in lorawan,” in *International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*. IEEE, 2020, pp. 1–6.
[7] O. Elijah, S. K. A. Rahim, V. Sittakul, A. M. Al-Samman, M. Cheffena, J. B. Din, and A. R. Tharek, “Effect of weather condition on lora iot communication technology in a tropical region: Malaysia,” *IEEE Access*, vol. 9, pp. 72 835–72 843, 2021.
[8] L. Parri, S. Parrino, G. Peruzzi, and A. Pozzebon, “Offshore lorawan networking: Transmission performances analysis under different environmental conditions,” *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–10, 2020.
[9] S.-Y. Wang, Y.-R. Chen, T.-Y. Chen, C.-H. Chang, Y.-H. Cheng, C.-C. Hsu, and Y.-B. Lin, “Performance of lora-based iot applications on campus,” in *2017 IEEE 86th vehicular technology conference (VTC-Fall)*. IEEE, 2017, pp. 1–6.
[10] T. Ameloot, P. Van Torre, and H. Rogier, “A compact low-power lora iot sensor node with extended dynamic range for channel measurements,” *Sensors*, vol. 18, no. 7, p. 2137, 2018.
[11] MULTITECH, *MultiTech Conduit® IP67 Base Station*, 2022.
[12] ELSYS, *Operating Manual - Elsys ELT-2*, 2020.

SDR Based 5G NR Scanner for WSN

Kaya Runge*, Fabian John*, Horst Hellbrück*

*Technische Hochschule Lübeck – University of Applied Sciences, Germany
Department of Electrical Engineering and Computer Science, Center of Excellence CoSA
Email: {kaya.runge, fabian.john, horst.hellbrueck}@th-luebeck.de

***Abstract*—5G networks enable massive machine-type communication for wireless sensor networks in various applications. However, deploying and operating 5G networks requires monitoring tools for debugging which should be cheap and easy to use. These are not yet available. We propose a flexible 5G network scanner based on a software defined radio and MATLAB, which scans the environment for 5G networks and decodes 5G master and system information block. Our scanner has a smart scanning routine that adapts the gain and compensates the center frequency offset of the SDR. We also present a modular and distributed system architecture, allowing flexible post-processing. We demonstrate our scanner in a 5G Testbed and show how it supports network configuration and debugging. Our tool is simple, accessible, and flexibly adaptable for different scenarios and purposes.**

***Index Terms*—Wireless Sensor Networks (WSN), 5G, NR, Standalone (SA), Monitoring, Campus Network**

I. Introduction

Massive Machine-Type Communication (mMTC) is one of the three main services provided with 5G [1]. Wireless Sensor Networks (WSN) in mMTC support all kinds of modern applications. This feature introduces a multiple of new 5G private networks [2], [3]. Campus networks for research or IoT networks for industrial purposes are only two examples. To set up and operate these networks correctly, easily accessible and cheap monitoring tools are necessary.

Before 5G, the mobile communication sector was driven by only a few large companies. Therefore, available monitoring tools are not a mass product and thereby expensive and designed for experts. Monitoring based on open-source and COTS is not available. We introduce a scanning setup with a software defined radio (SDR) as the receiver and MATLAB for data processing.

The contributions of this paper are:

- We propose a 5G network scanner based on accessible and adaptable hardware and software.
- We present a smart scanning routine with gain adaption and center frequency uncertainty compensation.
- We propose a modularized and distributed system architecture for flexible post-processing.

The rest of the paper is structured as follows. Section II describes our idea for a simple and adaptable 5G monitoring script. In Section III we shortly present measured results from a scan in our 5G Testbed. Finally, we conclude our work.

II. Idea and Realization

Continuous scanning of physical layer improves deployment progress and operation of WSN with private 5G networks. Since most of the user equipment (UE) in WSN is headless, we need to verify the correctness of network configuration under different conditions.

We propose a tool, that scans the environment for 5G networks using an SDR and processes the measurements with MATLAB. Decoding Master Information Block (MIB) and System Information Block (SIB) of the 5G network is based on a MATLAB example [4], that was adapted to meet our purposes. The structure of our script is shown in Figure 1.

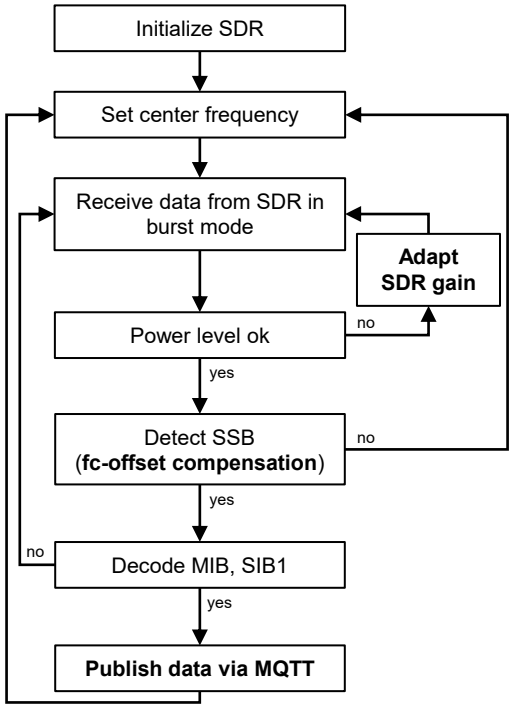


Fig. 1. Structure of the script for monitoring a 5G network

First, we initialize the SDR and set the parameters. Second, we scan the frequency range in burst mode and check the power level. Third, we adjust the gain of the SDR to avoid low or high-power signals.

Next, the script continues and tries to identify the signal synchronization block (SSB) of a 5G network with [4]. In this step, we propose an automatic frequency-offset compensation.

The 5G network and the monitoring setup use different systems, which may cause their clocks to diverge significantly. This affects the alignment of the center frequencies. As a result, no signal is detected, even though we are looking at the correct center frequency. We use the correlation results of the SSB detection to show whether a 5G network is available and adjust the offset compensation accordingly.

If we cannot detect any SSB, we select the next center frequency. Otherwise, we use [4] to decode the MIB, the SIB1 and further network data. If the decoding fails, we select the next data frame received in the burst and decode again. If decoding succeeds, we send the data to MQTT for further processing and visualization. We split MIB, SIB1 and other data into dedicated MQTT topics.

III. Results

To determine whether our monitoring script works properly, we evaluate results with the hardware setup shown in Figure 2.

Fig. 2. Hardware setup for 5G monitoring

We perform evaluation measurements in our 5G Testbed with the SDR B200 and MATLAB script running on PC. The 5G network is deployed according to the reference deployment in [5] set to a center frequency of 3.74016 GHz. A second MATLAB script visualizes scan results that are provided via MQTT. Figure 3 shows two example plots. The upper plot is the spectrogram at the center frequency where the network was detected. The SSB periodicity is set to 5 ms in the Testbed and is visualized in the spectrogram, that also shows data traffic between 0 ms and $\approx$ 15 ms. At $\approx$ 18 ms the SSB is observed without data traffic. The lower plot shows the decoded physical downlink channels (PDCCH and PDSCH) [1].

IV. Conclusion

In this paper, we introduced a simple, accessible and adaptable 5G monitoring solution based on a script and some cheap standard equipment. The solution provides a smart scanning routine and is modular. Our results show the 5G network configuration, such as the SSB periodicity or center frequency. Our tool supports the deployment and debugging, especially with headless devices in WSN.

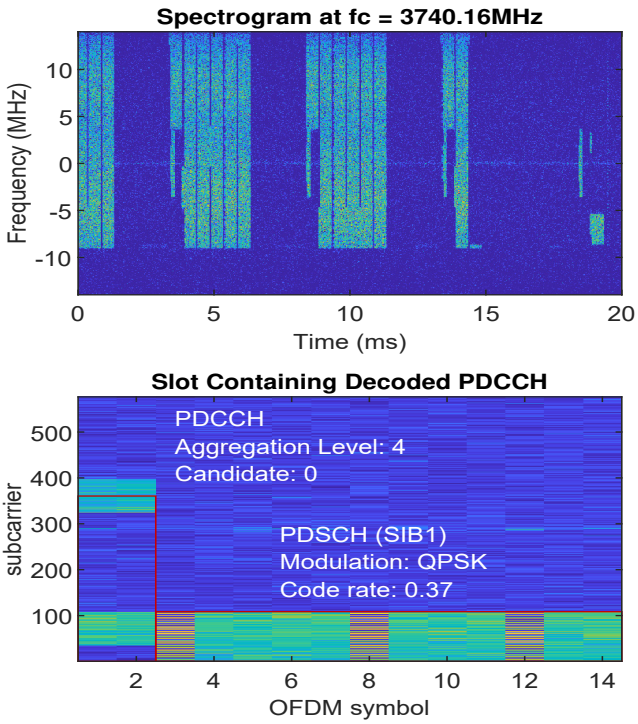


Fig. 3. Plotted results

Currently, the SIB1 is decoded as a binary message. We plan to add more SIB1 parsing in the future so that we can display the network parameters in a human readable format.

Acknowledgment

This publication results from the research of the Center of Excellence CoSA at the University of Applied Sciences Lübeck. It is funded by the Federal Ministry of Transport and Digital Infrastructure of the Federal Republic of Germany (Id 165GU056F, Project BalticFuturePort) and by the Federal State Schleswig-Holstein, Germany (Id 22021005, Project KI-5G, Project Management Agency: WTSH).

References

[1] M. Kottkamp, A. Pandey, D. Raddino, A. Roessler, R. Stuhlfauth, "5G New Radio" 4th ed. Germany: Rohde&Schwarz, 2019.

[2] I. S. H. Martínez, I. P. O. J. Salcedo and I. B. S. R. Daza, "IoT application of WSN on 5G infrastructure," 2017 International Symposium on Networks, Computers and Communications (ISNCC), Marrakech, Morocco, 2017, pp. 1-6, doi: 10.1109/ISNCC.2017.8071989.

[3] Charles Rajesh Kumar J., Ahmed Almasarani, M.A. Majid, 5G-Wireless Sensor Networks for Smart Grid-Accelerating technology's progress and innovation in the Kingdom of Saudi Arabia, Procedia Computer Science, Volume 182, 2021, Pages 46-55, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2021.02.007.

[4] The MathWorks, Inc. "NR Cell Search and MIB and SIB1 Recovery Documentation" mathworks.com. Accessed: July 11, 2023. [Online]. Available: https://de.mathworks.com/help/5g/ug/nr-cell-search-and-mib-and-sib1-recovery.html.

[5] F. John, J. Schuljak, L. B. Vosteen, B. Sievers, A. Hanemann and H. Hellbrück, "A Reference Deployment of a Minimal Open-Source Private Industry and Campus 5G Standalone (SA) System," 2022 IEEE 10th International Conference on Information, Communication and Networks (ICICN), Zhangye, China, 2022, pp. 1-9, doi: 10.1109/ICICN56848.2022.10006563.

