electronics

MDPI

Article

Impact of Real-World Energy Consumption Variance on Internet of Things Node Lifetime Predictions

Silvia Krug [1,2,*], Tino Hutschenreuther [2], Hannes Toepfer [3] and Mattias O'Nils [1]

[1] Department of Computer and Electrical Engineering, Mid Sweden University, Holmgatan 10, 851 70 Sundsvall, Sweden; mattias.onils@miun.se
[2] System Design Department, IMMS Institut für Mikroelektronik- und Mechatronik-Systeme Gemeinnützige GmbH (IMMS GmbH), Ehrenbergstraße 27, 98693 Ilmenau, Germany; tino.hutschenreuther@imms.de
[3] Advanced Electromagnetics Group, Technische Universität Ilmenau, Ehrenbergstraße 29, 98693 Ilmenau, Germany; hannes.toepfer@tu-ilmenau.de
* Correspondence: silvia.krug@miun.se

Abstract: Node lifetime predictions are a crucial design time tool when developing Internet of Things (IoT) solutions with constrained energy budgets. However, this analysis is typically based on simplistic analyses of current consumption values based on datasheets and static duty cycles. This leads to an optimistic prediction of the node lifetime. Real-world measurements show a variation in the energy consumption that can significantly reduce the predicted node lifetime. In this paper, we aim to analyze the impact of the experienced variation for a given IoT platform and typical sensing tasks. To do this, we present a design case study in smart agriculture, where we perform empirical measurements to analyze energy consumption variability and its effect on as well as challenges regarding different design decisions. In addition, we suggest an empirical modeling method to enhance the energy efficiency of IoT nodes. The results show that the variations have a significant impact on node lifetime and should be considered in estimations in the future, as they show the design space to be considered when building robust systems.

Keywords: internet of things; node design; energy consumption; variation impact

Citation: Krug, S.; Hutschenreuther, T.; Toepfer, H.; O'Nils, M. Impact of Real-World Energy Consumption Variance on Internet of Things Node Lifetime Predictions. *Electronics* **2024**, *13*, 4578. https://doi.org/10.3390/electronics13234578

Academic Editor: Petros Nicopolitidis

Received: 29 October 2024
Revised: 17 November 2024
Accepted: 19 November 2024
Published: 21 November 2024

1. Introduction

In times of climate change, sustainability becomes more and more important, even for electronic devices. The rise of Internet of Things (IoT) fosters many small battery-powered devices that capture various data that are analyzed in bigger systems. The sensors, on the one hand, help to understand environments and provide the basis for data-driven decisions that enable sustainable development, but the devices within the IoT should be energy efficient and sustainable themselves. Therefore, the node energy consumption and carbon footprint of devices and systems remain relevant and receive more attention [1,2].

In order to develop sustainable IoT systems, the node lifetime and related energy consumption are crucial measures. Several external and internal factors will alter the energy consumption profile for a given activity as described in [3]. In addition to hardware choices, software design is also a crucial aspect [4,5] if it does not exploit energy-saving mechanisms. Choosing an appropriate node design is, therefore, a challenging task. Estimating the energy consumption too optimistically results in quality issues as nodes are supposed to ensure a certain guaranteed lifetime. This is especially true for cases where the nodes are placed once and are hard to access afterwards, such as outdoor locations that are not to be disturbed or hazardous industrial sites. Achieving such guarantees is, however, often only possible by designing an appropriate buffer lifetime for the given application.

The goal in most applications is typically to provide devices and systems with a long lifetime under real-world conditions by reducing the energy consumption of the devices during the design phase as much as possible and potentially powering them using energy

harvesters. At this early design stage, the energy consumption is typically estimated based on analytical models that use datasheet values or a few characterization measurements of the envisioned duty-cycle activity as well as the sleep current as input. This is a valid option to obtain a rough estimate of the overall power consumption. It will, however, be somewhat optimistic since the typical consumption given by the datasheet is a rather ideal measure.

In addition to that, the energy consumption of a node will not be constant over time even for the same activity or the same hardware [6]. Therefore, continuous real-world measurements of the envisioned applications can show a significant variation for the same application task on different nodes. The reasons for this are manifold, but they make realistic lifetime estimations even more challenging and the results from simple analytic approaches rather unrealistic.

In ref. [7], the authors were able to show this effect for Unmanned Aerial Vehicle (UAV) lifetime estimation based on empirical data for distinct flight maneuvers, which was later used to predict battery life and a recharging schedule for the UAV. This approach exploits the variance reported in the measurements for a more realistic lifetime estimation.

To the best of our knowledge, this has not been transferred to IoT devices so far, i.e., the variance in the task-specific energy consumption is currently not considered as a design metric in IoT. Our hypothesis is that this, however, has the potential to provide more realistic energy consumption profiles and, thus, more realistic lifetime estimation at the design phase.

In this paper, we verify this hypothesis by performing a case study for a smart agriculture sensor node, in which we measure the energy consumption of different node configurations and activities within an design space. The campaign allows us, on the one hand, to examine the expected variance in the measurements and, at the same time, give insights into the node design to optimize the energy consumption. In addition, we use the statistics provided by the measurement to verify that using the variance results in better estimations than the traditional simple estimation approaches and build an empirical energy consumption model.

The remaining parts of this paper are organized as follows: in Section 2, we review existing work on energy modeling and open questions regarding this. Afterwards, we present our method in Section 3, where we detail the case study, the measurements, and the hardware and software used. In Section 4, we present the results of the case study as well as the comparison of a variance-aware estimation over traditional lifetime estimations, followed by a discussion of the impact on future design choices and open challenges in Section 5. Finally, the paper is concluded in Section 6.

2. Related Work

Energy modeling of devices and derived lifetime estimations based on an energy budget are nothing new and have been addressed in various ways [8,9]. However, the complexity of the problem with different levels of detail from high-level system perspectives to low-level circuit behavior renders this topic still open and challenging. The need to build sustainable systems [1,2] in the future only increases the importance of the field. This is especially true if, from a carbon footprint perspective, optimizing for energy or a low carbon footprint might be contradictions, as in ref. [1].

To assess a system, several factors are relevant [3], and researchers have focused on different areas. Essentially, there are studies targeting the overall nodes lifetime [10–13] and node design [14–20] in general, while others focus on subsystem components. In this category, the authors in [20] show the value of energy modeling for node design. The authors in [16] show how mission-oriented node design affects the energy consumption. Recent works focus on the reconfiguration of nodes [15], software-related aspects [14], and battery state estimation [10,11], while older papers target, e.g., the impact of power levels [13], system level design [12], peripheral impact [18], and system trade-offs in general [19].

Among those focusing on a subsystem of the device, networking protocols are very popular since communication is usually the major source of energy consumption. This field has been and still is very active, e.g., see [21–31]. This list shows the efforts targeted toward energy-efficient communication and the modeling or estimation of the corresponding consumption. Within the group, there are different aspects that are addressed.

In ref. [28], the authors show design considerations regarding protocols in general. These works focus on enhancing protocol-related energy consumption by employing energy-efficient mechanisms. In [21], the authors describe efficient scheduling schemes while the authors in [24] focus on routing aspects. The impact of different media access schemes is analyzed in [22]. Packet size is another option to optimize the energy, as shown in [31]. Other works show the challenges with dynamic environments when dealing with wireless communication, e.g., in [29,30] or what overhead is introduced by different protocols [27]. Another important point is how communication affects other design considerations such as node placement [23] and the suitability of harvester-powered devices [27].

In addition to node optimization, there are also approaches that focus on the application at hand [32–36]. This is similar to [16] and is crucial since the application or task that the network has to accomplish defines the constraints on the system activity and, thus, directly affects the energy consumption. Similarly to the previous areas, there are many different aspects to focus on. This ranges from partitioning and offloading [32,35] part of the workload to mitigate communication costs, to adapting sample rates [36], assessing the cost of data acquisition [34], and cross-device system aspects [33].

Regarding the modeling of device energy consumption, there is a similar diversity in methods and focus areas. In general, there are models for protocols and corresponding mechanisms as in [37], models predicting battery performance [38], tools to model devices [39], and networks [40]. For a design decision at an early development stage, models covering the device, the required components, and activities to solve the given task are required and, thus, the given models would be only applicable for a subset of this. In addition, all mentioned works use different approaches to achieve the modeling, which makes it hard to integrate different models.

Regarding the applied approaches, in recent years, machine learning (ML)-based models have gained attention as in [41,42], where the authors try to cover the node design. One major drawback for these approaches is the need for data to learn interesting information. This is difficult at the design stage, when one wants to estimate how a system will perform. One example for the traditional datasheet approach is presented in [43]. Such approaches are analytical and, thus, allow for estimations without any measurements but are most likely too optimistic with respect to the lifetime, as, e.g., ref. [44] shows. Models based on empirical data are, therefore, preferred and also very common, e.g., [45–47]. In ref. [45], the authors show how to derive models from power traces for one MCU. Regarding communications, the authors in [46] use LoRa field measurements to allow for the design of long-living sensor nodes. Similarly, the authors in [47] show how to use measurement-based models to estimate node lifetime in a network. Another interesting option is to focus on the behavior of the nodes and perform a black-box analysis of the desired activity by a given system. This was shown in [48] for flight maneuvers of UAVs and in [49] for nodes with adaptive behavior that changes due to the environment.

Another important topic is how to collect empirical data to evaluate a node's energy consumption. Several papers target the design of benchmark experiments [50–52]. In ref. [52], the authors introduce a method to accurately benchmark devices in an IoT context to later model battery performance. Similarly, in [51], the authors present an approach to benchmark networking aspects with a focus on low-power applications. A more recent approach is presented in [50], where the authors aim to benchmark edge devices. These approaches describe only the tools necessary to capture data but do not go into an analysis of the nodes or the modeling of the energy consumption. However, they add building

blocks to the current work, and we consider some of the aspects when designing our own measurement campaign.

