



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich



Institut für
Technische Informatik und
Kommunikationsnetze

Is low-power wireless networking a reproducible science?

Semester Thesis

Antonios Koskinas

Scholar of Onassis Foundation

akoskina@student.ethz.ch

Computer Engineering and Networks Laboratory
Department of Information Technology and Electrical Engineering
ETH Zürich

Supervisors:

Romain Jacob

Prof. Dr. Lothar Thiele

January 10, 2019

Acknowledgements

I would like to thank my supervisor, Romain Jacob, for his excellent guidance and support during this semester project. His insights and contributions have been vital and several of the presented results have been a joint effort.

In addition, I would like to thank Prof. Dr. Lothar Thiele and the Computer Engineering and Networks Laboratory for allowing me to realize this project.

Furthermore, I would like to thank Onassis Foundation for its financial support throughout my master's degree studies.

Abstract

Recently, a noticeable number of attempts to reproduce experimental results across many different scientific fields were unsuccessful. In the field of low-power wireless networking, the dynamic behavior of the environment causes an inherited variability in the performance of wireless networking protocols, placing under consideration their reproducibility.

In this semester project, a methodology for evaluating the performance of low-power wireless networking protocols targeted for periodic, non real-time data collection applications is proposed.

In addition, we present a case study, where we apply the proposed methodology to evaluate the performance of a state-of-the-art low-power protocol, Crystal.

Finally, based on this methodology, we propose definitions for repeatability, replicability and reproducibility in the context of low-power wireless networking.

Contents

Acknowledgements	i
Abstract	ii
1 Introduction	1
1.1 ACM definitions	1
1.2 Project's objectives	2
1.3 Challenges	3
2 Methodology for protocol evaluation	4
2.1 Inputs of the methodology	6
2.2 Step 1: Definition of output variables	6
2.2.1 Energy efficiency	6
2.2.2 Reliability	7
2.3 Frequency of data collection	7
2.4 Step 2: Time duration of a single test	7
2.4.1 Protocol period vs application period	8
2.4.2 Statistical analysis and number of samples	9
2.5 Step 3: Data processing	10
2.6 Step 4: Number of tests	11
2.7 Step 5: Results presentation	12
3 Case study: Crystal protocol	13
3.1 Background	13
3.2 Description of Crystal	14
3.3 Application of methodology on Crystal	15
3.3.1 Inputs	15
3.3.2 Step 1: Definition of output variables	15

CONTENTS	iv
3.3.3 Step 2: Time duration of a single test	16
3.3.4 Step 3: Data processing	17
3.3.5 Step 4: Number of tests	17
3.3.6 Step 5: Results presentation	19
4 Proposal of definitions	20
4.1 Repeatability definition	20
4.2 Replicability definition	21
4.3 Reproducibility definition	22
5 Conclusion and Future Work	23
Bibliography	24

Introduction

One of the most fundamental assumptions in science is the reproducibility of an experimental result. As Karl Popper stated already in 1959, “*non-reproducible single occurrences are of no significance to science*” [11]. Therefore, the concept of reproducibility is one of the most important issues for researchers, authors and reviewers, since it is vital in the conduct and validation of experimental science. Oxford English Dictionary [13] defines reproducibility as “the extent to which consistent results are obtained when produced repeatedly”. As it can be seen, this definition is rather broad and general, as reproducibility is a necessary merit across all scientific disciplines. It should be noted that repeatability and replicability are two additional concepts that are related to reproducibility. There are, however, substantial differences in their meaning, as we will explain.

The motivation to examine these concepts came from various recent studies showing that contributions in many fields (e.g biology, economics, computer science) could not be reproduced. One such study is [7], where the authors identify data unavailability as one of the main reasons for not succeeding in reproducing the results. In addition, they also claim that the observed discrepancies were caused mostly due to incomplete data annotation or specification of data processing and analysis. As a result, they recommend the adoption of generally accepted strict publication rules. Such rules would enforce public data availability and would encourage the explicit description of the methods that are used for data processing.