Towards Wireless Airflow Monitoring

Jan Schlichter
TU Braunschweig
Braunschweig, Germany
schlichter@ibr.cs.tu-bs.de

Sven Pullwitt
TU Braunschweig
Braunschweig, Germany
pullwitt@ibr.cs.tu-bs.de

Lars Wolf
TU Braunschweig
Braunschweig, Germany
wolf@ibr.cs.tu-bs.de

***Abstract*—Monitoring production parameters in the industry requires a wide range of sensors at different positions inside factories. The usage of wireless sensors offers great flexibility and lower deployment cost compared to wired sensors for this use case.**

In this paper, we show the adaption of a 3D airflow sensor from a wired to a wireless design and discuss different aspects, such as lifetime and deployment limitations. The design choices for the hardware and protocols used for the wireless version are shown with open research questions and reliability in mind.

***Index Terms*—3D Airflow Monitoring, Wireless Sensor Network, WSN**

I. Introduction

With the continuous expansion of Industry 4.0 [1], more and more processes driven by sensor data are implemented in factories. This results in a growing number of sensors to monitor as many relevant parameters as possible with fundamental decisions for each deployment. One is the way of transmitting the sensor data to a sink for further processing. In general, the data can be transmitted wired or wirelessly with many different techniques and protocols for both types.

In scenarios that do not allow wired solutions, e.g., due to the nature of the deployment locations, adapting wired solutions might be necessary. This paper shows the concept of adapting our wired 3D airflow sensor for industrial applications described in [2] to transmit its data wirelessly. We highlight the challenges during the development as well as the benefits and drawbacks of the solution. Furthermore, we show our selection process behind picking the technologies and platforms from an engineering and research perspective.

In the following, we will give a brief overview of the design of the wired sensor in Section II. The changes and our process for the new design are described in Section III. After highlighting some related articles in Section IV, the paper concludes with a conclusion and discussion in Section V.

II. Wired Design

The wired sensor is the result of our work presented in detail in [2]. It has a spherical design with a diameter of $115\,\mathrm{mm}$ and 6 evenly distributed funnels with a diameter of $80\,\mathrm{mm}$ on the surface. The funnels are connected with tubes to a differential pressure sensor from side to side to measure the differential pressure between the opposite sides. These measurements for all three axes are used to calculate the direction and speed of the airflow around the sphere.

Fig. 1b shows the sensor sphere with two funnels removed. Inside the sphere, three differential pressure sensors (Sensirion SDP31) are fixed to the tubes and connected with a wire to the communication board in the upper right. The sensors communicate through an I^2C-Bus. To enable a sufficiently long cable length between the sphere and sensor node, the communication board transforms the I^2C signal to a differential signal with a PCA9165 chip. Therefore, the cable length can be up to $25\,\mathrm{m}$ between the sphere and sensor node. The same chip is used on the sensor node side to convert the signal for communication with the MCU. Both sides use an RJ45 connector for cost-efficient deployment to connect the cable.

III. Wireless Design

The wireless design is shown in Fig. 1b. To keep all measurement properties of the sphere, we did not change the tubes, funnels or the form factor of the housing. The same differential pressure sensors as in the old design are connected to the tubes. Therefore, the collection of sensor values up to the I^2C-Bus connection inside the sphere is the same as before.

The connector board is switched out to enable the wireless operation. The new design uses a FireBeetle ESP32-E board as a control board. All sensor boards are connected to the same I^2C-Bus of the ESP32. The mounting plate inside the sphere was adjusted for the new board so that the USB-C connector of the ESP32 board is reachable from outside the sphere through a small cutout. Compared to the cutout of the RJ45 connector used previously, it uses less space.

The ESP32 board can be used for WiFi or Bluetooth communication to transmit the collected values to a server. A small $1100\,\mathrm{mA\,h}$ battery is used as a power supply. It is fixed with two zip ties to a new mount inside the sphere enclosure. Part of the battery is visible in the bottom left in Fig. 1b.

According to the datasheets, with the basic sleep functionality of the ESP32, the average current consumption will be around $4\,\mathrm{mA}$ for one measurement every minute. With the battery used here, this would lead to a lifetime of just 11 days. However, this value can be increased greatly by using larger batteries or improving the spheres' duty cycle and sleep functionality.

IV. Related Work

More and more wireless sensors are available in many different areas. For home automation, for example, Zigbee [3]

This research was funded by the German Federal Ministry for Economic Affairs and Climate Action (BMWK) by means of the 7th Energy Research Program of the German Federal Government under grant number 03EN1071A.

(a) Wired 3D Airflow Sensor (b) Wireless 3D Airflow Sensor

Fig. 1: Different Versions of the 3D Airflow Sensor

sensors and actuators are broadly available and build an IEEE 802.15.4-based mesh network for communication.

In industry, the availability seems to be more limited. There are, for example, sensors like the ALTA wireless air velocity sensor [4], which uses a proprietary protocol and multiple frequency bands to transmit its data. However, many other sensors are still based on Modbus or other wired protocols.

V. Conclusion and Discussion

As discussed above, due to the modular design of the sphere, converting it mechanically and electrically from wired to wireless communication is rather straightforward. With this conversion, the solution can be used in applications prohibiting wires connecting the spheres to a central sink. While the conversion of the hardware can be accomplished by swapping a few components with off-the-shelf modules, the resulting performance remains to be evaluated.

First, when moving an application from wired to wireless communication, special care must be taken to ensure the system's reliability. In wired industrial applications, the cost of deploying the infrastructure often exceeds the cost of the material, e.g., Ethernet cables. Therefore, over-provisioning the links w.r.t throughput and reliability is a viable solution. The shielded gigabit Ethernet links provide a high data rate and shielding against interference, so transferring a few bytes of sensor data can be achieved reliably. On the other hand, wireless links cannot be over-provisioned easily, as a broadband dedicated spectrum would be orders of magnitudes too expensive. As a result, many approaches can be found in the literature to provide reliable communication in interference-rich environments on shared frequency bands.

Another important factor for the wireless solution is the energy consumption and the resulting lifetime of the network. This is of secondary concern for most wired sensor applications since the power can be supplied over the same cable as the data. When using batteries, however, this needs to be taken into account. Again, as with reliability, a wealth of literature is available to achieve virtually infinite runtime in wireless sensor networks.

In addition, many approaches can be found for combining these partly conflicting two challenges, each with its own solutions. As a result, many scenarios can be covered with existing solutions from the literature.

For the first version of our wireless design, we opted to utilize off-the-shelf components with modules supporting only a few of the protocols in the literature. In fact, many protocols developed for specific challenges are rarely used outside of the research context in which they were developed. This can lead to projects not using the most capable solution researchers provide, even if they are very fitting for the application.

Implementing cutting-edge approaches from the literature often comes with significant implementation overhead. So, less fitting approaches with highly available modules are often used instead, even if it comes at the cost of reduced reliability of networks lifetime. To tackle this challenge, we propose three main design guidelines: First, novel protocol implementations should be made publicly available and documented accordingly. Evaluations of newly developed approaches should clearly state their strength and limitations to aid the selection in further applications. Implementations of research works should use standard hardware wherever possible to make the resulting protocol available to a larger group of developers.

References

[1] K. Schwab, *The fourth industrial revolution*. Currency, 2017.
[2] J. Schlichter, R. Ficht, and L. Wolf, "Under Pressure: 3-D Indoor Airflow Monitoring Based on Differential Pressure Measurements," *IEEE Sensors Journal*, vol. 23, no. 14, pp. 16 080–16 091, 2023.
[3] Zigbee Alliance, "Zigbee Specification," https://csa-iot.org/wp-content/uploads/2022/01/docs-05-3474-22-0csg-zigbee-specification-1.pdf, accessed: 2023-07-04.
[4] Monnit Corporation, "ALTA Wireless Air Velocity Sensors," https://monnit.blob.core.windows.net/site/documents/sensors/air-velocity/PS-AV-ADS-01.pdf, accessed: 2023-07-04.

Addressing the Complexity of Developing AI-based Applications for Low-power Sensor Nodes

Krzysztof Turchan, Krzysztof Piotrowski
IHP - Leibniz Institut für innovative Mikroelektronik
Frankfurt (Oder), Germany
{turchan, piotrowski}@ihp-microelectronics.com

***Abstract*—This document explores the integration of artificial intelligence (AI) with wireless sensor networks (WSNs). It addresses the challenges of finding suitable AI libraries that retain important features of WSN nodes. The focus is on low-power AI algorithms such as decision trees, Naive Bayes, and Support Vector Machines (SVM). Techniques like reduced models, dimensionality reduction, and distributed processing are proposed to overcome the limited computational power and resources of WSN nodes. Applications of AI in WSNs include edge computing, energy optimization, routing, network management, and self-adaptation. A framework is presented to support AI-based applications on WSN nodes, facilitating the development of AI-driven wireless sensor networks.**

***Index Terms*—wireless sensor network nodes, artificial intelligence algorithms, framework, microcontrollers**

I. Introduction

Wireless sensor networks (WSN) are a rapidly growing branch of distributed measurement systems. At a time when interest in artificial intelligence (AI) is growing at a very high rate, it is impossible not to use it in conjunction with WSNs. Problems arise when finding suitable libraries that provide the features of artificial intelligence while retaining the most important features of WSN nodes. While there are already tools that help in the implementation process of AI applications for embedded devices such as X-Cube-AI for STM32 microcontrollers [1], libraries that combine AI and wireless sensor networks are not yet a very popular topic.

This document is an attempt to collect the most important currently known artificial intelligence tools and algorithms, which will next be used as part of a framework to support the implementation of AI-Based applications for low-power WSN nodes. These tools will be described and their most important features will be extracted. Moreover, there will be a description of the concept of the mentioned Framework.

II. How to Combine AI and WSN Nodes

Thus, how to combine the features of artificial intelligence with the relatively low computational power of a node? One way is to use reduced models, algorithms with low computational complexity [2].

Another is to reduce the dimensionality of the data. Using dimensionality reduction techniques, such as principal component analysis (PCA) or feature selection methods, can help reduce the size of the data and consequently reduce the computational load.

Distributed processing is also a good practice. Instead of relying on a single base node to process data and make decisions, a distributed processing approach can be used. Nodes can collaborate, exchange information and make decisions based on local data and algorithms. In this way, the computational load can be distributed and energy consumption can be reduced at individual nodes.

As is well known, nodes in WSNs have limited resources, such as computing power, memory and energy. Therefore, when designing and implementing artificial intelligence algorithms, the main focus should be on optimizing the consumption of these resources. Energy optimization techniques, computational complexity reduction, data compression, etc., can be used to ensure efficient use of node resources [3].

III. Where to Use AI on WSN Nodes

At the outset, it is worth considering for what purpose artificial intelligence is used in wireless sensor networks. The first thing that comes to mind is so-called Edge computing. That is, analyzing and interpreting data collected by sensors to extract relevant information without sending it to the base node. Machine learning algorithms can be used for pattern recognition, data classification, anomaly detection or trend prediction. Such solutions bring nodes to a completely different level, while it is important to balance the advantages and disadvantages of such a solution. E.g. non-linear data processing can increase the consumption of computing power [3].