In ref. [6], the authors highlight that there is, in general, a variance if system benchmarks are performed. This variance also exists if the experiment is repeated on the same hardware. The authors give guidelines on how to minimize the effects with a possible increase in overall energy consumption in order to provide fair benchmark comparisons. In real-world deployments, we need another approach to handle the variance and rather have to be aware that it is there. This is especially crucial if we evaluate networked systems where the networking itself can lead to changes in the activity patterns of the node and, thus, increased power consumption [29].

Therefore, models that are based on empirical measurements that take the variance into account should provide better estimates. A similar approach has only been reported for UAVs in [7], where empirical data on a behavior level of the node were used to estimate the remaining flight time. We try to adopt this to IoT devices in this paper.

Recent work, e.g, [53], puts emphasis on the need for energy-efficient nodes, as the amount of small smart devices keeps increasing, and, thus, the total energy consumed by these devices. Consequently, there are many approaches that target the design of energy-efficient nodes. For example, in [54,55], the authors tried to achieve complete energy autonomy. Understanding the node's energy budget is essential for this. Other recent works focused on reducing the energy requirements of AI methods deployed at the edge [56,57]. In this case, understanding the design options, their respective energy requirements, and optimization potential is crucial. Our work support this research by providing a model to explore the impact of design choices on overall energy consumption.

3. Methodology

3.1. Case Study Setup

In this paper, we aim to characterize a sample IoT node developed for outdoor use in smart agriculture. This node is supposed to measure micro-climate within orchards with several possible configurations and is part of our modular sensor platform [58]. Hardware configuration options consist in the inclusion of up to three sensors that measure air temperature and humidity combined, as well as the presence of an additional leaf wetness sensor. The latter is essentially a variable resistor with a value depending on the amount of water on its surface. This results in six variants in total based on the same hardware. Figure 1 shows one node variant deployed in an orchard.

Figure 1. Micro-climate node with two temperature sensors with housing to shield solar radiation and leaf wetness sensor deployed in real outdoor environment.

The purpose of the given nodes is to monitor temperature and humidity in one to three different vertical positions within a tree and optionally also monitor the leaf wetness amount and duration as an indicator for potential pest infection risks. These nodes have been deployed in different agricultural settings since 2020 as, e.g., described in refs. [58,59]. The temperature data are used in for multiple purposes in this case: one is to monitor climate in general, but the other is to detect states that are critical for the harvest, such as frost events in spring, missing temperature gradients between day and night for color development, and excessively high temperatures in summer. Part of that data are also used as input for models to estimate the stress of the plants and, thus, have to be pushed regularly to other systems [59]. As such, the application requires continuous data acquisition by the sensors at a sampling and sending interval of 15 min. This sets the constraint for activity in this case. The desired node lifetime is a minimum of two years for all options: we use this as a design goal for the battery-powered device in our case study.

The hardware selection plays a vital role regarding the energy consumption. Therefore, we provide details on the chosen hardware next. Then, the node is built from the components listed in Table 1, with given datasheet values for the current draw.

Table 1. Node hardware components with datasheet energy characteristics.

Component	Peripheral	State	Current Consumption
SoC (nRF52840) [60]	Radio	TX	6.4 mA
		RX	6.53 mA
	MCU	Sleep—Idle	1.5 µA
		Calculating	6.3 mA
	ADC	Sampling	1.24 mA
	I2C	Sleep	6.3 mA
		Active	6.3 mA
Temperature/Humidity Sensor (SHT20) [61]		Idle	0.15 µA
		Sampling	3000 µA
I2C Muxer (TCA9546A) [62]		Idle	1 µA
		Active	7.02 mA
Battery (2 AA)		2000 mA h	

The described node corresponds to a traditional sensor as depicted in Figure 2 without any energy-harvesting capabilities to simplify the study for now. Such an energy-harvesting module could, however, be easily added to the study and method. We use the nRF52840 System on Chip (SoC) device from Nordic Semiconductor (sourced via DigiKey, Munich, Germany) that includes an MCU and a BLE-capable radio transceiver and list the current draw of the used sub-components and peripherals in Table 1. In addition, we list all external components and sensors of the node with their respective current draw. Since the leaf wetness sensor is essentially a resistor, we use the ADC of the nRF52840 as a sensor. The node can be equipped with up to four Sensirion SHT20 sensors (sourced via DigiKey, Munich, Germany) accessible via a bus multiplexer. The values for the current draw are taken from the datasheet of the respective component, and the system is operated on a supply voltage of 3 V.

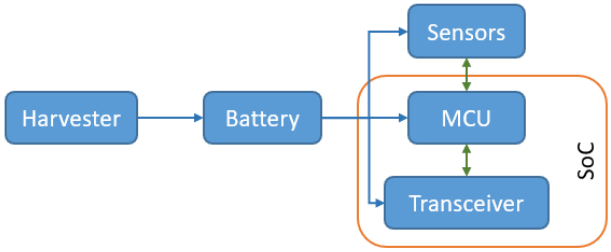


Figure 2. Basic components and interactions of a traditional sensor node, highlighting the energy and dataflows between the components considered in this study.

In addition to the variation of the number of attached sensors, there are a few additional design choices. Here, we focus on two aspects only: the power source for I2C and connected sensors, as well as the usage of UART for debugging. The power supply of the I2C bus can be connected directly to the battery or to a GPIO pin in our design. In the first case, the bus is always powered while it can be switched of via the pin in the second case. Regarding the usage of UART, it is well known that initializing the interface to debug the node increases its energy consumption. We still include one variant with and one without initialized UART to highlight the differences. As a result, we have, in total, a design space of 24 options (six sensor configurations and four power options).

In order to understand the importance of these variations and the empirical variance of the energy consumption, we perform the following steps (cf. Figure 3). The steps are described in the following subsections.

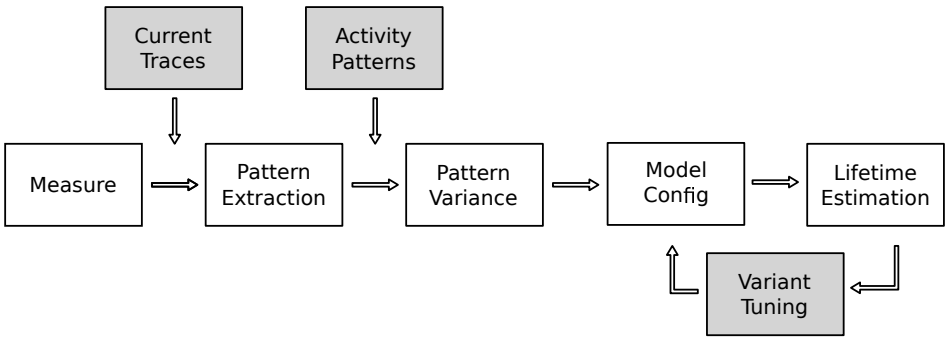


Figure 3. Flowchart showing the steps of our proposed method to estimate the node lifetime based on empirical data.

3.2. Measurement Campaign

We performed energy consumption measurements for our example sensor node as described in the previous section. To do this, we approach the system similar to [7] and use a black-box approach to characterize the system behavior when executing distinct activities. As activities, we consider the following isolated actions:

1. Read Leaf Wetness
2. Read SHT
3. Read Battery

To characterize each option, we set up a number of scenarios according to Table 2. We use two distinct base PCB versions in this case because the used SHT20 sensor can be configured to two distinct addresses on the I2C bus only. Since the requirement is to have up to three sensors in the final system, we chose to add an I2C multiplexer IC to the design, which allows us to switch up to four channels without causing address conflicts. The design using the multiplexer is denoted as i2c mux in the table and the other one without it as i2c single. The software used for each test is an actual firmware that is either used for functional tests of sensors or the actual application. We did not add any measurement-specific features to the software, such as pin toggling. Therefore, we do not expect any influence of the measuring procedure on the measured energy consumption.

The sensors are powered either via VDD, resulting in an always-on mode, or via a GPIO pin of the MCU, which allows for the sensor to be switched off when it is not needed. In addition, we check the impact of the UART being enabled for each test configuration. The UART is used for debugging purposes only in this design and, therefore, useful during development. During normal operation, it is not needed but would, however, be handy for fault diagnosis in the field, in case of errors.

Table 2. Tests performed to estimate activity energy consumption and explore the design space.

Test	Activity			Current Consumption			PCB Version
	SHT21	LW	Bat	Network	VDD	GPIO	
Test 1		x				x	i2c single
Test 2	1				x		i2c single
Test 3	2				x		i2c mux
Test 4	2					x	i2c mux
Test 5	2	x				x	i2c mux
Test 6			x				i2c single
Test 7	1		x	real		x	i2c mux

To perform the measurements, we prepared the firmware reading the corresponding sensor, including its activation if needed, in LumenRadio's MiraOS (https://lumenradio.com/products/mira/, accessed on 29 October 2024). MiraOS features the MiraMesh communication stack based on BLE. The node executes the desired activity sequence repeatedly for 100 iterations, with a distinct sleep phase of inactivity in between two iterations. Once the process is complete, we activate a LED resulting in a higher current draw, indicating the end of the test in the measurement signal.

We then perform the actual power measurement using an Agilent N6705B Power Analyzer from Keysight, sourced directly from Keysight, Böblingen, Germany. The analyzer device powers the node with a programmed arbitrary wave signal, in this case a step function from 0 V for 10 s and then switching to 3 V. We use the initial delay to safely capture the node startup and then capture the current draw of the system over the duration of the measurement run. The total duration is set to allow for approximately 100 repetitions of the measurement cycle with some guard time (10 s) between individual activities. Afterwards, we export the data as CSV-file for post processing and analysis. Figure 4 shows the interface between node and analyzer as a dotted line, with the power analyzer replacing the battery.