1.1 ACM definitions

In Computer Science, the need for reproducible experiments motivated one of the main publishers in the field, namely ACM, to provide various initiatives to support reproducibility. One such action was to establish a committee to propose “Best Practices Guideline for Data, Software, and Reproducibility in Publication”, which is, in essence, a list of recommendations that could potentially

improve the reproducibility of scientific work, if followed. These recommendations were admittedly not prescriptive enough, due to the substantive differences among the various sub-disciplines within the field of Computer Science [3]. Moreover, ACM created the Digital Library-Curation Platform Integrations in order to encourage authors to submit a snapshot of their software and data sets for permanent archiving along with their papers [1]. Finally, ACM introduced a novel system of Artifact Review and Badging, where terminology and guidelines for reviewing research artifacts are proposed. The aim is to provide some uniformity in the labeling of successfully reviewed papers across publications [2]. For completeness, the proposed definitions are presented here. Based on them, individual researchers would be able to report whether they could validate the experimental results of other researches, and papers would be awarded with the respective badge. The main characteristics of each term are illustrated in figure 1.1.

Repeatability: An experiment is repeatable if the measurements can be obtained with stated precision by the same team using the same measurement procedure and the same measuring system, under the same operating conditions, in the same location on multiple trials.

Replicability: An experiment is replicable if the measurements can be obtained with stated precision by a different team using the same measurement procedure and the same measuring system, under the same operating conditions, in the same or a different location on multiple trials.

Reproducibility: An experiment is reproducible if the measurements can be obtained with stated precision by a different team using a different measuring system, in a different location on multiple trials.

<i>Term</i>	<i>Characteristics</i>		
<i>Repeatability</i>	<i>same</i> result	<i>same</i> team	<i>same</i> experimental setup
<i>Replicability</i>	<i>same</i> result	<i>different</i> team	<i>different</i> experimental setup
<i>Reproducibility</i>	<i>same</i> result	<i>different</i> team	<i>different</i> experimental setup

Figure 1.1: Synopsis of the ACM definitions

1.2 Project's objectives

As it can be seen from the above terminology, these definitions are rather abstract. This project focuses on low-power wireless networking and one of its

objectives is to specify and adapt the general definitions of repeatability, replicability and reproducibility in this context. In addition, although reproducibility is desirable in experimental science, practical questions regarding the test duration or the number of the tests that are required to allow the researcher to extract meaningful conclusions do not have a clear answer. To illustrate this, the following example is presented. A new low-power wireless networking protocol has been designed and given a well-defined experimental setup and an application scenario, the performance of that protocol needs to be assessed. How long should each test be? How many tests need to run, in order to obtain statistically meaningful results? The typical answer to such questions is as vague as: run long tests for many times.

It is commonly accepted across all researches in the field of wireless communications that it is natural to expect some variation in the performance of a wireless protocol, due to the hardly controllable behavior of the environment. Thus, it makes sense to define that a protocol is reproducible in a statistical way. Therefore, we propose a methodology that is composed of a set of steps and it is based on statistics in order to assess the repeatability, replicability and reproducibility of a protocol. The number of samples that need to be collected per test as well as the required number of the tests that will be executed are determined based on statistical sound arguments.

1.3 Challenges

Developing such methodology is demanding, since it has to be widely applicable. More specifically, the methodology should accommodate a large variety of application scenarios and also different traffic profiles, as the number of packets generated across the network can vary. Moreover, no other similar methodology has been found in the low-power networking field and hence it is a novel approach. However, the most challenging factor is the uncontrollable and unpredictable variations of the wireless environment. This variability in the environment can have an important impact on the performance of the protocol. Therefore, the key question is: How much variability should be tolerated to characterize a protocol reproducible, given the intrinsic variability of the wireless environment?