Referring to the mentioned energy efficiency, AI can also be used to optimize energy consumption in wireless sensor networks. Machine learning algorithms can analyze patterns and sensor data to optimize network scheduling and energy management. For example, machine learning algorithms can predict the load on sensor nodes and adjust operating schedules to, for example, store more electricity and give it away when demand is highest.

A third possible application of AI algorithms is routing and intelligent network management. Machine learning algorithms can analyze network conditions, predict node availability and determine optimal communication routes. AI can also help manage and configure networks to adjust network parameters in real time to optimize network performance and availability.

As a final example, AI can help sensor networks self-adapt to changing conditions and environments. Machine learning

algorithms can analyze sensor data to identify changes and automatically respond to them. For example, AI can adjust network parameters, such as sampling times, to optimize network performance or frequency bandwidth usage time (duty cycling).

IV. Low Power AI Algorithms Suitable for WSN Nodes

In the context of WSNs, it is crucial to carefully select low-power AI algorithms that align with the resource constraints and energy limitations of WSN nodes. Here are several exemplary low-power AI algorithms that are well-suited for WSN nodes:

A. Decision Trees

Decision trees are lightweight machine learning algorithms that offer efficient classification capabilities, making them ideal for WSN nodes. Due to their low computational requirements, decision trees can effectively classify sensor data or make decisions based on specific conditions, all while conserving energy [4]. Here is an example of the use of decision trees in WSNs:

WSN nodes can use decision trees to classify sensor data. For example, in an environmental monitoring system, a decision tree can analyze sensor readings, such as temperature, humidity, or pollution levels, and classify them as ”normal” or ”emergency.” This makes it possible to quickly identify potential hazards and take appropriate action.

B. Naive Bayes Classifier

The Naive Bayes algorithm is known for its simplicity and efficiency, making it a favorable choice for low-power WSN nodes. It operates on probabilistic principles, enabling it to predict and classify sensor data based on the probability of different events occurring. Naive Bayes classifiers require minimal computational resources, contributing to their suitability for WSN applications.

The Naive Bayes algorithm can be used in WSN nodes to filter unwanted packets in wireless communications. Based on the content of the messages, the Naive Bayes Classifier can assign them a probability of being, for example, an intended attack on the network. This allows nodes to make decisions on whether to discard or continue forwarding a packet based on this probability.

C. Support Vector Machines

SVM is a well-established machine learning algorithm that can handle complex decision boundaries, offering high classification accuracy within WSN nodes. However, due to its slightly higher computational demands compared to other algorithms mentioned, SVM should be carefully deployed in resource-constrained WSN environments where more intricate decision boundaries are necessary.

WSN nodes can collect energy consumption data over a specified period of time. Using an SVM algorithm, nodes can analyze this time-course data and forecast future energy consumption patterns [5]. Based on these forecasts, actions can be taken to optimally manage energy in the WSN, such as adjusting transmission schedules or regulating node operation.

V. Framework Supporting Implementation of AI Applications for Low-Power WSN Nodes

Just having a list of algorithms does not make it easier to implement applications for WSN nodes. This requires a tool that helps in the process of adapting these algorithms and generating code. For this purpose, a framework will be implemented, which tasks are selecting the appropriate artificial intelligence algorithm according to the needs, defining and learning of the model, and generating the code. Initially, the Framework will have the ability to create artificial intelligence-based applications for individual WSN nodes. The main task of this part will be to create a project with generated code for the selected node, along with all necessary libraries and artificial intelligence model. Obviously using the most energy-efficient AI algorithms. This will provide the functionality described in the first paragraph of Chapter III. In the future, it is planned to implement functionalities that will allow intelligent management of the energy demand of the entire WSN and the ability of nodes to adapt depending on the conditions in which they operate. Such a framework will allow easier development of wireless sensor networks based on artificial intelligence, consisting of nodes that allow non-linear data processing.

Acknowledgment

This work was supported by the European Regional Development Fund within the BB-PL INTERREG V A 2014-2020 Programme, “reducing barriers - using the common strengths”, project SpaceRegion, grant number 85038043. The funding institutions had no role in the design of the study, the collection, analyses, or interpretation of data, in the writing of the manuscript, or in the decision to publish the results.

References

[1] “X-CUBE-AI,” STMicroelectronics, https://www.st.com/en/embedded-software/x-cube-ai.html (accessed Jul. 19, 2023).

[2] Z. Długosz, M. Rajewski, R. Długosz, and T. Talaśka, “A novel, low computational complexity, parallel swarm algorithm for application in low-energy devices,” MDPI, (accessed Jul. 19, 2023).

[3] K. Zaimen, M.-E.-A. Brahmia, L. Moalic, A. Abouaissa, and L. Idoumghar, ‘A Survey of Artificial Intelligence Based WSNs Deployment Techniques and Related Objectives Modeling’, IEEE Access, vol. 10, pp. 113294–113329, 2022.

[4] B. Sugiarto and R. Sustika, ‘Data classification for air quality on wireless sensor network monitoring system using decision tree algorithm’, in 2016 2nd International Conference on Science and Technology-Computer (ICST), 2016, pp. 172–176.

[5] S. Preda, S.-V. Oprea, A. Bâra, and A. Belciu (Velicanu), ‘PV Forecasting Using Support Vector Machine Learning in a Big Data Analytics Context’, Symmetry, vol. 10, no. 12, 2018.

Sensor Fault Diagnosis for Precision Agriculture

Florian Mikolajczak, Bettina Schnor
Institute of Computer Science
University of Potsdam
Potsdam, Germany
{fmikolaj,schnor}@uni-potsdam.de

Geraldín Montañez Huamán
Faculty of Electronic and Electrical Engineering
National University of San Marcos
Lima, Peru
geraldin.montanez@unmsm.edu.pe

***Index Terms*—Sensor fault detection, IoT, fuzzy logic**

I. Introduction

Sensors are an integral part of IoT systems. They collect data about the surrounding environment. These data can be used for monitoring the environment, but also to infer human or actuator-based actions. Any inference is therefore not only dependent on the availability of data, but also on the data quality. A faulty sensor, i.e. one reporting incorrect data could have detrimental effects on an IoT system [1]. For example, in precision agriculture, IoT systems are deployed within a crop field or greenhouse. They include sensors to monitor environmental data like soil moisture or temperature and water pumps for irrigation. The behavior of the water pumps can be automated based on the sensor readings e.g. triggering irrigation if the soil is dry. However, faulty high soil moisture values may lead to wilting crops. Hence, to ensure a well-operating IoT system, faulty sensors have to be detected and replaced. The talk will give an overview over sensor fault diagnosis and presents our experiences in our Precision Agriculture testbed [2].

Classification of Faults

Different categorizations of sensor faults exist (see for example [1], [3]). Li et al. divide sensor faults into *incipient* and *abrupt* failures [1]. Abrupt failures result in complete failure of the sensor such that no data is collected or sent. The reason may be a broken sensor or sensor board, a broken network connection or a discharged battery. These types of faults can be easily detected by a monitoring system.

In contrast, incipient failures are caused by an abnormal sensor status in which incorrect data is sent. Incipient failures can be further distinguished and include similar fault classes like proposed by Zou et al. as so-called soft failures such as drift, bias, stuck fault, accuracy decline, and spike fault [3].

Classification of Sensor Faults Detection

A simple approach for identifying faulty sensors is redundancy: using multiple sensors of the same type. Then, a faulty sensor can be identified by its measurements differing from the majority. However, this approach increases costs, maintenance and system complexity. In a different approach, detection of a faulty sensor is based on historic data of the sensor or knowledge about the sensor's behavior [1], [3].

Research for detecting faulty sensors is ongoing and there exists no agreed classification yet. Here, we follow Li et al. [1] and distinguish model-, knowledge- and data-based approaches. A **model-based** approach consists of a mathematical model, which describes the system behavior. The values obtained by the sensor are compared to those predicted by the model. However, developing a model that can accurately describe the behavior of sensors is complicated in practice, especially if different types of sensors are used. This approach is suited for extreme challenging projects for example within spacecraft control systems [4].

Knowledge-based approaches are characterized by an expert system. The expert system consists of a knowledge and a rule base and a reasoning mechanism. The expert's knowledge about the system forms the knowledge base. Based on the reasoning mechanism, knowledge-based systems are divided into rule-based or fuzzy inference systems. Rule-based systems require specific binary (true/false) rules that can be hard to obtain for complex systems. In a fuzzy inference system, the *weak* knowledge about the modeled system is formulated using fuzzy logic. A rule-based system has been applied in a setting with greenhouse environmental sensors [5], and also fuzzy inference systems have been used in a variety of settings [6]–[8].

The **data-based** approach is the most current of the three. From large amounts of labeled data, a classifier is obtained by training. Possible approaches include neural networks of varying complexity, and also support vector machines [9], [10]. However, large data requirements and necessary (re)-training in data-based methods is time- and resource consuming.

II. Precision Agriculture Testbed

To evaluate and demonstrate the benefits of the semantic-based approach of the MYNO project [11], a precision agriculture testbed was set up at the University of Potsdam [2]. The testbed contains a Raspberry Pi 3B as an edge component, and several microcontroller boards which monitor and water a group of plants. Communication is done over WiFi. The Raspberry Pi runs the MYNO components: an MQTT-broker and the NETCONF-MQTT bridge as well as a NETCONF-client, which provides a user-interface. The user-interface displays current sensor values and allows user input to control actuators and set up automations. The microcontroller boards are based on the low-priced ESP32 NodeMCU Module.

Multiple sensors are used per board since each board monitors a number of plants. Additionally, multiple sensors of the same type are used in proximity to compare their readings to confirm our fault detection diagnosis. The following sensors are used per board:

- 3 different soil moisture sensors (discussed below)
- a temperature, humidity and air pressure sensor combined in a GY-BME280 module
- a GY-302 BH1750 light sensor
- a touch-free capacitance sensor to detect the water level within the water reservoir (one board only)

The following actuators are deployed:

- a 9V mini water pump (one board only)
- a RGB LED module as a simple actuator to test board responsiveness

Observed Sensor Faults

Running our testbed since summer 2021, we did not observe most of the aforementioned sensor fault classes known from literature. In our case, we noticed most faults in soil moisture sensors. Abrupt faults were limited to a single light sensor and full sensor boards. All other faults were incipient failures of soil moisture sensors. We observed soil moisture sensors reporting incorrect, almost constant values. Motivated by this, we attached a database to collect all sensor values from March to May 2023. During this period we noticed the following faults in our system:

- Sensor board fault, two times: hard fault, unresponsive sensor board
- Soil moisture sensor fault, 10+ times:
  - intermittent largely differences between measurements (spike fault in [3])
  - permanently constant measurement values (stuck fault in [3], possibly with prior drift)

From the incipient faults mentioned in literature we did not observe accuracy decline or bias. The soil moisture sensors faults are caused by sensor deterioration over time. In the most severe cases, electronic parts were eventually exposed to the environment, i.e. water, leading to corrosion. It is assumed that this is due to the materials used and the manufacturing process.