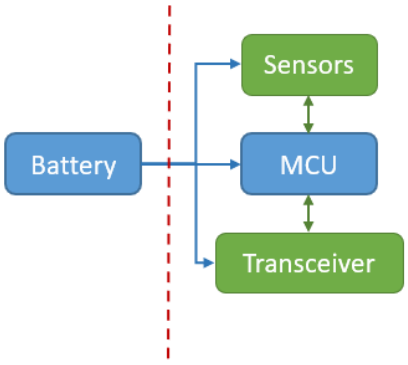


Figure 4. Schematic view of the energy measurement point used in this study. The analyzer both powers the node and captures its energy consumption.

This results in a black-box scenario, since the power analyzer cannot distinguish between individual components causing the observed current draw. However, we believe this is sufficient if we analyze different activities as described above to obtain an estimate of the total power requirement and potentially energy-hungry components.

3.3. Data Extraction

Initially, we perform a manual check of the first activity pattern to derive some key information like pattern duration and current peaks as well as the typical sleep current between patterns. Figure 5 explains this in a schematic manner. We use software that follows a duty cycle, switching between sleep (equals no activity) and the actual activity performed by the node. Therefore, no further synchronization is needed if the pattern can

be clearly distinguished from the sleep phase. Based on the initial pattern, we define the threshold, window length (red box), and pattern offset. To obtain all repetitions of the pattern in the trace, we let the window run over the measurement file, and if the hull-curve is above the threshold for a suitable duration, we treat the identified signal as one pattern. Finally, we verify the detected patterns for wrongly detected patterns and prepare a clean dataset for further evaluation.

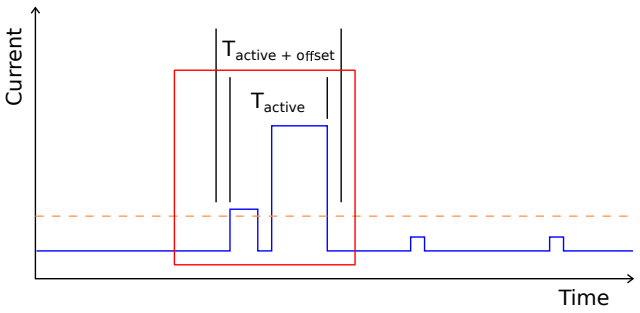


Figure 5. Schematic view of the pattern identification process. Each current consumption trace is checked for activity spikes (T_{active}) that belong to the desired pattern. A threshold (orange dotted line) is used to filter other spikes in the signal.

In order to compare the pattern duration and, thus, calculate the energy consumption per pattern, we normalize the time axis of all patterns. This means each pattern starts at 0 s, and the time column of the pattern shows the duration. The spacing is kept the same; only the offset due to repetitive execution is removed. This, then, allows for an easier comparison of the patterns and the identification of variances.

This results in $90 + x$ patterns per activity, which we then use for statistics and base for node lifetime modeling as well as design choices.

3.4. Network Overhead Estimation

Regarding the required communication, there are several things to consider. First the nodes need to connect to the network and find a route to the gateway. This connection is maintained during operation. If changes happen, e.g., due to node failure or energy-saving mechanisms, the nodes update their routes accordingly or have to search for alternatives. As a result, an energy model for node lifetime estimation has to consider several of these aspects that are, however, difficult to predict beforehand.

In order to estimate the impact of variable networking operations such as route finding and updating, we study the observed connection changes of each node in a real-world deployment. This is possible since each packet contains information about the sending node and its current packet as well as timestamps and packet sequence numbers.

Based on this information, we then extract the following:

- The number of parent changes and, thus, events with route searching;
- The duration of a connection, defined as the duration without a parent change;
- The number of connectivity losses, defined as periods with a parent change that result in missing packets as indicated by the sequence numbers;
- The duration of interrupted connections as periods of prolonged attempts to rejoin the network.

We then use these metrics to estimate the power consumption of the individual networking events in addition to normal data exchanges, similar to the measurement campaign, as in ref. [29].

As a sample scenario, we use the same real-world scenario of a smart vineyard as described in [58,59]. The data are collected over one month of operation and cover all deployed nodes. Figure 6 shows the topology and is annotated with information on interruptions and connection duration.

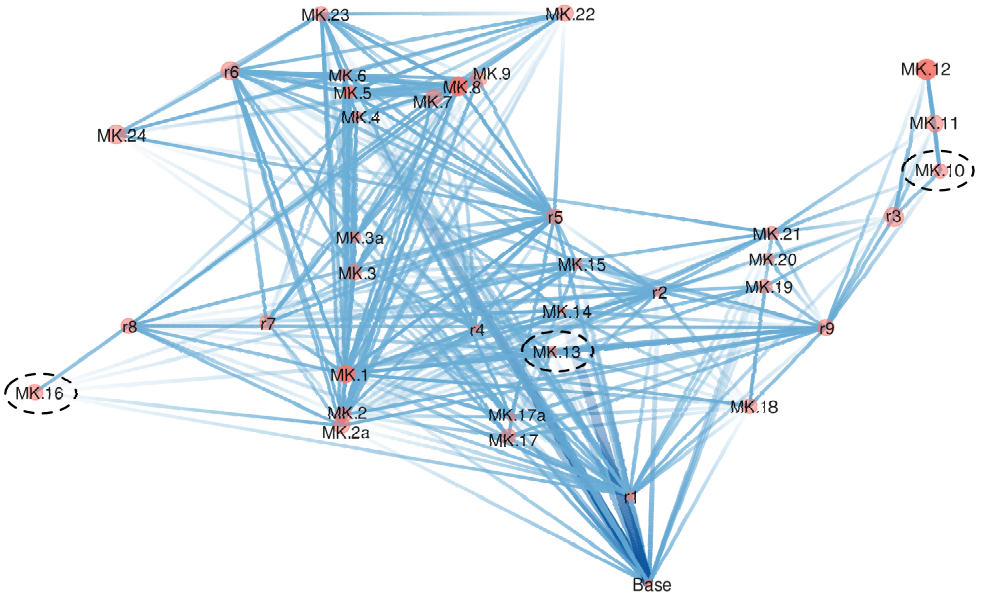


Figure 6. Network topology of the sample scenario with the sink node (Base) located at the bottom and active links as lines between them. On top of the topology, we visualize connection interruptions per node (red) and duration (link width and color). Selected nodes for further analysis are highlighted with black dashed circles.

The nodes are configured in such a way that they attempt to repeat the transmission of a packet for a maximum of four times before considering a connection broken. Packets that cannot be sent or that are lost on the way are not resent. Due to the locations of the nodes in the network, we also experience different traffic patterns. Nodes at the edge of the network have to handle almost no traffic from other nodes while more central nodes do a substantial amount of forwarding for child nodes. This is also taken into account in the estimation by selecting few nodes from the network for a detailed analysis. The nodes are as follows: M.10, an edge node with child nodes; M.13, a central node showing stable behavior; and M.16, a node at the edge of the network.

3.5. Node Lifetime Estimation

Bases on the derived patterns and the resulting statistics, we then perform three sets of node lifetime estimations. The first one is a traditional datasheet approach, where we use the timing from the measurements as a base input together with datasheet current consumption values of all sensors and the MCU with corresponding peripherals. The second method uses the same approach but replaces the datasheet current values with the measured ones. Finally, we look at the statistics and model the lifetime based on average consumption over all patterns per activity. The results of these approaches are then compared to highlight the importance of variance in node energy consumption even if the same activity is executed.

For the first two approaches, we calculate the energy consumption in W h per activity using the following formula:

$$E_A = \int_{t_0}^{t_A} I_{comp_A}(t) * V_{CC} \tag{1}$$

where t_A denotes the duration of the activity and I_{comp_A} is the current consumption of the active components for activity A. t_A is derived from the empirically captured consumption profile, while I_{comp_A} corresponds to the values in Table 1 for the datasheet approach and the empirical value from one pattern for the second estimate.

For the third evaluation method, we use a different approach. There, we calculate the energy consumption of the activity based on the energy per pattern E_P by calculating the mean of all observed patterns.

$$E_{A_P} = \frac{\sum_{i=0}^{n} E_P(i)}{n} \tag{2}$$

The energy per pattern is calculated as a step-wise integration for all sample points i of the corresponding pattern P.

$$E_P = \sum_{i=0}^{n} (t_P(i) - t_P(i-1)) * I_P(i) * V_{CC} \tag{3}$$

Based on the energy per activity and the planned duty cycle, we then estimate the node lifetime based on the capacity of two AA batteries (capacity; see Table 1). To do this, we first calculate the complete energy per cycle by adding up the energy consumption for the sleep period of the duty-cycle and then divide the total capacity by this value.

4. Results

In this section, we present results for the different aspects of our study. The following points give an overview of subsequent explanations:

- Activity patterns and observed pattern variance showcase the activity patterns and their variance as a base for the modeling.
- Impact of networking aspects looks into energy consumption due to communications.
- Impact of design choices discusses how the mentioned power options affect the energy consumption.
- Node lifetime estimations presents the results of the different modeling approaches (i.e., datasheet vs. our method).

4.1. Individual Activity Patterns

Regarding the individual activities as described in Table 2, Figures 7–13 show the current consumption patterns for each activity, as single patterns and combined.

The leaf wetness sensor is probed five times before an average reading is calculated. This is depicted in Figure 7.