In chapter 2 the proposed methodology is explained in detail. In chapter 3, a case study of a state-of-the-art low-power networking protocol, Crystal, is analyzed using the proposed methodology. In chapter 4, we propose definitions for repeatability, replicability and reproducibility based on the described methodology. In chapter 5 comes the conclusion along with some extensions for future work.

Methodology for protocol evaluation

As presented in the first chapter, one of the problems in constructing a methodology to evaluate a given low-power wireless networking protocol is the large variety of possible application scenarios. In an attempt to narrow that broad spectrum, the proposed methodology will aim to evaluate protocols that cater the application scenario of non real-time periodic data collection. In such application scenario, a many-to-one traffic is generated periodically from source nodes to the sink node. Since it is non real-time, the packet delay is not a concern. The overall methodology is illustrated in figure 2.1.

At this point some important comments are highlighted. To begin with, it should be noted that the aim of the proposed methodology is not to define a strict set of metrics, based on which the repeatability, replicability and reproducibility should be evaluated. On the contrary, the proposed methodology is independent of the used metrics. This is true because the goal is not to compare across different protocols but rather assess if one protocol is reproducible, and hence examine the same set of metrics. The choice of the metrics, though, can affect the final result, as it could be that some metrics have more stable values than others.

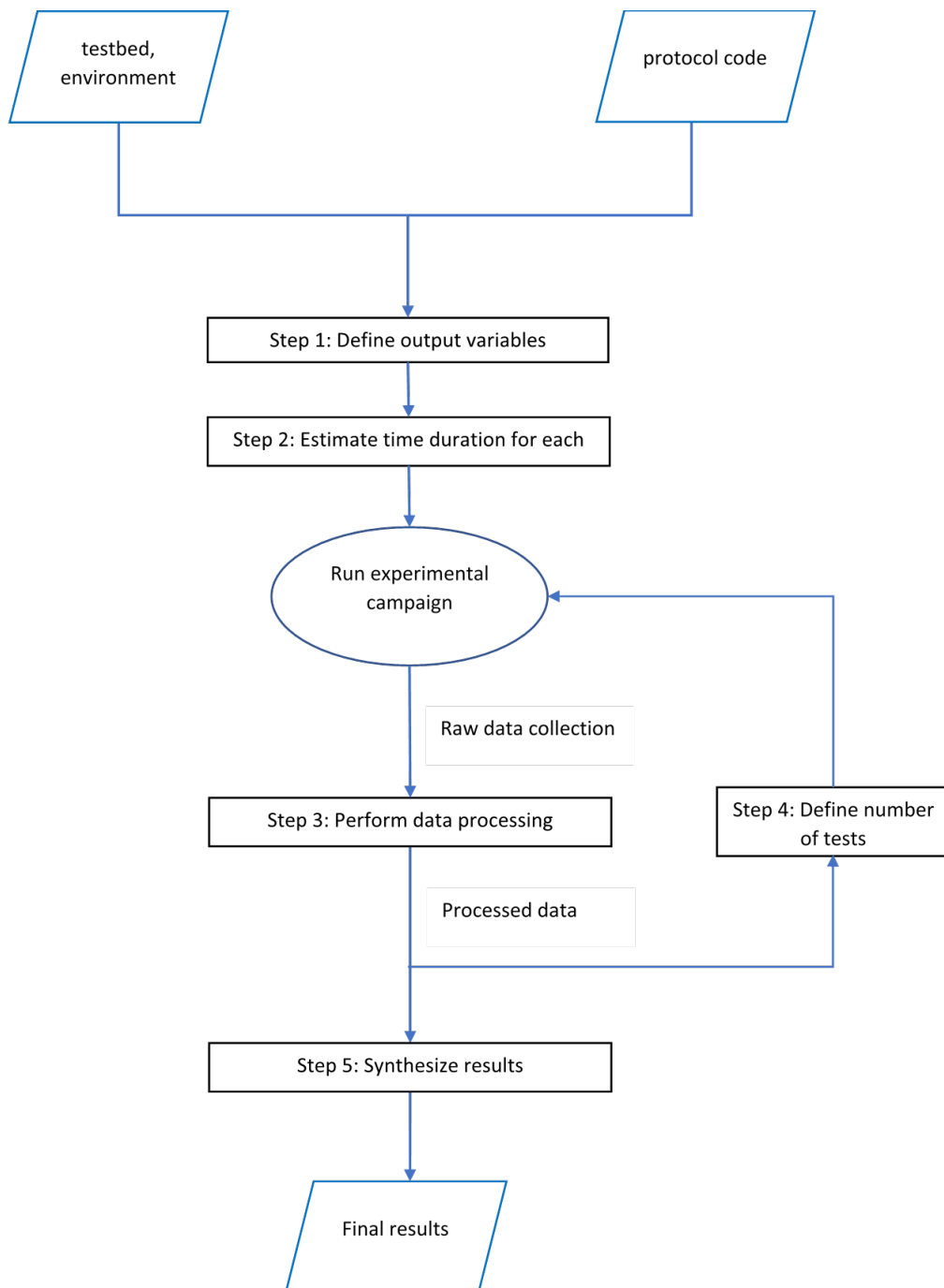


Figure 2.1: Proposed methodology for low-power wireless networking protocol evaluation

2.1 Inputs of the methodology

As it can be seen from the figure 2.1, the proposed methodology takes two inputs: the protocol code and the testbed/environment. The protocol code is the one that is actually evaluated. The testbed refers to the facility that is used to run the protocol code. During the last decade many testbeds for Wireless Sensor Networks have been deployed, like FlockLab [10] and Indriya [5]. These testbeds allow for testing in a more realistic setting compared to simulators and provide visualization tools and additional capabilities like power profiling and tracing. As discussed in the introduction, the environment has an impact on the performance evaluation. Furthermore, it can be difficult to assess if it stays in the same condition throughout the experimental campaign. For example, many Wireless Sensor Networks operate in the frequency band around 2.4 GHz, which is also used by the WiFi. This coexistence can cause a lot of interference. Since WiFi is largely correlated with the people's activity, there is a significant difference between day and night time. It should be noted that these two inputs are orthogonal to each other, i.e they can be chosen independently.