We employed three different sensors that use two different ways of measuring soil moisture. A resistive soil moisture sensor by AZ-Delivery (≈ 3€) and capacitive soil moisture sensors by AZ-Delivery (≈ 2€) and BeFlE (≈ 13€).

In resistive soil moisture sensors, the electrodes are exposed to the environment by design and their sensitivity to electrolytic corrosion has been identified [12], [13]. Corrosion and erroneous measurement values occurred within one week of deployment. A corroded sensor reported constant values of 0% soil moisture. Capacitive soil moisture sensors try to circumvent this problem by employing a different technique. They determine the dielectric constant of the soil, which changes depending on the water content [14]. Capacitive soil moisture sensors do not expose electronic parts to the environment directly. However, we noticed corrosion in many of the capacitive sensors by AZ-Delivery as well, albeit after weeks or months. The protective layer of the sensor was bloated and partly broken off. We assume that this is due to water-intake as the protective layers are only pressed and glued together, leaving an open edge. Corroded sensors of this type report a constant 0% or high values of soil moisture. The BeFlE sensor included a "durable protective layer" [15] similar to epoxy resin in appearance. However, this protective layer developed small blisters over time. If blisters are present, the sensor reports low-varying high values of soil moisture.

Fuzzy-Logic based Detection of Faulty Soil Moisture Sensors

While abrupt failures are easily detected by MYNO's monitoring component, incipient failures remained a challenge. Since it is hard to find a mathematical model and on the other hand a data-based approach seems not to be sustainable (regarding different sensor types), we developed a Fuzzy-Logic based Sensor Fault Detection. The Fuzzy-Logic approach is in a sense a mixture of the model- and data-based approach. It does require some *fuzzy* knowledge about the system at hand. Therefore, we inpected collected previous data and formulated a set of *fuzzy rules*:

- If the pump has been triggered, the moisture must rise to at least 89%
- Over three days, the soil moisture must decrease by a value of $m \in \mathcal{N}$, where $x \leq m \leq y$ percent,, where x and y are positive thresholds. This holds unless the pump is activated. The thresholds x and y depend on the manufacturer's specifications for sensor variability and the concrete environment itself. For our testbed, we chose $8 \leq m \leq 15$

The inputs are the water pump state and the difference between the current and old soil moisture values, whereas the output is the diagnosis that indicates fault and non-fault state.

The talk will present the fuzzy-logic system in detail and also our experiences during operation.

References

[1] D. Li, Y. Wang, J. Wang, C. Wang, and Y. Duan, "Recent advances in sensor fault diagnosis: A review," *Sensors and Actuators A: Physical*, vol. 309, p. 111990, 2020.

[2] K. Sahlmann, F. Mikolajczak, and B. Schnor, "Interoperability in the iot – an evaluation of the semantic-based approach," in *19th GI/ITG KuVS Fachgespräch: Drahtlose Sensornetze*, Berlin, Germany, 2022.

[3] X. Zou, W. Liu, Z. Huo, S. Wang, Z. Chen, C. Xin, Y. Bai, Z. Liang, Y. Gong, Y. Qian *et al.*, "Current status and prospects of Research on sensor fault diagnosis of agricultural internet of things," *Sensors*, vol. 23, no. 5, p. 2528, 2023.

[4] L. Yuqing, Y. Tianshe, L. Jian, F. Na, and W. Guan, "A fault diagnosis method by multi sensor fusion for spacecraft control system sensors," in *2016 IEEE International Conference on Mechatronics and Automation*, 2016, pp. 748–753.

[5] S. A. Beaulah, Z. S. Chalabi, and D. G. Randle, "A real-time knowledge-based system for intelligent monitoring in complex, sensor-rich environments," *Computers and electronics in agriculture*, vol. 21, no. 1, pp. 53–68, 1998.

[6] M. Geetha and J. Jerome, "Fuzzy expert system based sensor and actuator fault diagnosis for continuous stirred tank reactor," in *2013 International Conference on Fuzzy Theory and Its Applications (iFUZZY)*. IEEE, 2013, pp. 251–257.

[7] Y. L. Ou, "Fault diagnosis with fuzzy expert system," *Applied Mechanics and Materials*, vol. 48, pp. 519–522, 2011.

[8] R. Shahnazi and Q. Zhao, "Adaptive Fuzzy Descriptor Sliding Mode Observer-based Sensor Fault Estimation for Uncertain Nonlinear Systems," *Asian Journal of Control*, vol. 218, no. 4, pp. 1478–1488, 2016.

[9] T. Luo and S. G. Nagarajan, "Distributed anomaly detection using autoencoder neural networks in WSN for IoT," in *2018 ieee international conference on communications (icc)*. IEEE, 2018, pp. 1–6.

[10] J. Loy-Benitez, Q. Li, K. Nam, and C. Yoo, "Sustainable subway indoor air quality monitoring and fault-tolerant ventilation control using a sparse autoencoder-driven sensor self-validation," *Sustainable Cities and Society*, vol. 52, p. 101847, 2020.

[11] K. Sahlmann, T. Scheffler, and B. Schnor, "Ontology-driven device descriptions for iot network management," in *2018 Global Internet of Things Summit (GIoTS)*, 2018, pp. 1–6.

[12] Y. J. Jeong, K. E. An, S. W. Lee, and D. Seo, "Improved durability of soil humidity sensor for agricultural IoT environments," in *2018 IEEE international conference on consumer electronics (ICCE)*. IEEE, 2018, pp. 1–2.

[13] M. Saleh, I. H. Elhajj, D. Asmar, I. Bashour, and S. Kidess, "Experimental evaluation of low-cost resistive soil moisture sensors," in *2016 IEEE International Multidisciplinary Conference on Engineering Technology (IMCET)*. IEEE, 2016, pp. 179–184.

[14] P. Placidi, L. Gasperini, A. Grassi, M. Cecconi, and A. Scorzoni, "Characterization of low-cost capacitive soil moisture sensors for iot networks," *Sensors*, vol. 20, no. 12, p. 3585, 2020.

[15] BeFle, "BeFlE SoMoSe v2.2 Documentation," 2022. [Online]. Available: https://github.com/BeFlE/SoMoSe

Towards the Optimal Sensors for WSN Applications: Effective Rainfall Monitoring

Przemysław Zielony, Krzysztof Piotrowski
IHP - *Leibniz Institut für innovative Mikroelektronik*
Frankfurt (Oder), Germany
{zielony, piotrowski}@ihp-microelectronics.com

***Abstract*— Embedded systems are used to monitor many things. One of the key factors in the quality of life and operation of the systems is the ability to effectively monitor rainfall. This article compares three rain sensors with different measurement methods: a piezoelectric sensor, a radar sensor relying on the Doppler effect, and a tipping bucket sensor. The sensors were tested in a real environment from different aspects. Based on the collected data, recommendations for selecting a rain sensor in embedded systems are presented.**

Keywords— embedded systems, rain sensors, comparison, rainfall monitoring, piezoelectric measurement, Doppler effect measurement, tipping bucket

I. INTRODUCTION

One of the key factors that has a direct impact on many aspects of our lives is rainfall. Effective real-time rain detection and monitoring is important in many areas, such as: agriculture, aviation, road infrastructure, hydrological monitoring or urban infrastructure [1].

The height of rainfall is measured in millimeters. It is the height of the layer of water that would form on the surface of the earth with an area of 1 square meter if the water from the rainfall did not run off, soak in, and evaporate. Rainfall intensity is the amount of rainfall (in millimeters) per unit of time (1 hour).

Professional rainfall measurements are made using a Hellman rain gauge also called a pluviometer. It is a container with sharp edges and a certain area of the top opening (usually 200cm^2). The instrument is mounted at a height of 1 meter so that water bouncing off the ground does not affect the accuracy of the measurement. The container in which the water has collected has a scale from which the volume of water can be read.

Measuring rainfall with a Hellman rain gauge is done manually, i.e., a human being is needed to read the water level in the instrument. This process can be automated through the use of embedded systems.

In many cases, when there is a need to automate measurements, particularly in areas with limited manual access and no continuous power supply, the selection of appropriate sensors becomes crucial. Therefore, it is necessary to consider several aspects, including energy efficiency, maintenance-free operation, and accuracy. The choice of a rain sensor is presented, taking into account the aforementioned factors.

II. USED TYPES OF SENSORS

A. Piezoelectric rainfall sensor

Piezoelectric precipitation sensors measure precipitation based on the piezoelectric effect [2]. It involves the generating of an electric charge, in a material under the influence of an external mechanical stress. The sensors use specific piezoelectric materials such as, quartz crystals, barium titanate or piezoelectric polymer. Rain, snow or hailfall exerts pressure on the surface of the sensor, causing deformation of its structure. This deformation generates an electric charge, which is processed by the sensor and determines the intensity of the precipitation.

Advantages:

- Resistance to weather conditions
- Maintenance-free operation
- No mechanical parts

Disadvantages:

- Susceptibility to interference generated by mechanical vibration, noise
- Power consumption
- High price

B. Doppler radar for rainfall detection

Precipitation sensors based on the principle of the Doppler effect make measurements based on the movement of particles and thus the change in frequency of the electromagnetic wave between the receiver and the source. In the case of a precipitation sensor, these are raindrops, snowflakes or hail. The sensor sends a microwave in the direction of the precipitation and the precipitation reflects some of the wave energy toward the sensor. Based on this effect, the speed and intensity of precipitation can be determined.

Advantages:

- Radar allows more accurate measurement compared to other methods
- Ability to detect different types of precipitation: rain, snow, hail
- No mechanical components

Disadvantages:

- Sensitivity to interference, e.g. strong winds
- High price
- Problem with detecting very small precipitation particles

C. Tipping bucket rainfall sensor

The tipping bucket is a type of rain gauge commonly used to measure rainfall in many projects, including commercial [3]. The rain is collected in the small funnel-shaped bucket, which must be accurately sized, as this affects the measurement result. Typical bucket capacities in sensors range from 0.1mm to 2mm, which means that for each bucket fill, the sensor counts rainfall with a value corresponding to the bucket capacity. When the bucket fills to a sufficient capacity, the water presses down on the cradle, which tilts. Each tilt of the cradle sends a pulse/interrupt to the microcontroller, which counts the pulses. Through this, it is possible to determine what the rainfall is during the measured period.

Advantages:

- High-end sensors offer good measurement precision
- They are used in many projects
- Low cost of the sensor

Disadvantages:

- If the sensor is located around trees, the funnel may become dirty or clogged making accurate measurement impossible
- Snow can completely clog the funnel, making accurate measurement impossible (some sensors have heating which eliminates this drawback)
- That types of sensors have moving parts, mechanical problems can very much affect accuracy

III. Tests

The following sensors were used for testing:

- HongYuv RS2E 24GHz doppler radar rain gauge
- HongYuv RS3E piezoelectric rainfall sensor
- DFRobot Gravity: Tipping Bucket Rainfall Sensor

The measurements were collected on 16/07/2023 from 5:50AM to 7:50AM and are presented in Figure 1.