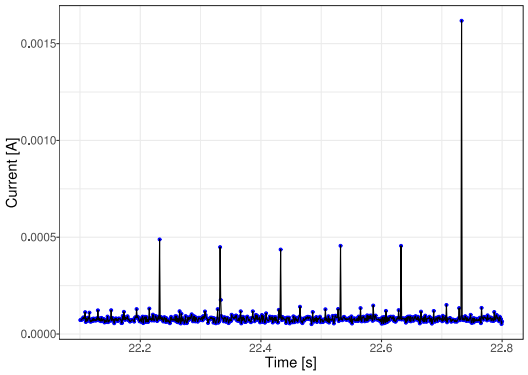


Figure 7. Energy consumption profile when reading leaf wetness sensor.

The patterns of the SHT20 sensor options are depicted in Figures 8 and 9. Each subplot shows a different configuration. The first one shows the reading of one sensor while the other two show two sensors with the difference that in Figure 9a, the sensors are powered via the battery directly (via the VDD pin), and in Figure 9b, they are powered via a port pin, allowing the nodes to switch off the sensors completely if not used. One can see the difference in the pattern shape for each configuration. Since the variants in Figure 8 and 9a are both powered via the battery, the difference in levels for the sensor activity shows an example of possible variation as all three cases use the same sensor settings and driver.

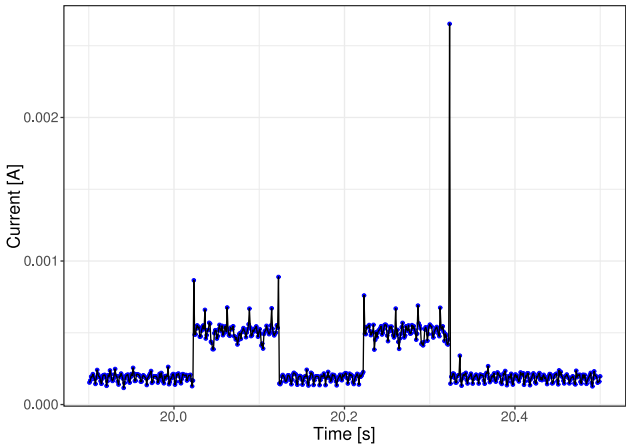
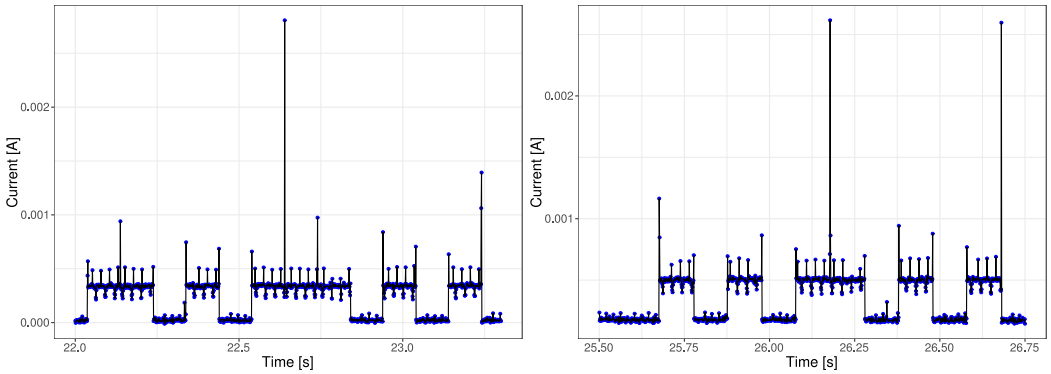


Figure 8. Single SHT sensor.



(**a**) Two SHT sensors (**b**) Two SHT sensors—GPIO-powered

Figure 9. Energy consumption profile when reading SHT sensors.

Finally, Figure 10 shows the example trace for reading all sensors consecutively, as it is performed under productive conditions for the node shown in Figure 1 without the networking. This covers the GPIO-powered SHT20 sensors as well as the battery and the leaf wetness sensor. To better highlight the different activity phases, we mark them in Figure 11.

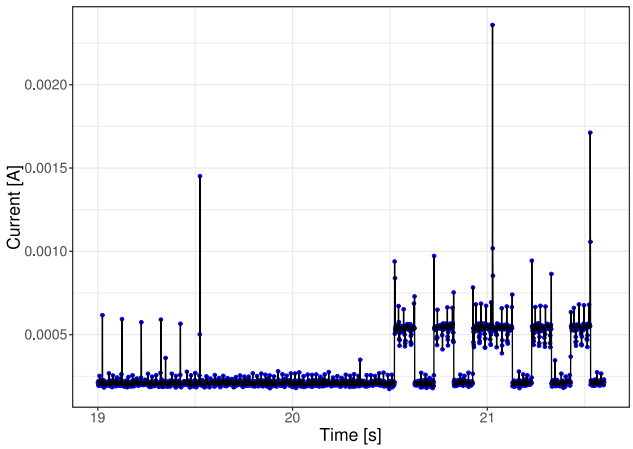


Figure 10. Energy consumption profile when reading a complete sensor set (leaf wetness, two SHT20 sensors) for Test 5.

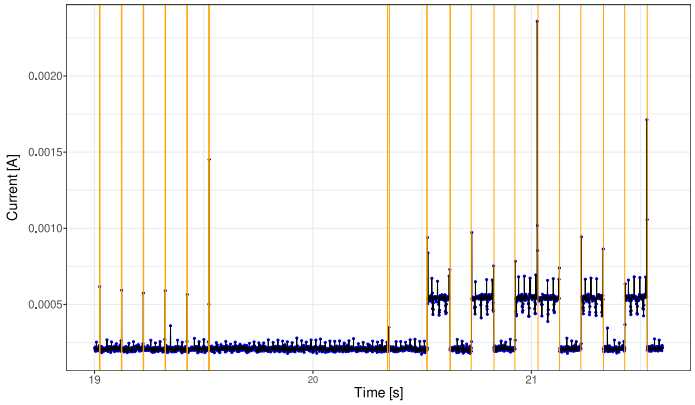


Figure 11. Current per pattern with highlighted single activities.

Networking also results in distinct activity patterns that are, however, more difficult to explain, especially for maintenance activities that are handled within the protocol stack of the nodes without direct access from the node application software. Figures 12 and 13 show sample activities related to networking.

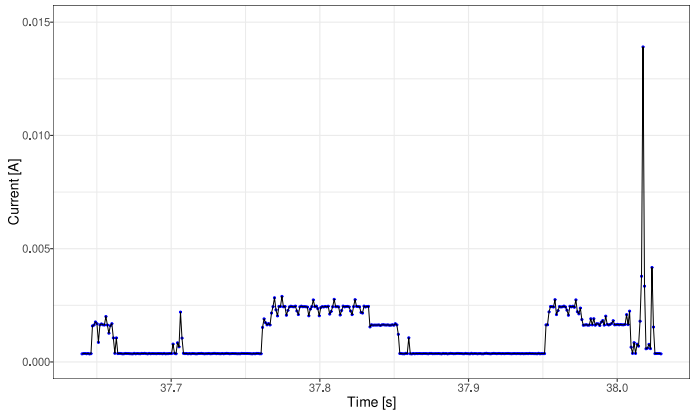
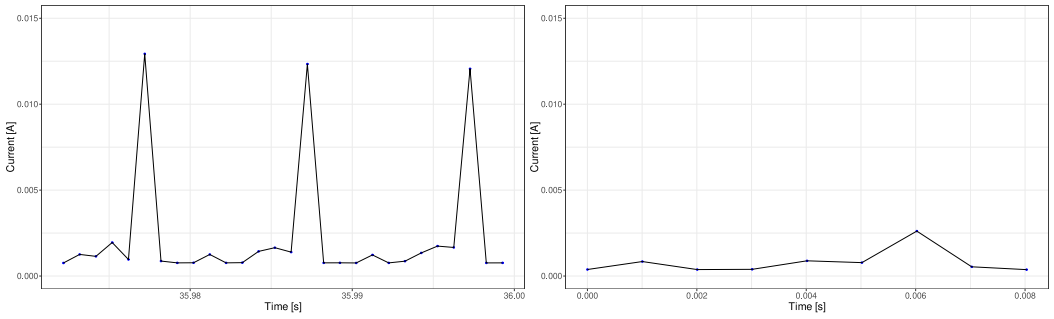


Figure 12. Data packet with sensing.



(**a**) Connection change (**b**) Maintenance packet

Figure 13. Energy consumption profile for different network activities.

To calculate the energy of the node, we obtain the timing of these phases as a base input as well as the individual current consumption per activity for the second method. Finally, we integrate the energy of each pattern in a step-wise manner and calculate an average value across the individual patterns. Table 3 shows the results.

Table 3. Overview of and comparison to state-of-the-art work.

Test	Activity		Avg Energy [µW s]	Duration [ms]
Test 1	LW		177.2	672.9
Test 2	1 SHT	VDD	711.2	275.6
Test 3	2 SHT	VDD	1167.1	1321.6
Test 4	2 SHT	GPIO	177.2	672.9
Test 5	2 SHT + LW	GPIO	1129.4	1129.4
Test 6	Bat		14.4	41.1
Test 7	Network	data transfer	104.0	16.1
		connect	149.3	34.1
		maint	71.7	24.1

4.2. Pattern Variance

Figure 14 shows a histogram of the observed energy per pattern for the complete sensor set (Test 6).