2.2 Step 1: Definition of output variables

After the two inputs are specified, it is then necessary to define the output variables to be collected throughout the experimental campaign. In other words, at this step the metrics, with respect to which the protocol is evaluated, need to be chosen. This choice heavily depends on the application scenario. For non real-time data collection protocols, we are interested in two different dimensions: the energy efficiency and the reliability.

2.2.1 Energy efficiency

In order to evaluate the energy efficiency of a protocol, several options are available. Two of the most prominent of ones are (1) the radio duty cycle (DC) and (2) the energy consumption. Each of these two metrics has its own advantages and disadvantages, which will be briefly discussed here. Since it has been shown that typically the radio component of a wireless sensor node consumes more energy compared to the other components by orders of magnitude, it is reasonable to estimate the energy efficiency of a node using the ratio of the radio-on time to the application period, i.e the radio duty-cycle. This metric is normalized and it can provide a fair ground to compare protocols independently of the platform. On the other hand, accessing the energy efficient through the current consumption of the nodes is also possible and can provide a more refined information for the lifetime of the wireless sensor nodes. The measured current values can be useful in designing power supply for a network, however, they may

not be an ideal way to compare protocols across different platforms, since the energy consumption of the platform itself can have an impact in the protocol evaluation.

2.2.2 Reliability

In a periodic data collection scenario, it is common that each source node produces a new packet every application period, which contains useful information about the sensed quantity (i.e temperature, humidity, etc). It is a task of the protocol to aggregate all the created packets at the sink in a reliable way. Therefore, it is reasonable to use the Packet Reception Ratio (PRR) as metric for the reliability. This PRR is defined as the ratio of the total received packets at the sink to the total number of created packets from all source nodes.