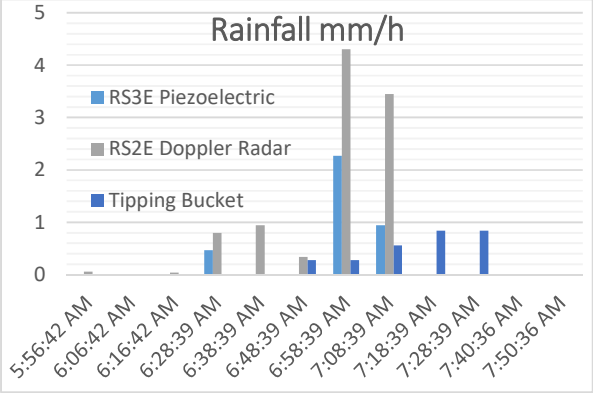


Figure 1. Rainfall Measurements Comparison Chart

The sensors described above were mounted on the roof side by side in such a way as not to interfere with each other. A CC1352R1 microcontroller from Texas Instruments was used to collect data from the sensors.

Sensor readings were taken every 10 minutes. The website worldweatheronline.com reports that the average rainfall in the measured area from 4:30AM to 7:30AM was 0.1mm. Average rainfall for the period is included in TABLE I.

TABLE I. AVERAGE RAINFALL COLLECTED FROM SENSORS COMPARED TO WORLDWEATHERONLINE.COM

Average precipitation from sensors compared to worldweatheronline.com	
RS3E Piezoelectric	0.2038mm
RS2E Doppler Radar	0.5517mm
Tipping Bucket	0.2483mm
Worldweatheronline.com	0.1mm

IV. Conclusion and Futher Steps

Real environmental measurements are characterized by the fact that data is collected only when it rains, and during the measured time the precipitation was not too heavy. Tests show that the sensors have trouble measuring small values.

Each of the sensors showed some deviation from the actual average rainfall reported by worldweatheronline.com. It must be assumed that the data provided by this service may apply to a much larger area such as an entire city.

The Tipping Bucket sensor has the lowest resolution, which means it may not be able to accurately measure small amounts of rainfall, such as average rainfall of 0.1 mm/m^2. It also has low resistance to mechanical damage and requires frequent maintenance. However, it takes incomparably less current compared to other sensors.

In the case of the piezoelectric sensor, it gave the closest measurements to the real thing, requires no maintenance and its power consumption does not disqualify it for use in WSN systems.

The sensor equipped with Doppler radar showed higher values than the other sensors, draws much more current, but has the ability to determine the type of precipitation measured.

Tests show that there is no single good rain sensor that will work in all weather conditions and be suitable for any system.

Each sensor has its own characteristics and applications, which can affect the final choice depending on specific needs and preferences shown in TABLE II.

TABLE II. RAINFALL SENSORS COMPARISON

Rainfall Sensors Comparison			
	Tipping Bucket	**Doppler Radar**	**Piezoelectric**
Resolution	0.28mm	0,01mm	0,01 mm
Resistance to mechanical failures	LOW	HIGH	HIGH
Maintenance	HIGH	LOW	LOW
Energy consuming	<1mA@3.3V	130mA@ 12V	15mA@ 12V
Price	LOW	HIGH	HIGH
Determining the type of precipitation	NO	YES	NO
Calibration Required	NO	NO	NO

A similar problem of choosing the right sensor also applies to many other quantities that are measured in WSNs.

Further tests would have to be performed on a larger sample of measurements.

Acknowledgment

This work was supported by the European Regional Development Fund within the BB-PL INTERREG V A 2014-2020 Programme, “reducing barriers – using the common strengths”, project SmartRiver, grant number 85029892. The funding institutions had no role in the design of the study, the collection, analyses, or interpretation of data, the writing of the manuscript, or the decision to publish the results.

References

[1] F. Testik, G. Mekonnen, “Rainfall: State of the Science,” American Geophysical Union, January 2010.

[2] W. Heywang, K. Lubitz, W. Wersing, “Piezoelectricity Evolution and Future of a Technology,” Springer, November 2010

[3] D. A. Segovia-Cardozo, C. Bernal-Basurco, L. Rodriguez-Sinobas, “Tipping Bucket Rain Gauges in Hydrological Research: Summary on Measurement Uncertainties, Calibration, and Error Reduction Strategies,” MDPI, June 2023.

Automated Testing of Hardware Abstraction Layers on Microcontrollers

Marian Buschsieweke, Mesut Güneş
Communication and Networked Systems (ComSys)
Otto-von-Guericke University Magdeburg
Universitätsplatz 2, 39106 Magdeburg, Germany

***Abstract*—In this paper we introduce the RIOT Peripheral Selftest Shield, an extension board attached to MCU boards to aid low level testing. A comparison of different board extension formats motivates the design choice for an Arduino UNO compatible extension board. We detail how the extension board aids the test application in testing correctness of MCU peripheral drivers. Our evaluation shows that the extension board allows testing of all peripheral systems in regard to most modes and aspects.**

***Index Terms*—HiL Testing, Continous Integration, Internet of Things, Operating Systems, Open Source**

I. Introduction

RIOT [1,4] is an open source OS for Microcontroller Unit (MCU) powered hardware such as IoT devices developed by a distributed and diverse community. RIOT provides Hardware Abstraction Layers (HALs) to access peripherals of different MCUs of different vendors using the same API. Testing these drivers has proven to be challenging, as RIOT as of this writing supports 44 different MCU families that differ in one or more peripheral driver from every other supported MCU family. Hence, any rigorous attempt at testing for correct behavior of the peripheral drivers must scale.

Software directly interacting with hardware, such as peripheral drivers, is inherently fragile: Strict timing requirements between two memory mapped I/O accesses may no longer be met when the peripheral is combined with a faster CPU or the code is compiled by a more aggressively optimizing compiler. As a result, peripheral drivers require regular testing even when the code has not changed since the last test cycle.

No single contributor in the RIOT community has access to each and every board RIOT supports. Therefore, testing needs to be done in a distributed fashion when aiming for a wide test coverage. This means testing equipment should ideally be easily available and affordable, so that even hobbyist code contributors can also contribute to the testing effort.

II. Related Work

The Zephyr Project [5] employs a custom testing automation software, twister [7], that makes use of a fixtures to model dependencies on hardware for tests. For example the `gpio_loopback` test fixture depends on pairs GPIO pins being connected. This loop-back mode self testing is cost efficient and effective. However, manually preparing the hardware and providing the configuration describing the connections made requires some time to set up.

Weiss et al. proposed with PHiLIP on the HiL [6] the use of specialized testing hardware connected to the device under test (DUT). Unlike loop-back testing, even low level aspects such as SPI clock phase can also be validated. The downside is higher setup effort and costs. Given the use case of attaching PHiLIP permanently to the DUT and connecting it to a continuous integration (CI) testing pipeline, this one time setup cost will amortise over time, though.

Extension Standard	I^2C	SPI	UART	PWM	ADC	GPIO	Adoption
Arduino UNO	✓	✓	✓	✓	✓	✓	High
Arduino Mega	✓	✓	✓	✓	✓	✓	Medium
Arduino MKR	✓	✓	✓	✓	✓	✓	Low
Arduino Nano	✓	✓	✓	✓	✓	✓	Medium
Adafruit Feather	✓	✓	✓	✓	✓	✓	High
D1 Mini	✓	✓	✓	✗	✗	✓	Low
Microduino	✓	✓	✓	✓	✓	✓	Low
micro:bit	✓	✓	✓	✓	✓	✓	Low
Olimex UEXT	✓	✓	✓	✗	✗	✗	Low

Table I: Comparison of extension board interfaces regarding the provided interfaces and the availability of boards compatible with it.

Testbeds, as often used for research of wireless mesh networks [2], provide an established real hardware platform with many providing free access to third party researchers. In RIOS [1,4] the use of testbeds for automated tests prior releases [3] is well established. This is particularly useful for performing end to end networking tests. However, testing peripheral drivers especially with focus on low level aspects is not feasible with the hardware deployed in most testbeds.

III. Board Extension Formats

Table I compares popular extension board interfaces regarding the features and their adoption. Here, adoption refers to availability of different host boards designs compatible with the standard rather than the availability of extension boards. Only the Arduino UNO interface and the Adafruit Feather interface have a high adoption and provide all peripheral interfaces we plan to test. Hence, we conducted a thorough comparison of the two interface standards in Table II. As shown in Table II, for 25 of 44 supported MCU families an

MCU Family	Arduino UNO	ISP	Adafruit Feather
ATmega	Arduino UNO	✓	Feather 328P
EFM32/EFR32/EZR32	–	–	Thing Plus Matter
ESP32	–	–	HUZZAH32
ESP32S2	–	–	Feather S2
ESP32S3	–	–	Feather S3
ESP8266	–	–	HUZZAH
FE310	HiFive 1	✗	Thing Plus FE310
Kinetis	frdm-k64f	✗	–
nRF51	nRF51 DK	✓	–
nRF52	nRF52840 DK	✓	Feather nRF52840 Express
nRF9160	nRF9160 DK	✓	Thing Plus nRF9160
QN908x	QN9080-DK	✗	–
RP2040	ArduPico	✗	Feather RP2040
SAM3	Arduino Due	✓	–
SAMD21	Arduino Zero	✓	Feather M0 Express
SAMD5x	–	–	Feather M4 Express
SAML1x	–	–	Thing Plus SAMD51
STM32F4	Nucleo-F446RE	✗	Feather STM32F405
13 other STM32	Nucleo-64 or Nucleo-144	✗	–
Total	**25**	**6**	**15**

Table II: Availability of boards compatible with a given extension format by MCU family for each supported MCU family in RIOT.

Peripheral	Tested By	Shield Required?
GPIO	loop-back (two pins connected)	✓
	I^2C GPIO extender	✓
UART	loop-back (TXD to RXD)	✓
	timer to estimate symbol rate	✗
SPI	loop-back (serial out to serial in)	✓
	$\overline{\texttt{CS}}$ connected to GPIO pin	✓
	SCK connected to GPIO extender pin	✓
	timer to estimate clock frequency	✗
I^2C	I^2C GPIO extender	✓
	GPIO extender pin connected to GPIO pin	✓
ADC	connected to PWM DAC	✓
	connected to 4 bit R-2R DAC	✓
PWM	connected to ADC with low-pass filter	✓

Table III: Testing approaches used in our test application and which make use of the Peripheral Selftest Shield

Arduino UNO compatible board is available, while the Feather extension standard would cover 15 MCU families.