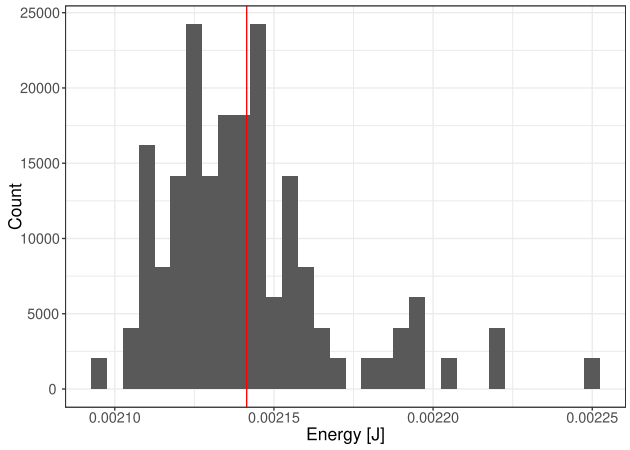


Figure 14. Histogram of energy consumption for test 5 and a bin-width of 5 µJ for a total of 100 individual patterns, highlighting the distribution and variance for this activity. The red line highlights the average energy consumption across all patterns, typically the only value considered for estimations.

This shows that the energy consumption is not constant even if the same actions are performed, confirming the results of [7]. It should be noted that the histogram does not show normally distributed values. While the majority of values could be approximated by a normal distribution, there are some outliers in this case: a few events that show an increased energy consumption and affect the mean value. Similar effects are visible for all other observed activities, including the networking patterns. For a modeling perspective, the normally distributed data would allow for variance-based sampling to be used for more realistic modeling.

Using a black-box approach, one can detect the behavior but not analyze its root cause further since details on contributing aspects are missing. However, this is not a problem since we accept that there is variance and use that as an input for our modeling idea.

4.3. Impact of Networking

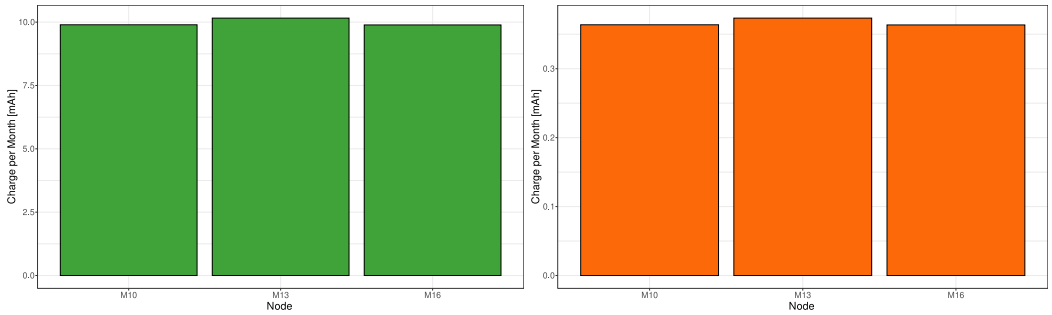
As mentioned earlier, the overhead due to networking is variable too and can be quite different depending on the position of the node in the network and the connection properties. When we study the metrics related to networking as described in Section 3.4, Table 4 shows the statistics for our chosen nodes over one month (744 h) of operation.

Table 4. Results of network analysis for selected nodes.

Node	Route Changes	Different Parents	Children	Interrupts	Uptime	Downtime
M.10	43	7	4	16	720:22:51	23:24:06
M.13	5	3	1	1	739:29:38	04:30:21
M.16	71	6	0	24	719:53:48	24:06:12

These values result in a significant increase in network overhead and different loads regarding the data to be handled by each node. Given that all nodes are connected 96% of the time during that month shows that the network works in general and there is no significant data loss. However, the cost to maintain this connectivity is different, and in other scenarios, the connectivity could be worse, resulting in more overhead.

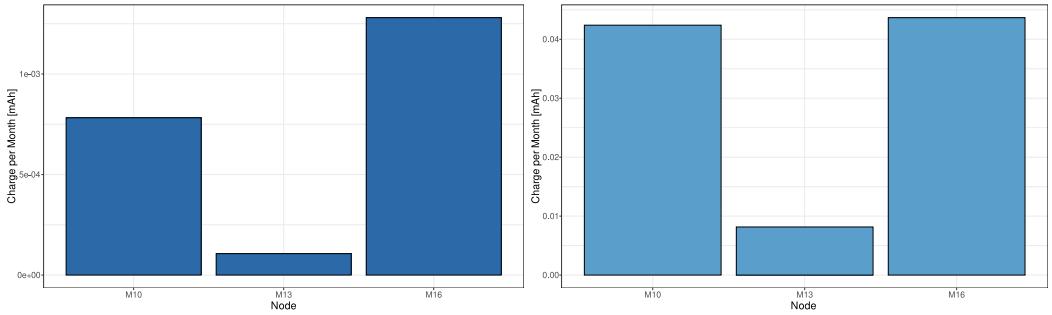
Node MK.13 requires only five route changes resulting in five connection events with one longer interruption resulting in frequent probing for new connection opportunities. MK.16 instead requires 19 times as many connection events and spends 24 h probing for new connections. MK.10 shows an intermediate behavior regarding route changes and downtime but has to handle some traffic from child nodes in addition. Using these base values, Figure 15 compares the resulting energy for each node.



(**a**) Maintenance (**b**) Data Transfers

Figure 15. Comparison of energy consumption due to connection maintenance and data transmission.

This shows that the different scenarios do have an impact on the overall result. However, the differences are not as significant in this case since the nodes are able to communicate most of the time, which results in a high maintenance load that dominates the other quantities. If we look at the effort to connect initially, the required probing to enable the re-connection in the case of a connection loss, as well as the effort to forward data, we see differences, as shown in Figures 16 and 17.



(**a**) Connection establishment (**b**) Probing

Figure 16. Comparison of energy consumption due to connection maintenance and data transmission, continued.

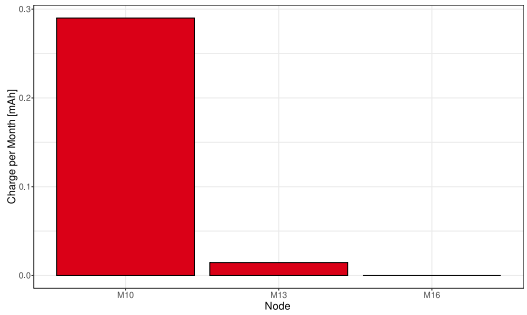


Figure 17. Comparison of energy consumption due to connection maintenance and data transmission, continued. Here: Forwarding.

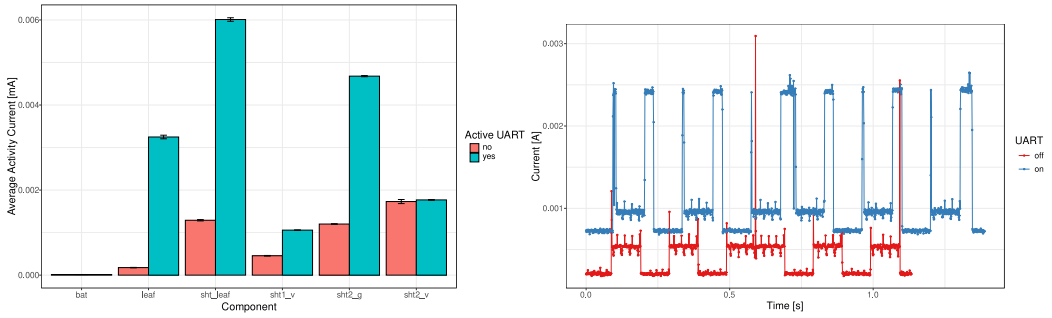
This shows that the network overhead due to role (router having children or not) and position in the network (stable or intermittent connectivity) does have an impact on the energy consumption, confirming earlier findings. When looking at the numbers, node M.13 uses less energy since it experiences stable connectivity and rarely forwards messages. In case of M.10, the overhead is related to a lack of network stability and forwarding data of four child nodes. Finally, M.16 spends the most energy trying to stay connected to the network.

One should keep in mind here that the nodes are still well connected and able to deliver most of their data to the gateway. If the situation was worse, with either less stability or more traffic due to an increased number of child nodes, the observed energy consumption would go up as well and, thus, have a bigger impact. Modeling these activities becomes complex but allows us to evaluate the impact of different effects that exceed the actual sampling and sending scheme of the node.

We expect similar results for further networking technologies. Any ad hoc or mesh-based network will exhibit a similar behavior. Star topologies with single-hop-only communication will lack the forwarding effort, but connection maintenance and re-connection efforts will happen there, too.

4.4. Impact of Design Choices

We included two design aspects in this study. One is the activation of different peripherals, such as a UART for debugging purposes, and the other one is the choice of whether the sensors are always on or can be switched off via a GPIO pin. Figure 18 shows the comparison for tests 1 through 6 accordingly.



(**a**) UART-enabled (green) vs. inactive (red) (**b**) Pattern comparison for two SHT sensors

Figure 18. Comparison of energy consumption due to activated peripherals.

When the UART is active, two things happen. The peripheral is initialized and ready to use. As a result, the node consumes about 400 µA more current as base load (cf. Figure 18), even if no UART communication takes place. In addition, each active communication adds additional activity (additional time and increased current draw) and, thus, prolongs the pattern and increases its energy requirement.

4.5. Comparison of Node Lifetime Estimations

Based on the previous results, we then build models according to the explanations in Section 3 to estimate the node lifetime. Table 5 shows the corresponding results for the node with one SHT sensor. This variant is used throughout the remainder of this paper and includes networking functionality as well. The sample node is set to a sensing and sending interval of 15 min, which matches the nodes of our sample scenario.

Table 5. Comparison of node lifetime estimation.

Case	Model Base	Lifetime [months]
1	datasheet	724.5
2	single-shot measurement	23.4
3	pattern-based	23.3

When comparing these values, one can see that the datasheet-based model is, as expected, overly optimistic, while the other two combinations show similar results. The reason for this is that the node has active networking but, otherwise, sleeps for the greater part of 15 min. In that case, the sleep current is dominant over all other variations. To study this effect further, we tuned the empirical variance-based model to explore the impact of the interval length and different sleep currents on the node's lifetime. Figure 19 shows the results.