2.3 Frequency of data collection

In many research papers, only one value is aggregated and reported after all the experimental campaign. Although this approach might be simpler to implement and handle, it lacks depth of information. An example is presented on figure 2.2, to illustrate the limited information obtained from only a single reported value. In this example, two hypothetical time series of values are presented, which have the same mean value. Hence, if only the average value is reported, then there is no information about the trend. However, this lack of information might be problematic, for example if the average value is meant to estimate the expected metric on a longer time period. Since running a test for an arbitrarily long time interval is not feasible, it is desirable to run this test for a sufficient amount of time and based on that to be able to make predictions of what will follow and obtain results that are representative. In the above example, trying to make a prediction based on solely one value can lead to incorrect predictions. For these reasons, it is necessary to collect and observe data on a finer grain. Therefore, an important point of the proposed methodology is the suggestion that data should be collected every application period.

To continue illustrating the methodology, DC is selected to be the metric for energy efficiency and PRR to be the metric for reliability. Therefore, information about the radio DC for each node is collected per application period. Moreover, the sink records the packets that receives from each source node.

2.4 Step 2: Time duration of a single test

To describe fully the experimental campaign, it is necessary to specify the time duration for each test. This duration depends on the number of samples that are

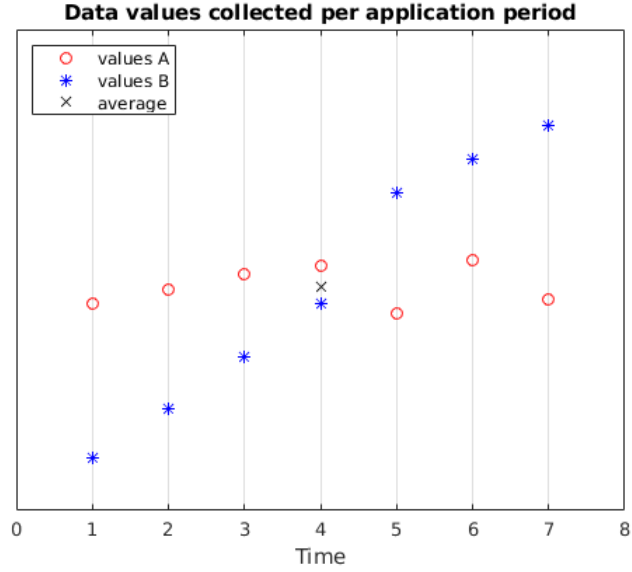


Figure 2.2: Two data series of values. *Two distributions can have the same mean, but very different tendencies*

required from the statistical analysis, in order to be able to make a statistically meaningful statement.

2.4.1 Protocol period vs application period

So far the assumption is that the proposed methodology is applied to periodic data collection scenarios. This, however, does not imply that the protocol to be evaluated will actually be periodic with the same period as the application scenario. This situation is illustrated in figure 2.3, where the protocol has twice as big period as the application scenario. The arrows on the top axis indicate when packets are generated and the arrows on the bottom axis when the protocol decides to send/forward the packets towards the sink. In such a case, nodes remain silent for half of the application periods and hence the metric of DC will have apparently half of its collected samples equal to zero. Therefore, any attempt to estimate a median in such sample set will be biased towards zero. In fact, the collected samples are not representative of the true median any more, because they are collected in a more fine grain than needed. Hence, it makes sense to first average some samples together before the statistical analysis. The exact number of samples that need to be aggregated will depend on the protocol functionality, thus must be flexible. We describe this by introducing an aggregation parameter p . This parameter describes the number of raw data samples one aggregates to produce “protocol samples”. In the given example, it makes sense to average two samples together to get a more meaningful sample

of the protocol.