IV. Performing Tests Using the Peripheral Self-Testing Shield

Our Peripheral Selftest Shield follows the popular Arduino UNO Shield format and supports an SPI bus in both common configurations: on ISP header or on pins D11, D12, and D13. As shown in Table III most peripheral drivers are tested by feeding the output of the peripheral into the input of the same peripheral (loop-back test) or into a second peripheral. The test application contains a test suite for each peripheral which contains tests cases that generate a specific output and compare the looped back input with the expected values. The GPIO and ADC peripheral have redundant tests: Correct operation of GPIO is tested both via loop-back to other GPIO pins and to the I^2C attached GPIO extender. Similar, the ADC is tested both via the DAC using Pulse Width Modulation (PWM) and a low-pass filter and the R-2R resistor ladder DAC. This way a single failing test case can be easily tracked down to the peripheral driver that is misbehaving.

The board consists of passive components only with the exception of the I^2C attached GPIO extender. Apart from the 8 surface mount resistors in the 0805 package for the R-2R resistor ladder, only through-hole parts with a $2.54\,\mathrm{mm}$ pitch were used that are particularly easy to solder on by hand. The reason to for using surface mount resistors for the resistor ladder is that sourcing affordable through hole resistors with a high accuracy is difficult. However, resistors in the 0805 package are relatively large and still easy to solder on by hand. All components required to assembly a single PCB costs less than $\$\,10$ while PCBs at a quantity of five can be ordered for less than $\$\,1$ per PCB (excluding shipping).

V. Evaluation

The test application we implement is capable of testing a total of 6 peripheral drivers in regard to 19 of 24 modes or attributes by making use of the Peripheral Selftest Shield we developed, as shown in Table IV. The setup requires mating the shield with the DUT, selecting logic level, and flashing the test application. In total the setup is ready in typically less than 30 seconds. The low cost of the board makes it affordable to manufacture dozen of shields to permanently mate with boards that can be attached to a CI server for autonomous testing. Even distributing free Peripheral Selftest Shields among the community at social events such as the RIOT summits is feasible.

Peripheral	Mode / Aspect	Covered By Test?
GPIO	Floating Input	✓
	Push-Pull Output	✓
	Input with Pull-Up	✓
	Input with Pull-Down	✓
	Open-Drain	✓
	Open-Drain with Pull-Up	✓
	Interrupts	✓
UART	Data Integrity	✓
	Symbol Rate	(✓)
	Stop Bits	✓
	Parity Bit	(✓)
	Power Off Behavior	(✓)
SPI	Data Integrity	✓
	Bit Order	✗
	Clock Frequency	(✓)
	Clock Polarity	✓
	Clock Phase	✗
	$\overline{\texttt{CS}}$ Signaling	✓
I^2C	Data Integrity	✓
	Clock Frequency	✗
	Clock Stretching	✗
ADC	Accuracy	✓
PWM	Duty Cycle	✓
	PWM Frequency	✗
Summary	**19 of 24 covered**	

Table IV: Modes and aspects of peripheral operation covered by the testing application using our Peripheral Selftest Shield.

VI. Summary and Conclusions

This paper introduces an extension board that allows Hardware in the Loop (HiL) testing both in CI pipelines and manually by a distributed and heterogeneous developer community. We have shown that this board can automate testing of MCU peripheral drivers with minimal setup effort. Compared to existing approaches such as PHiLIP on the HiL [6] the effort to setup up the hardware, to configure the hardware, and to write tests is significantly reduced. The downside is that it is impossible to detect certain classes of bugs with our approach that PHiLIP would detect, such as incorrect SPI bit order or clock polarity. For this reason our board cannot replace PHiLIP in RIOTs testing landscape, but rather complement it. The fact that we were able to detect a number of previously unknown bugs across different MCU families shows that our approach indeed provides value to the testing landscape of RIOT.

References

[1] Emmanuel Baccelli, Oliver Hahm, Matthias Wählisch, Mesut Günes, and Thomas Schmidt. RIOT: One OS to Rule Them All in the IoT. Technical Report RR-8176, INRIA, 2012. URL: http://hal.inria.fr/hal-00768685.

[2] Bastian Blywis, Mesut Güneş, Felix Juraschek, and Jochen Schiller. Trends, Advances, and Challenges in Testbed-based Wireless Mesh Network Research. *Mobile Networks and Applications*, 15:315–329, 2010. 10.1007/s11036-010-0227-9. URL: http://dx.doi.org/10.1007/s11036-010-0227-9.

[3] Contributors of the RIOT project. Riot release specs, June 2023. URL: https://github.com/RIOT-OS/Release-Specs/.

[4] Oliver Hahm, Emmanuel Baccelli, Mesut Günes, Matthias Wählisch, and Thomas C. Schmidt. RIOT OS – Towards an OS for the Internet of Things. In *Proceedings of the 32nd IEEE International Conference on Computer Communications (INFOCOM 2013)*, pages 79–80. IEEE, 4 2013. Poster. doi:10.1109/INFCOMW.2013.6970748.

[5] Miguel Silva, David Cerdeira, Sandro Pinto, and Tiago Gomes. Operating systems for internet of things low-end devices: Analysis and benchmarking. *IEEE Internet of Things Journal*, 6(6):10375–10383, December 2019. doi:10.1109/jiot.2019.2939008.

[6] Kevin Weiss, Michel Rottleuthner, Thomas C. Schmidt, and Matthias Wählisch. Philip on the hil: Automated multi-platform os testing with external reference devices. *ACM Trans. Embed. Comput. Syst.*, 20(5s), sep 2021. doi:10.1145/3477040.

[7] Zephyr Project members and individual contributors. Test runner (twister), July 2023. URL: https://docs.zephyrproject.org/latest/develop/test/twister.html.

Concept for building edge devices application using the AI4U approach

Kamil Wołoszyn, Krzysztof Piotrowski
IHP – Leibniz Institut für innovative Mikroelektronik
Frankfurt (Oder), Germany
{woloszyn,piotrowski}@ihp-microelectronics.com

***Abstract*—This article presents the concept of a universal, modular framework that enables fast development of AI applications on edge devices. The aim of this conceptual tool is versatility in adapting to changing environments and integrating various types of sensors to accomplish specific tasks, such as computer vision or environmental monitoring. The framework is built upon the AI4U [1] approach, which facilitates the entire process of creating AI applications and simplifies the testing process.**

***Index Terms*—concept, AI4U, monitoring, vision, ML**

I. Introduction

With the tremendous growth in the amount of data being generated, traditional approaches to processing and analysis are becoming insufficient. It has become necessary to offload some of the data processing tasks to edge devices, which are closer to the source of data generation.

Edge devices, such as IoT sensors, smartphones, or medical devices, have significant potential for on-device data processing instead of relying on sending data to remote servers or the cloud. There are several reasons why moving these tasks to edge devices makes sense:

- Reduction in latency: Processing data on edge devices allows for immediate analysis and response to the collected information. It eliminates the need to send data to remote servers and wait for a response.
- Network bandwidth savings: Transmitting large volumes of data to the cloud can strain the network and require substantial bandwidth. Offloading some processing to edge devices allows for local data processing, reducing the need to transmit a significant amount of information over the network.
- Data privacy protection: In certain applications like medical monitoring or security systems, safeguarding data privacy is paramount. Processing data on edge devices eliminates the necessity of sending it to external servers, which can impact the security and confidentiality of the information.
- Efficiency and resource optimization: Processing data on edge devices can be more efficient and resource-friendly. Local processing allows for better utilization of computational power and memory resources on the edge devices.

However, transferring some of the processing tasks to edge devices is not without its challenges. Limited computational and memory resources on edge devices require algorithm and model optimization.

Despite these challenges, the transfer of data processing tasks to edge devices is becoming increasingly popular and relevant in the era of vast data generation. Further advancements in edge device technologies, algorithm optimization, and tools, along with improved performance, contribute to the growing importance and utilization of these solutions.

II. Related work

In recent years, there has been growing interest in the field of artificial intelligence (AI) on edge devices. Several studies and research initiatives have explored various aspects of deploying AI models on edge devices, aiming to leverage their computational capabilities and address the limitations of traditional cloud-based AI solutions.

Numerous edge AI frameworks and architectures have been proposed to facilitate the deployment of AI on edge devices. For instance, TensorFlow Lite [2] and PyTorch Mobile [3] are popular frameworks that provide optimized versions of their deep learning libraries for resource-constrained devices. These frameworks enable efficient inference of AI models directly on edge devices, minimizing the reliance on cloud computing. Edge architectures, such as MobileNets [4] and EfficientNets [5], have been specifically designed to achieve high accuracy with low computational and memory requirements. These lightweight models make it feasible to deploy AI applications on edge devices with limited resources, such as IoT sensors and smartphones.

Microsoft and Intel are two prominent technology companies that provide platforms and tools for building applications on edge devices. Intel has created Intel® Edge Insights, which is a comprehensive software provided by Intel that enables intelligent processing and analysis of data on edge devices. It is designed to facilitate deployment, management, and data analysis at the edge of the network. Intel® Edge Insights [6] offers a range of features and capabilities dedicated to edge environments.This platform is an open and modular software development kit based on the Open Source ROS 2 [7] (Robot Operating System 2) system. ROS2 is a flexible framework for developing robotic systems. It provides a collection of software libraries and tools that enable communication, control, and coordination among various components of a robot system. Microsoft Azure AI Platform [8] is a comprehensive suite

of artificial intelligence (AI) services and tools offered by Microsoft as part of its Azure cloud computing platform and edge devices. It provides a wide range of AI capabilities that enable developers to build, deploy, and scale AI-powered applications and solutions.

However, both platforms place a number of requirements or specially designed devices to work with them.

III. Proposed approach

The concept of the creating a modular application based on AI4U with defined interfaces for running artificial intelligence (AI) on edge devices and transmitting processed data through WSN (Wireless Sensor Network) [9] using the MicroPython [10] language is an idea that combines the benefits of modularity, AI, and efficient programming for microcontrollers. MicroPython is an optimized programming language that offers high performance on microcontrollers. As a result, modular AI applications run smoothly on edge devices, utilizing their available computational and memory resources efficiently. MicroPython allows for the creation of applications with a modular architecture, enabling easy addition, removal, and modification of AI modules. This flexibility allows for the expansion of system functionality, including various machine learning algorithms and data processing techniques.Additionally defining interfaces, you can divide the code into modules that are independent but collaborate through interfaces. This facilitates code management and allows for easier expansion and maintenance of the application. For example, if the application uses various sensors, you can define an interface that specifies common methods for reading data from those sensors. As a result, regardless of which sensors are used, the code can utilize a uniform interface. With interfaces, it becomes easy to swap out implementations of components in the application. If all components use the same interface, you can seamlessly replace one implementation with another without modifying the rest of the code. This simplifies development, testing, and maintenance of the application. Interfaces enable abstraction and separation of different layers in the application. For instance, you can define interfaces for the business logic layer, network communication, or interaction with peripheral devices. This enhances code readability, scalability, and comprehensibility.

Utilizing interfaces in the code of a concept of the modular application in MicroPython offers numerous benefits, such as code modularization, unified access interfaces, component interchangeability, testability, and abstraction and layer separation. Interfaces in MicroPython facilitate the creation of flexible and scalable code, which is particularly valuable in the context of microcontrollers and embedded systems. By leveraging interfaces, you can easily develop and maintain the application while improving its readability and modularity.