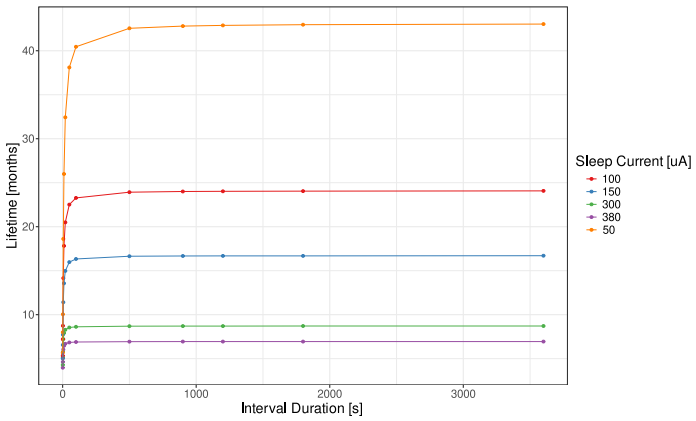


Figure 19. Impact of different sleep current consumption levels on estimated node lifetime.

The purple line corresponds to the actual measurement, while the orange one denotes the minimum sleep current achievable with the given software/hardware combination while still having the node follow its duty cycle without external triggers. In this case, it is obvious that sleep is dominant from interval lengths of about 2 min and that in that case, the actual sleep current limits the achievable node lifetime.

This corresponds with the results regarding active yet unused peripherals that may also increase the sleep current and, thus, further limit the node's lifetime. This is actually what happened to the measured values for the purple line. It maxes out at about 7–8 months of lifetime, while the nodes in the field are closer to the red line with approximately 24 months of lifetime, which was also the design requirement for this node. However, if intervals are below 2 min, the lifetime is less dependent on the sleep current and the variance shows an impact. This is the case for all situations that require frequent sampling.

Therefore, we analyze next how one can use the pattern-based model to evaluate different design considerations and the impact of dynamic variance. First, we look at an option to change the actual current required by the sensing pattern. Figure 20 shows a setup where we use the model to estimate lifetimes with reduced and increased energy consumption during the sensing phase, with the blue line denoting the baseline (the red one in Figure 19). This simulates the use of different sensors.

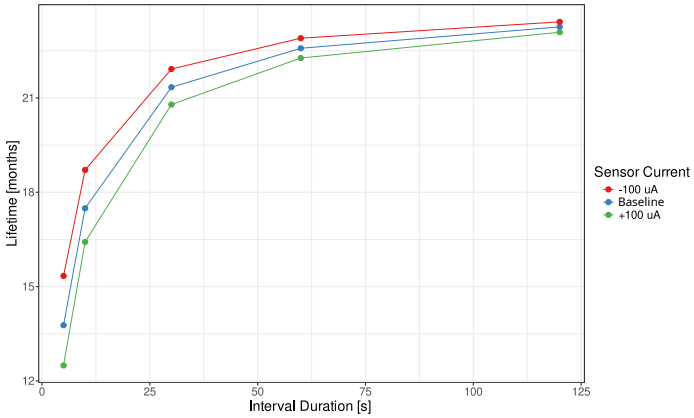


Figure 20. Impact of different current draws per sensor reading on estimated node lifetime.

The trend is similar to Figure 19; however, the impact of the different sensors becomes obvious. This clearly shows an optimization potential during the design phase of, e.g., vibration sensors with short sampling intervals.

Next, we look into the impact of different networking scenarios as described in Table 4. There, two factors are mentioned: the number of children a node has and, thus, the corresponding additional traffic it has to handle and the connectivity of the node with disconnects or changing parents. To better highlight the impact of both, we look at them separately. Figure 21 shows the impact of the connectivity. The setup assumes that a different number of disconnects happens per month and each event results in an average downtime of 1 h. The dark red line acts as a baseline without any interrupts, representing the ideal case, while the dark blue one shows the worst scenario under test.

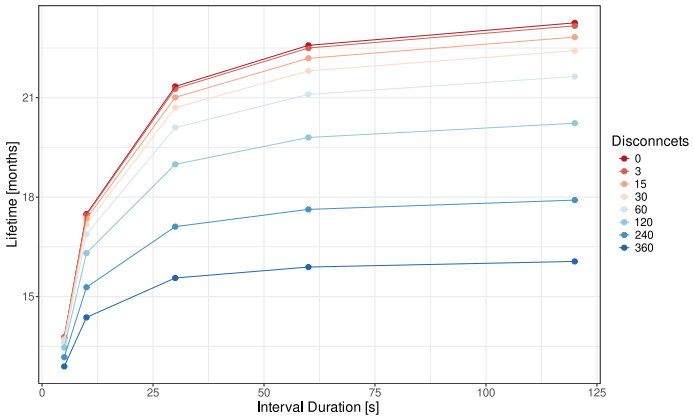


Figure 21. Impact of network connectivity on estimated node lifetime.

This shows a significant impact of the connectivity on the lifetime. The red to orange lines represent the nodes from our real-world scenario and confirm that their connectivity is quite good. During our field test, we, however, also experienced the dark blue scenario with a shortened lifetime for the corresponding node that is more busy connecting to the network than actually performing its sensing task. In such a case, the model clearly indicates the need to improve the network stability. This could be achieved by adding intermediate nodes or alternative paths to the actual deployment.

Finally, we look at the impact of child nodes. Figure 22 shows the corresponding results including a combination of disconnects and child nodes.

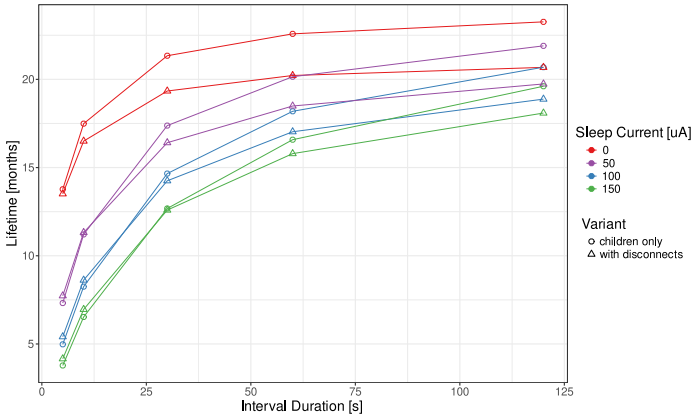


Figure 22. Impact of child nodes and disconnects on estimated node lifetime.

The number of child nodes decreases the lifetime of the parent node as expected due to the additional traffic (receiving a message from the child and forwarding it upstream) required per child. Depending on the given interval, this can lead to a reduction of about 10 months of lifetime and is, thus, a clear indication as to why routers often require a permanently powered operation. Regarding this aspect, our model can give new insights into how to build an energy-autonomous node for different scenarios. One other aspect visible in Figure 22 is the fact that for the higher number of children, the disconnects actually affect the lifetime positively at short intervals. This effect happens because the forwarding can only take place if the node has an upstream connection in the network.

5. Discussion

Initially, we proposed that the variance in the task-specific energy consumption of an IoT sensor node has the potential to provide more realistic energy consumption profiles and, thus, a more realistic lifetime estimation at the design phase. Our results regarding theoretical and one-shot measurements prove this hypothesis true. While datasheet values with ideal power saving modes are not able to capture the real energy consumption, empirical data give much better results. When looking at Table 5, one could, however, argue why a model with variance is needed if a model based on a single-shot measurement gives similar results.

To answer this question, we performed additional analyses also based on our real-world scenario. These clearly showed different lifetimes for the given nodes that result from dynamic changes in the environment. Especially, network dynamics such as additional child nodes and experienced intermittent connectivity issues can lead to a higher energy consumption and, thus, results that differ from the estimations. This is nothing new but has only been considered as part of network simulations with datasheet or single-shot measurements. In addition, these simulations are often used to prove and verify the functionality of network protocols but do not reflect real-world scenarios or node setups. Our proposed method enables researchers to take the variance into account and estimate different scenarios based on real hardware. As a result, tuning the model to given real nodes allows for predictions that match lifetimes experienced with similar nodes in the field.

Evaluating the impact of the pattern variance enables, in addition, two more aspects: to study how certain design choices affect the energy consumption and to review whether a given setup has the expected energy consumption. Since the model can show when the sleep current is dominant, one can use it to estimate what benefit could result from a further reduction or what impact an increase, e.g., due to additional or more energy-hungry sensors, may have. If the sleep current is not dominant, further analyses show the impact of network dynamics and sensor variability. Such insights are valuable when designing battery-powered systems with a desired lifetime under a given energy budget. They allow one to estimate potential lifetime gains and identify crucial areas for optimization. For example, if a node has a low sleep current, reducing this further might not be feasible with

a given effort. Instead, one could think about changing the sensing or sending schemes for the node, which might be achievable with less effort.

These results are promising, but the black-box approach is limited in identifying specific sources of higher energy consumption. This could, however, be mitigated by using additional measurement points. Such an approach should show the same performance as our current version once the detailed patterns are combined. We plan to explore this in the future.

To further enhance the study options of our modeling approach, we plan to (a) integrate models for energy harvesting and (b) integrate our approach into a simulation environment. The addition of energy harvesting options based on empirical data will enable us to also consider the recharging of the node in lifetime estimations and open up the design space of harvester dimensioning in combination with node optimization. The full potential of this will become available if we integrate our model into a simulation environment, which will allow us to study more dynamic node behaviors, such as adaptation strategies of the node, e.g., to sample less frequently or to send only measured values that have changed. In terms of energy harvesters, we plan to also include estimations based on the characteristics of the harvesting source, e.g., changes/availability of solar radiation. This focuses on the prominent energy source in the smart agriculture setting as a first step. Other energy sources such as vibrations could also be added to cover different applications.