The proposal for a modular application in MicroPython will enable it to work on any device supporting AI and the MicroPython language. As a result, the application can be run on various platforms, such as microcontrollers, System on a Chip (SoC) devices, or IoT modules. In addition AI on edge devices can locally process and analyze data, identifying relevant information. This allows for the transmission of only essential processing results, eliminating the need to send the entire set of raw data. Consequently, the amount of data transmitted through the network is reduced, resulting in energy savings.

IV. The application area

The application area for this modular application can be vast and diverse, as it enables the deployment of artificial intelligence (AI) on edge devices and the seamless transmission of processed data through WSN (Wireless Sensor Network). Some potential application areas include:

- Smart Cities: The modular AI application can be utilized for various smart city applications, such as intelligent traffic management, environmental monitoring, waste management optimization, and energy consumption optimization.
- Agriculture: In the agricultural sector, the modular AI application can be used for crop monitoring, precision agriculture, irrigation management, pest detection, and yield optimization, helping farmers make data-driven decisions and improve productivity.
- Surveillance and Security: The modular AI application can enhance surveillance and security systems by providing real-time video analysis, anomaly detection, facial recognition, and intruder detection, enhancing safety and security in various environments.

Acknowledgment

This work was supported by the European Regional Development Fund within the BB-PL INTERREG V A 2014-2020 Programme, “reducing barriers - using the common strengths”, project SpaceRegion, grant number 85038043. The funding institutions had no role in the design of the study, the collection, analyses, or interpretation of data, in the writing of the manuscript, or in the decision to publish the results.

References

[1] K. Wołoszyn, K. Turchan, M. Rąpała, K. Piotrowski, AI4U: Modular Framework for AI Application Design
[2] TensorFlow Lite, https://www.tensorflow.org/lite/guide?hl=pl, last viewed 17.07.2023
[3] PyTorch Mobile, https://pytorch.org/mobile/home/, last viewed 17.07.2023
[4] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam, MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications
[5] Mingxing Tan, Quoc V. Le, EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks
[6] Intel® Edge Insights, https://www.intel.com/content/www/us/en/internet-of-things/industrial-iot/edge-insights-industrial.html, last viewed 17.07.2023
[7] ROS2, https://docs.ros.org/en/foxy/index.html, last viewed 17.07.2023
[8] Azure AI Platform, https://azure.microsoft.com/en-us/solutions/ai/, last viewed 17.07.2023
[9] WSN, https://edis.ifas.ufl.edu/publication/AE521, last viewed 17.07.2023
[10] Micropython, https://micropython.org, last viewed 17.07.2023

Distributed Energy Generators as SmartGrid Sensor Network Application

Miłosz Krysik, Krzysztof Piotrowski
IHP - Leibniz Institut für innovative Mikroelektronik
Frankfurt (Oder), Germany
{krysik, piotrowski}@ihp-microelectronics.com

***Abstract*—Growing popularity and demand to adopt renewable energy sources cause the need to use distributed generators (DGs) with converters as energy interfaces. This paper explores the concept of treating DGs as sensor networks within a decentralized hierarchical microgrid architecture. DGs can provide valuable real-time measurements and data about voltage, current, frequency, and power output, which can be considered as sensor data in the context of the microgrid system. The created sensor network communicates implicitly through changes in voltage and frequency within the grid without the need for physical connection link, therefore it is not vulnerable to communication disruptions or security threats and lowers the investment cost. In the proposed approach, part of the tasks of the microgrid controller is transferred to the local controller.**

Keywords—microgrid, hierarchical control, implicit communication,

I. Introduction

Control of an implicitly communicated microgrid (MG) refers to a control approach that operates without relying on real-time communication links between its components. In this type of control system, the various components within the microgrid make decisions and adjustments based on local measurements, decentralized algorithms, or predefined control strategies. Implicit communication control offers simplicity, robustness, and independence from potential communication disruptions. The disadvantages related to wireless control like limited bandwidth, power consumption, or interferences do not occur [1]. What is more, providing the physical connection between inverters could be difficult because of their location and the long distances between the inverters [2].

The control in hierarchical MG is divided into three groups, primary, secondary, and tertiary. The role of primary control is to maintain basic operations such as power sharing, stabilizing voltage and frequency [3]. As most of the literature describes, the primary control layer is based on droop control which is decentralized and does not require explicit communication. [4]. The secondary layer eliminates the steady-state error that leaves the primary layer. In the Secondary layer, the majority of documented approaches require direct communication [5] but up to now, no standard for communication in the secondary level has been established [4]. The role of tertiary control is system-wide optimization, coordination, and long-term planning of the microgrid. It relies on communication between the MG controller and external units like the energy market or the grid operators.

The DGs can be treated as a sensor network, where each generator with its local controller (LC) is a sensor node. The communication line is shared with the power grid. By treating DGs as a sensor network, the MG control system can collect and utilize the measurements from DGs for various purposes, such as monitoring, control, and optimization. DGs being sensor nodes are providing information about the state and performance of the system.

In this paper, the hierarchical MG control is presented with an analogy to Sensor Network. The proposed approach puts more responsibility on LC, thereby more control tasks could be performed by the implicit communication. The presented architecture of transferring part of the responsibility to the local controller is presented in Figure 1.

II. Primary and Secondary Control

A. Droop control

The inverters without direct communication links can still be aware of what is happening with the other units with the use of the droop control. It is a decentralized method used in power systems, including microgrids, to regulate the power output of multiple energy sources (such as inverters or generators) and maintain stability. It works by adjusting the frequency or voltage of each energy source based on changes in the total power demand. The curves of the droop control method are presented in Figure 1.

Each DG adjusts its output reactive power based on changes in voltage. In case of decreasing the grid voltage, indicating a higher reactive power demand or increased system losses, each generator increases its power output proportionally according to its voltage droop characteristic. Conversely, if the grid voltage increases, each DG reduces its power output accordingly. This adjustment is made independently by each DG without direct communication.

In addition to voltage droop, frequency droop is also employed. When the grid frequency decreases, indicating a higher power demand, each generator increases its power output proportionally according to its predetermined droop characteristic. Similarly, if the grid frequency increases, indicating a lower power demand, each generator decreases its power output accordingly.

B. Control tasks

If the voltages and frequency are also used and analyzed as communication signals between DGs, then the primary control could fast detect faults like voltage swell or sags, short circuits, or overcurrent conditions.

Plug and play is highly desired in MG. Easily integrating new distributed energy resources is critical to gain flexibility. Without explicit communication setup and prior knowledge of the system topology, this task could be effortlessly achieved in implicitly communicated MG.

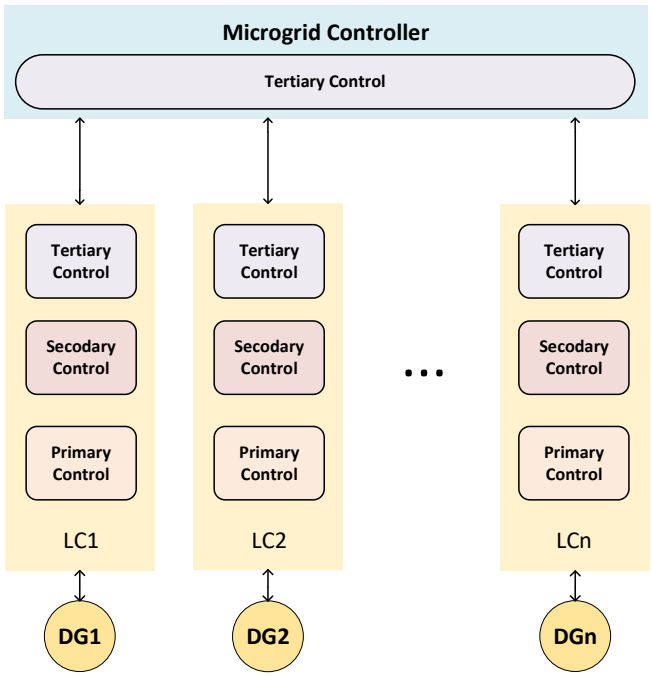


Fig. 1. Architecture of proposed hierarchical microgrid

III. Tertiary Control

While the primary and secondary control layers are typically implemented within LCs associated with individual components (such as distributed generators or energy storage systems), tertiary control is often implemented at a higher-level central controller or a supervisory control system. The proposed approach is to transfer some of the responsibility to LC, and thus the DG itself will be more autonomous. Set-point adjustments could be to some point handled by the LC. The proximity of LC to DG is beneficial due to lower propagation time and reduced vulnerable information exchange.

In some microgrids, the tertiary control layer may determine the participation of individual DGs in providing ancillary services, such as frequency regulation or voltage support. The local controller can then implement the necessary control strategies to meet these service requirements while maintaining local objectives. There again, the use of implicit communication with before mentioned ability to fault detection ensures quick reaction.

The tertiary control layer implemented in local control can optimize the utilization of energy storage systems. By analyzing the state of charge and discharge rates of energy storage units, the tertiary control layer can determine the most efficient and cost-effective storage strategies. The local controllers then adjust the energy storage operation accordingly.

However, it is important to note that implicit communicated tertiary control on LC may have limitations in terms of system-wide optimization and coordination compared to explicit communication implemented in MG controller approaches. The lack of real-time communication can restrict the ability to adapt to dynamic conditions and may result in suboptimal performance in certain scenarios. Therefore, the trade-off between implicit communication

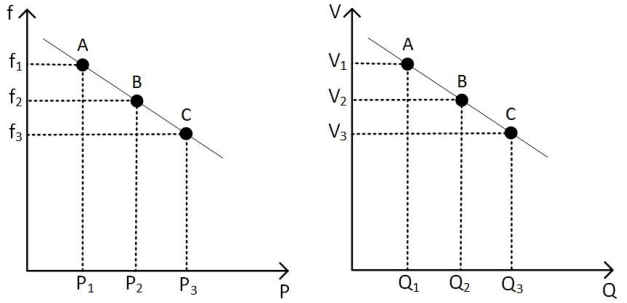


Fig. 2. Example of curves of droop control

operation and explicit communication operation should be carefully considered based on the specific requirements and constraints of the microgrid system. The trade-off between depends on the goals, complexity, and capabilities of the microgrid system.

IV. Conclusions

The paper presented the approach to treat the hierarchical microgrid as a sensor network. LC can take advantage of implicit communication and behave as sensor nodes. By leveraging the measurements provided by DG, a vast amount of data becomes available for monitoring, control, and optimization purposes. It allows to scale up of the Microgrid with the use of plug and play capability without the need of providing additional unnecessary communication links. The implicit communication allows to transfer of part of the duties of the tertiary control from the upper MG controller to the LC and therefore part of control relies on decentralized decision-making. These duties cover ancillary services, energy storage management, or set-point adjustments.

Acknowledgment

This work was supported by the European Union ebalance-plus project under the H2020 grant no. 864283. The funding institutions had no role in the design of the study, the collection, analyses, or interpretation of data, in the writing of the manuscript, or in the decision to publish the results.