Despite being focused on smart agriculture as the only application area, our method is not limited to this application area. Any other scenario can benefit, too, if the steps described in Section 3 are followed. In essence, if a node based on a different MCU or sensor is to be estimated, one needs to take the corresponding measurements and can use these to tune the estimation model. This also applies to application characteristics such as sending intervals and networking options (e.g., node position and communication technology). With this information, the modeling approach can be tuned to the scenario in question. This also includes adding more sophisticated processing tasks on the MCU as an activity pattern to cover the execution of machine learning at the edge. We plan to explore how this can help to understand the requirements for sensor nodes with high sample rates in industrial environments as a future case study.

6. Conclusions and Future Work

In this paper, we proposed to use empirical energy consumption patterns of a node's activity to model the energy consumption of IoT sensor nodes and achieve realistic lifetime estimations. Our base idea is that the energy consumption is nothing static and can vary depending on different factors. We, therefore, study under which circumstance this variance matters in node lifetime estimation and how our model can be used to evaluate different design aspects. The results show a significant difference to datasheet-based approaches and are able to capture the lifetime of different real-world nodes realistically. By tuning the model and exploring the impact of different setting, e.g., for networking, we are able to study different aspects of the design space for a given node and, thus, go beyond the question: how long will my system survive? We are able to answer that and, by looking at different configurations, extend it further.

In the future, we plan to add an energy-harvesting module to the model and integrate it into a simulation environment in order to enable studies of different scenarios following the given approach. In addition, we will explore how more detailed measurement points would affect the estimations as well as the runtime performance.

Author Contributions: Conceptualization, S.K. and T.H.; methodology, S.K.; software, S.K.; validation, S.K., T.H. and H.T.; formal analysis, S.K.; investigation, S.K. and T.H.; data curation, S.K.; writing—original draft preparation, S.K. and T.H.; writing—review and editing, S.K., T.H., M.O. and H.T.; visualization, S.K. and T.H.; supervision, M.O.; project administration, S.K.; funding acquisition, T.H. and M.O. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially funded by the Swedish Knowledge Foundation in the research profile Next generation Industrial IoT (NIIT). In addition, this work was partially funded by the German Federal Ministry of Education and Research under the grand reference number 16ME0703 as part of the HoLoDEC project.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Alvarado-Alcon, F.J.; Asorey-Cacheda, R.; Garcia-Sanchez, A.J.; Garcia-Haro, J. Carbon Footprint vs Energy Optimization in IoT Network Deployments. *IEEE Access* **2022**, *10*, 111297–111309. [CrossRef]
2. Maistriaux, P.; Pirson, T.; Schramme, M.; Louveaux, J.; Bol, D. Modeling the Carbon Footprint of Battery-Powered IoT Sensor Nodes for Environmental-Monitoring Applications. In Proceedings of the 12th International Conference on the Internet of Things, Delft, The Netherlands, 7–10 November 2022; pp. 9–16.
3. Girban, G.; Popa, M. A glance on WSN lifetime and relevant factors for energy consumption. In Proceedings of the 2010 International Joint Conference on Computational Cybernetics and Technical Informatics, Timisoara, Romania, 27–29 May 2010; pp. 523–528.
4. Gelenbe, E.; Siavvas, M. Minimizing energy and computation in long-running software. *Appl. Sci.* **2021**, *11*, 1169. [CrossRef]
5. Kunkel, J.; Dolz, M.F. Understanding hardware and software metrics with respect to power consumption. *Sustain. Comput. Inform. Syst.* **2018**, *17*, 43–54. [CrossRef]
6. Ournani, Z.; Belgaid, M.C.; Rouvoy, R.; Rust, P.; Penhoat, J.; Seinturier, L. Taming energy consumption variations in systems benchmarking. In Proceedings of the ACM/SPEC International Conference on Performance Engineering, Edmonton, AB, Canada, 25–30 April 2020; pp. 36–47.
7. Dietrich, T.; Krug, S.; Zimmermann, A. An empirical study on generic multicopter energy consumption profiles. In Proceedings of the Annual IEEE International Systems Conference (SysCon), Montreal, QC, Canada, 24–27 April 2017; pp. 1–6.
8. Richa, M.; Prévotet, J.C.; Dardaillon, M.; Mroué, M.; Samhat, A.E. High-level power estimation techniques in embedded systems hardware: An overview. *J. Supercomput.* **2023**, *79*, 3771–3790. [CrossRef]
9. Kumar, T.P.; Krishna, P.V. A survey of energy modeling and efficiency techniques of sensors for IoT systems. *Int. J. Sens. Wirel. Commun. Control* **2021**, *11*, 271–283. [CrossRef]
10. Rodrigues, L.M.; Bitencourt, N.L.; Rech, L.; Montez, C.; Moraes, R. An analytical model to estimate the state of charge and lifetime for batteries with energy harvesting capabilities. *Int. J. Energy Res.* **2020**, *44*, 5243–5258. [CrossRef]
11. Rodrigues, L.M.; Montez, C.; Budke, G.; Vasques, F.; Portugal, P. Estimating the lifetime of wireless sensor network nodes through the use of embedded analytical battery models. *J. Sens. Actuator Netw.* **2017**, *6*, 8. [CrossRef]
12. Morin, É.; Maman, M.; Guizzetti, R.; Duda, A. Comparison of the Device Lifetime in Wireless Networks for the Internet of Things. *IEEE Access* **2017**, *5*, 7097–7114. [CrossRef]
13. Miranda, F.A.M.; Cardieri, P. The impact of multiple power levels on the lifetime of Wireless Sensor Networks. In Proceedings of the 2016 IEEE International Symposium on Consumer Electronics (ISCE), Sao Paulo, Brazil, 28–30 September 2016; pp. 103–104.
14. Schaarschmidt, M.; Uelschen, M.; Pulvermüller, E. Hunting Energy Bugs in Embedded Systems: A Software-Model-In-the-Loop Approach. *Electronics* **2022**, *11*, 1937. [CrossRef]
15. Herzog, B.; Reif, S.; Hügel, F.; Schröder-Preikschat, W.; Hönig, T. Bears: Building Energy-Aware Reconfigurable Systems. In Proceedings of the 2022 XII Brazilian Symposium on Computing Systems Engineering (SBESC), Fortaleza, CE, Brazil, 21–24 November 2022; pp. 1–8.
16. Büsching, F.; Garlichs, K.; Kulau, U.; Rottmann, S.; Wolf, L. Design Considerations of Mission-Oriented Sensor Node Architectures. In *Mission-Oriented Sensor Networks and Systems: Art and Science*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 11–47.
17. Brini, O.; Deslandes, D.; Nabki, F. A system-level methodology for the design of reliable low-power wireless sensor networks. *Sensors* **2019**, *19*, 1800. [CrossRef]
18. Friesel, D.; Buschhoff, M.; Spinczyk, O. Parameter-aware energy models for embedded-system peripherals. In Proceedings of the 2018 IEEE 13th International Symposium on Industrial Embedded Systems (SIES), Graz, Austria, 6–8 June 2018; pp. 1–4.
19. Anderson, M.E. Technical trade-offs of IoT platforms. In Proceedings of the Autonomous Systems: Sensors, Vehicles, Security, and the Internet of Everything. International Society for Optics and Photonics, Orlando, FL, USA, 15–19 April 2018; Volume 10643, p. 1064316.
20. Martinez, B.; Montón, M.; Vilajosana, I.; Prades, J.D. The power of models: Modeling power consumption for IoT devices. *IEEE Sens. J.* **2015**, *15*, 5777–5789. [CrossRef]
21. Maudet, G.; Maillé, P.; Toutain, L.; Batton-Hubert, M. Energy Efficient Message Scheduling with Redundancy Control for Massive IoT Monitoring. In Proceedings of the Wireless Communications and Networking Conference (WCNC), Glasgow, UK, 26–29 March 2023; pp. 1–6.
22. Loh, F.; Raffeck, S.; Geißler, S.; Hoßfeld, T. Generic Model to Quantify Energy Consumption for Different LoRaWAN Channel Access Methods. In Proceedings of the 18th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Thessaloniki, Greece, 10–12 October 2022; pp. 290–295.