References

[1] M. Hua, H. Hu, Y. Xing and J. M. Guerrero, "Multilayer Control for Inverters in Parallel Operation Without Intercommunications," *IEEE Transactions on Power Electronics,* vol. 27, no. 8, pp. 3651-3663, 2012.

[2] Y. Li and F. Nejabatkhah, "Overview of control, integration and energy management of microgrids," *Journal of Modern Power Systems and Clean Energy,* vol. 2, no. 3, pp. 212-222, 2014.

[3] U. Ekanayake and U. Navaratne, "A Survey on Microgrid Control Techniques in Islanded Mode," *Journal of Electrical and Computer Engineering,* pp. 1-8, 2020.

[4] J. M. Rey, P. Martí, M. Velasco and J. M. a. M. Castilla, "Secondary Switched Control With no Communications for Islanded Microgrids," *IEEE Transactions on Industrial Electronics,* vol. 64, no. 11, pp. 8534-8545, 2017.

[5] Y. Khayat, M. Naderi, Q. Shafiee, M. Fathi, H. Bevrani, T. Dragicevic and F. Blaabjerg, "Communication-less Optimal Frequency Control of Islanded Microgrids," in *20th European Conference on Power Electronics and Applications*, Riga, Latvia, 2018.

Environment monitoring backend and dashboard

Igor Koropiecki, Krzysztof Piotrowski
IHP - Leibniz Institut für innovative Mikroelektronik
Frankfurt (Oder), Germany
{koropiecki, piotrowski}@ihp-microelectronics.com

***Abstract*—The paper presents the backend implementation for receiving, processing and visualizing environmental data collected through a Wireless Sensor Network (WSN) distributed over the twin city of Frankfurt (Oder), Germany and Słubice, Poland.**

***Index Terms*—middleware, services, smart city, dashboard**

I. Introduction

The technologies available today, allow us build systems that improve our lifestyle and create safer environments. From simple embedded devices to cloud-based solutions, we have readily available building blocks that can change how we interact with our surroundings. The general concept that focuses on the technology making our lives easier is called smart city. It utilizes data-driven solutions to optimize city operations, enhance the quality of services, improve sustainability, and promote citizen engagement [1]. A smart city system often deploys an advanced infrastructure with sensors and digital systems to collect and analyze data, enables informed decision-making and efficient resource allocation [2]. Examples include intelligent buildings, monitoring of traffic, environment or crowd congestion. The definition and examples analysis shows that the most important aspect is the ability to acquire, process and store vast amounts of data. Therefore, it is necessary to have a robust platform that enables seamless communication, efficient data storage and the ability to quickly react to changes [3] [4]. Overall, the backend should act as a hub that receives real-time data from various environmental sensors and devices, which enables continuous monitoring and data acquisition.

II. Related Work

The design and implementation of a backend with a web interface is a well-known and popular topic. Notable solutions include GigaSpaces, ThingsBoard, KaaIoT or alternatives from major tech companies, such as Microsoft, Amazon, Google, IBM, Bosch or Cisco. Also, there are plenty of abandoned non-commercial or academic solutions which lack publicly available binaries or source code.

III. Design

The presented work was done within a project [5] that aims to monitor environmental parameters related to weather, air pollution, water tanks, rivers and soil. Certain aspects were imposed from that project, such as system requirements, data platform, measurement frequency and overall functionality. The proposed logical structure reflects the physical layout of the deployment. Fig. 1 presents the key components in the structure and relations between them. Additional components represent node locations (grouping) and device locations (specifying offsets). The measurement system collects data

Fig. 1. Components and relations in the structure

from a distributed network of 110 measurement stations. It is forwarded to gateways which pass raw and unaltered measurements for storage. Later, the data is processed, stored and displayed. To simplify the processing and avoid inaccurate analysis of asynchronous data, time windows were used. The width of the time window was set to one hour.

IV. Implementation

In order to implement the data exchange and storage, a platform called smartDSM [6] was used, which allows to create services that implement functionalities by interacting with an interface that offers operations similar to those in tuple spaces. Instead of a tuple, the data structure is called a variable. Each write results in a new entry, which is owned by the stakeholder that runs the service. The data can be shared between stakeholders on an opt-in basis. The services can subscribe to actions performed on the variables, which makes it easy to create a distributed event-driven system. Each measured parameter (temperature, wind, etc.) has its own variable that contains unaltered measurements and their metadata, such as geolocation or origin station. Several services were created that implement the functionalities required for the project.

A. Gateway Service

The gateway acts as a bridge between the storage and the WSN. It receives measurements over a serial interface and uses the structure definition to decode the station type and parameters. It forwards the parameters to the storage platform.

B. Weather Service

The weather service acts as a data processor. It listens to data from the gateway and converts it into meaningful structures, such as weather or pollution information. When defining a structure, it is possible to specify the output variable name, required parameters (raw measurements) and custom logic that is invoked when all measurements for a given slot are available. The Fig. 2 visualizes the concept. Raw measurements (e.g. temperature, etc.) are received, then the

timestamp, network and node addresses are used to identify a slot (time window) within a structure (e.g. weather) for a given node that sent the data. When a slot is full (all parameters received), the processed data is written to the output variable.

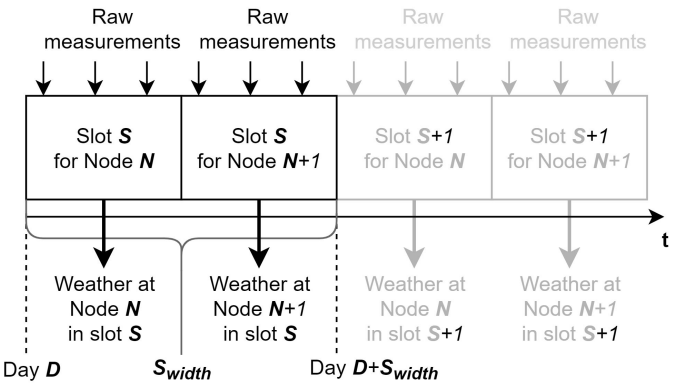


Fig. 2. Visualization of processing and time windows

C. Dashboard Service

The dashboard service acts as the user interface. It allows to view the list of all stations with locations and types of parameters. It can be used to view heat maps. The administrators can create (or accept auto-generated) warnings that are visible to everyone. It allows to check weather and air pollution history averaged from stations across the twin city. Available history is currently limited to a 24-hour timeframe. The frontend (Fig. 3) has been implemented using the React framework, with addition of several libraries, such as OpenStreetMap (Leaflet), Recharts, i18next and others. The backend of the dashboard has been created using the Java Spring Boot framework.

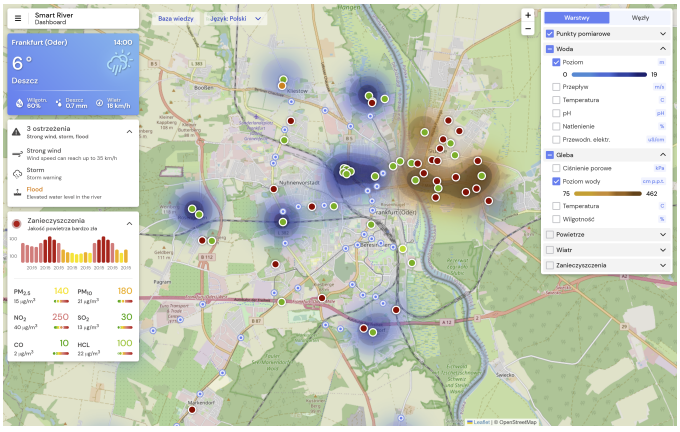


Fig. 3. Visualization of processing and time windows

V. Evaluation

The footprint of the web application is 4.5 MB. The largest file (3.5 MB) is the JavaScript code with the React framework, libraries and implementation. The dashboard has 4 public API endpoints: status (weather, pollutants, warnings), network structure, parameter names and values. Each endpoint was stress tested by sending 100 concurrent requests, two times. The aim was to check the overall performance and caching behavior. The results were merged for each endpoint and presented on Fig. 4. The chart shows that the response times of the first 100 concurrent requests for each endpoint have a linearly decreasing trend with the first response time being lower than the consecutive ones. It is caused by the cache implementation – first request blocks until calculations are available. The next 100 concurrent requests have much lower and steadier response times – the cached calculations are returned. The network endpoint has a significantly higher response time and deviation between consecutive calls, for both of the iterations. This is most likely caused by the size of data to process (final response is 865 kB).

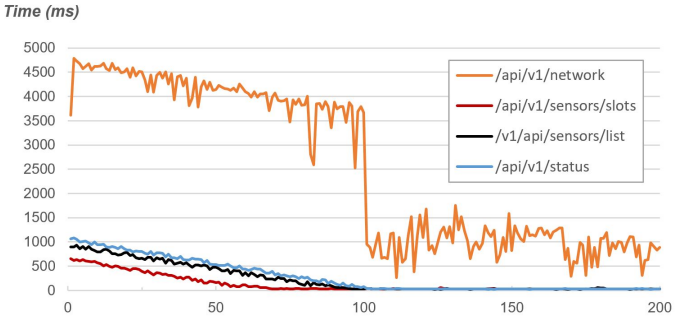


Fig. 4. API stress testing results

VI. Conclusion and Further Steps

The presented backend enables data collection from sensor networks, processing and real-time visualization. It is a strong foundation for continued exploration and development. Further efforts could focus on optimization and processing to provide more meaningful insights into collected information.

Acknowledgement

This work was supported by the European Regional Development Fund within the BB-PL INTERREG V A 2014-2020 Programme, “reducing barriers - using the common strengths”, project SmartRiver, grant number 85029892. The funding institution had no role in the design of the study, the collection, analyses, or interpretation of data, in the writing of the manuscript, or in the decision to publish the results.

References

[1] V. Albino, U. Berardi and R. M. Dangelico, Smart Cities: Definitions, Dimensions, Performance, and Initiatives, Journal of Urban Technology, 2015, 22:1, 3-21.
[2] A. Ramaprasad, A. Sánchez-Ortiz and T. Syn, A Unified Definition of a Smart City”, Electronic Government Conference, 2017.
[3] S. S. Hajam and S. A. Sof, IoT-Fog Architectures in Smart City Applications: A Survey. China Communications (Volume: 18, Issue: 11, November 2021).
[4] L. Wieclaw, V. Pasichnyk, N. Kunanets, O. Duda, O. Matsiuk and P. Falat, Cloud Computing Technologies in “Smart City” Projects. IDAACS Conference, 2017.
[5] E. Michta, K. Piotrowski, P. Powroznik, R. Rybski, R. Szulim, U. Kolodziejczyk and J. Kostecki, Flood embankments monitoring system, 13th Scientific Conference on Measurement Systems in Research and in Industry. Zielona Góra, 2020.
[6] I. Koropiecki, K. Piotrowski and R. Szulim, SMARTDSM: Data Space Middleware for Distributed Measurement Systems, 14th Scientific Conference on Measurement Systems in Research and in Industry. Łagów, Poland, 2022.