23. Krug, S.; Miethe, S.; Hutschenreuther, T. Comparing BLE and NB-IoT as Communication Options for Smart Viticulture IoT Applications. In Proceedings of the 2021 IEEE Sensors Applications Symposium (SAS), Sundsvall, Sweden, 23-25 August 2021; pp. 1–6.
24. Kaur, G.; Chanak, P.; Bhattacharya, M. Energy-efficient intelligent routing scheme for IoT-enabled WSNs. *IEEE Internet Things J.* **2021**, *8*, 11440–11449. [CrossRef]
25. Krug, S.; Bader, S.; Oelmann, B.; O'Nils, M. Suitability of Communication Technologies for Harvester-Powered IoT-Nodes. In Proceedings of the 15th IEEE International Workshop on Factory Communication Systems, Sundsvall, Sweden, 27–29 May 2019.
26. Krug, S.; O'Nils, M. Modeling of IoT Edge Communication Technologies Enabling Data Transmission Cost Estimation. *IEEE Access* **2019**, *7*, 58654–58675. [CrossRef]
27. Krug, S.; Shallari, I.; O'Nils, M. A Case Study on Energy Overhead of Different IoT Network Stacks. In Proceedings of the 5th World Forum on Internet of Things (WF-IoT), Limerick, Ireland, 15–18 April 2019.
28. Ayers, H.; Crews, P.; Teo, H.; McAvity, C.; Levy, A.; Levis, P. Design Considerations for Low Power Internet Protocols. *arXiv* **2018**, arXiv:1806.10751.
29. Krug, S.; Schreiber, A.; Rink, M. Poster: Beyond Static Sending Intervals for Sensor Node Energy Estimation. In Proceedings of the 12th Workshop on Challenged Networks (CHANTS), Snowbird, UT, USA, 20 October 2017.
30. Kulau, U.; Müller, S.; Schildt, S.; Martens, A.; Büsching, F.; Wolf, L. Energy efficiency impact of transient node failures when using RPL. In Proceedings of the 2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), Macau, China, 12–15 June 2017; pp. 1–6.
31. Domingo Prieto, M.; Martínez, B.; Monton, M.; Guillen, I.V.; Guillen, X.V.; Moreno, J.A. Balancing power consumption in IoT devices by using variable packet size. In Proceedings of the 8th International Conference on Complex, Intelligent and Software Intensive Systems (CISIS), Birmingham, UK, 2–4 July 2014; pp. 170–176.
32. Sanchez Leal, I.; Saqib, E.; Shallari, I.; Jantsch, A.; Krug, S.; O'Nils, M. Waist Tightening of CNNs: A Case study on Tiny YOLOv3 for Distributed IoT Implementations. In Proceedings of the Cyber-Physical Systems and Internet of Things Week 2023, San Antonio, TX, USA, 9–12 May 2023; ACM: New York, NY, USA, 2023; pp. 241–246.
33. Marantos, C.; Maidonis, N.; Soudris, D. Designing Application Analysis Tools for Cross-Device Energy Consumption Estimation. In Proceedings of the 11th International Conference on Modern Circuits and Systems Technologies (MOCAST), Bremen, Germany, 8–10 June 2022; pp. 1–4.
34. Paganelli, A.I.; Sarmento, A.; Branco, A.; Endler, M.; Nascimento, N.; Alencar, P.; Cowan, D. Assessing Energy Consumption in Data Acquisition from Smart Wearable Sensors in IoT-Based Health Applications. In Proceedings of the 2022 IEEE International Conference on Big Data (Big Data), Osaka, Japan, 17–20 December 2022; pp. 2882–2885.
35. Shallari, I.; Leal, I.S.; Krug, S.; Jantsch, A.; O'Nils, M. Design space exploration for an IoT node: Trade-offs in processing and communication. *IEEE Access* **2021**, *9*, 65078–65090. [CrossRef]
36. Kulau, U.; van Balen, J.; Schildt, S.; Büsching, F.; Wolf, L. Dynamic Sample Rate Adaptation for Long-Term IoT Sensing Applications. In Proceedings of the 3rd World Forum on Internet of Things (WF-IoT), Reston, VA, USA, 12–14 December 2016.
37. Vilajosana, X.; Wang, Q.; Chraim, F.; Watteyne, T.; Chang, T.; Pister, K. A Realistic Energy Consumption Model for TSCH Networks. *IEEE Sens. J.* **2014**, *14*, 482–489. [CrossRef]
38. Maddikunta, P.K.R.; Srivastava, G.; Gadekallu, T.R.; Deepa, N.; Boopathy, P. Predictive model for battery life in IoT networks. *IET Intell. Transp. Syst.* **2020**, *14*, 1388–1395. [CrossRef]
39. Sliper, S.T.; Wang, W.; Nikoleris, N.; Weddell, A.S.; Merrett, G.V. Fused: Closed-loop performance and energy simulation of embedded systems. In Proceedings of the 2020 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), Boston, MA, USA, 23–25 August 2020; pp. 263–272.
40. Lages, D.; Borba, E.; Tavares, E.; Balieiro, A.; Souza, E. A CPN-based model for assessing energy consumption of IoT networks. *J. Supercomput.* **2023**, *79*, 12978–13000. [CrossRef]
41. Kanso, H.; Noureddine, A.; Exposito, E. Automated power modeling of computing devices: Implementation and use case for Raspberry Pis. *Sustain. Comput. Inform. Syst.* **2023**, *37*, 100837. [CrossRef]
42. Friesel, D.; Spinczyk, O. Regression model trees: Compact energy models for complex IoT devices. In Proceedings of the Workshop on Benchmarking Cyber-Physical Systems and Internet of Things (CPS-IoTBench), Milan, Italy, 3–6 May 2022; pp. 1–6.
43. Sieber, A.; Nolte, J. Datasheet vs. Real world: A look on sensor node energy consumption. In Proceedings of the International Conference on Green Computing and Communications (GreenCom), Besancon, France, 20–23 November 2012; pp. 641–643.
44. Feeney, L.M.; Hartung, R.; Rohner, C.; Kulau, U.; Wolf, L.; Gunningberg, P. Towards realistic lifetime estimation in battery-powered IoT devices. In Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems, Delft, The Netherlands, 6–8 November 2017; ACM: New York, NY, USA, 2017; p. 67.
45. Friesel, D.; Kaiser, L.; Spinczyk, O. Automatic energy model generation with MSP430 EnergyTrace. In Proceedings of the Workshop on Benchmarking Cyber-Physical Systems and Internet of Things, Nashville, TN, USA, 18 May 2021; pp. 26–31.
46. Mayer, P.; Magno, M.; Brunner, T.; Benini, L. LoRa vs. LoRa: In-Field Evaluation and Comparison For Long-Lifetime Sensor Nodes. In Proceedings of the 2019 IEEE 8th International Workshop on Advances in Sensors and Interfaces (IWASI), Otranto, Italy, 13–14 June 2019; pp. 307–311.
47. Gomez, K.; Boru, D.; Riggio, R.; Rasheed, T.; Miorandi, D.; Granelli, F. Measurement-based modelling of power consumption at wireless access network gateways. *Comput. Netw.* **2012**, *56*, 2506–2521. [CrossRef]

48. Dietrich, T.; Krug, S.; Hotz, T.; Zimmermann, A. Towards Energy Consumption Prediction with Safety Margins for Multicopter Systems. In Proceedings of the 11th International Conference on Performance Evaluation Methodologies and Tools (ValueTools), Venice, Italy, 5–7 December 2017; pp. 227–228.
49. Dasgupta, S.; Dutta, P. A Study on Collapse Time Analysis of Behaviorally Changing Nodes in Static Wireless Sensor Network. In *Intelligent Computing Paradigm: Recent Trends*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 11–20.
50. Aslanpour, M.S.; Toosi, A.N.; Gaire, R.; Cheema, M.A. WattEdge: A holistic approach for empirical energy measurements in edge computing. In Proceedings of the 19th International Conference of Service-Oriented Computing (ICSOC), Online, 22–25 November 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 531–547.
51. Boano, C.A.; Duquennoy, S.; Förster, A.; Gnawali, O.; Jacob, R.; Kim, H.S.; Landsiedel, O.; Marfievici, R.; Mottola, L.; Picco, G.P. IoTBench: Towards a Benchmark for Low-power Wireless Networking. In Proceedings of the 1st Workshop on Benchmarking Cyber-Physical Networks and Systems (CPSBench 2018), Porto, Portugal, 10–13 April 2018.
52. Bramas, Q.; Dron, W.; Fadhl, M.B.; Hachicha, K.; Garda, P.; Tixeuil, S. WiSeBat: Accurate energy benchmarking of wireless sensor networks. In Proceedings of the 2015 Forum on Specification and Design Languages (FDL), Barcelona, Spain, 14–16 September 2015; pp. 1–8.
53. Wang, W.; Sobral, V.A.L.; Billah, M.F.R.M.; Saoda, N.; Nasir, N.; Campbell, B. Low power but high energy: The looming costs of billions of smart devices. *ACM SIGENERGY Energy Inform. Rev.* **2023**, *3*, 10–14. [CrossRef]
54. Badri, S.; Saini, M.; Goel, N. Design of Energy Harvesting based Hardware for IoT Applications. *arXiv* **2023**, arXiv:2306.12019.
55. Šabović, A. Energy-Aware Design of Battery-Less IoT Devices. Ph.D. Thesis, University of Antwerp, Antwerpen, Belgium, 2024.
56. Tundo, A.; Mobilio, M.; Ilager, S.; Brandić, I.; Bartocci, E.; Mariani, L. An energy-aware approach to design self-adaptive ai-based applications on the edge. In Proceedings of the 2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE), Luxembourg, 11–15 September 2023; pp. 281–293.
57. Anusha, P.; Chouhan, N.; Chandhok, G.A.; Sugumaran, D.; Aswal, U.; Suganya, A. Empowering IoT Devices with Energy-Efficient AI and Machine Learning. In Proceedings of the 2024 7th International Conference on Circuit Power and Computing Technologies (ICCPCT), Kollam, India, 8–9 August 2024; Volume 1, pp. 720–725.
58. Krug, S.; Goetze, M.; Schneider, S.; Hutschenreuther, T. A Modular Platform to Build Task-Specific IoT Network Solutions for Agriculture and Forestry. In Proceedings of the International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor), Pisa, Italy, 6–8 November 2023.
59. Grass, R.; Boedeker, H.; Hofmann, M.; Schieck, M.; Krug, S.; Hutschenreuther, T.; Mollenhauer, H. Plot-specific drought stress simulation in vineyards using a microclimatic monitoring system in combination with a radiation and water balance model. In Proceedings of the International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor), Pisa, Italy, 6–8 November 2023.
60. Nordic Semiconductor ASA. *nRF52832 Product Specification*; Rev. 1.4; Nordic Semiconductor ASA: Trondheim, Norway, 2017.
61. Sensirion. *Datasheet SHT20—Humidity and Temperature Sensor IC*; Sensirion: Staefa, Switzerland, 2022.
62. Texas Instruments. *TCA9546A Low Voltage 4-Channel I2C and SMBus Switch with Reset Function*; Texas Instruments: Dallas, TX, USA, 2019.