There are two options to average the collected samples. If the averaging of the collected samples is done using a sliding window, then each collected sample is used more than once to obtain a protocol sample. In this case, however, less collected samples are required to reach a specific number of protocol samples. On the other hand, if a sliding window is not utilized, each collected sample is used only once and the protocol samples in the end are independent. However, in this case one protocol sample is created out of p collected samples and hence the number of the collected samples required to reach a specific number of protocol samples is much larger. Which aggregation methods should be used remains an open question, as both ways of averaging have their own advantages and disadvantages. In other words, the parameter p performs a downsampling of the collected data and creates a second version of them, even before any kind of processing is applied.

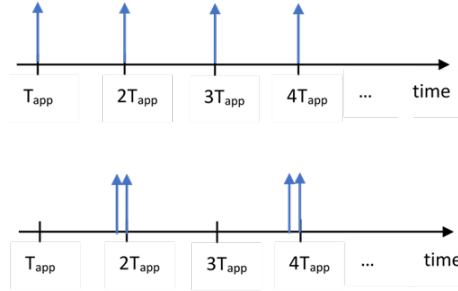


Figure 2.3: The top axis illustrates the periodic data generation of the application scenario. The bottom axis illustrates the periodic operation of a protocol with double period compared to the application scenario.

2.4.2 Statistical analysis and number of samples

In [12], authors underline that, in the case of a small sample size, reporting only the mean and standard deviation of that sample can be very misleading, even if the assumption that all the samples are taken from a Gaussian distribution holds. Even more important is that, typically, the distributions that occur are not Gaussian in the first place. In addition, the authors argue that the use of confidence intervals (CI) to find a range for the percentiles can provide a more meaningful insight and they propose a method to estimate such intervals independently from the underlying distribution of the examined quantity. Their method is based on the binomial distribution, and by specifying a desired confidence interval and a desired percentile, the minimum amount of samples needed, $N_{min.samples}$ is defined. By increasing the amount of samples, one can increase the chance to reduce the impact of extreme values in determining the bounds

of the confidence interval. In other words, the use of more samples can possibly lead to tighter confidence intervals.

To summarize, given (1) a desired confidence interval, CI (i.e 95%) and (2) the percentiles of interest (i.e median and/or 90th percentile), one can derive the minimal number of protocol samples that are required. Then, for a given aggregation parameter p and a defined aggregation method, one can derive the minimal number of raw data samples required. From the latter, the minimum number of application periods that one test should contain can be computed. Finally, from this number of application periods the minimal test length is estimated.

2.5 Step 3: Data processing

Having already defined the metrics that are of interest, the next decision to be made is to select the appropriate indicators that describe the protocol performance. An indicator expresses the way that is used to look at the specified metrics. In order to continue the illustration of the methodology, it is assumed that the chosen metric for the energy efficiency is the radio DC and for the reliability is the PRR.

There are several choices for the indicator for the DC. One such possible indicator could be the average DC across all nodes throughout the time duration of the test. Another indicator could be the median of the average DC throughout the time duration of the test, which provides similar information about the energy efficiency of a randomly chosen node. Last but not least, the maximum DC per protocol period could be an interesting indicator, as it can be used to estimate the time until the first node of the network fails.

Regarding the PRR, a possible indicator would be the average PRR across all the nodes. Another option would be to use the average PRR per source node, because such indicator can reveal information about the source nodes, whose packets are less likely to reach the sink. It has to be noted, though, that the choice of the indicators is not at the core of the methodology. Any valid indicator may be used with our methodology.

Provided that each test runs for a sufficiently long time interval, enough protocol samples are collected in order to apply the methodology suggested in [12]. More specifically, we consider the 95% confidence interval (CI) for the median of the maximal DC per protocol period. Hence, each test can be summarized by a vector that contains three values: the lower bound for the median of the maximum DC per protocol period, the upper bound for the median of the maximum DC per protocol period and the average PRR over all nodes across the test.

2.6 Step 4: Number of tests

Since each test is executed in a dynamic wireless environment, it is expected that the collected data will be characterized by some variability. As a result, intuitively it would be beneficial to run more than one tests to compensate for this variability. The minimum number of the tests that are required is again related to the statistical analysis that will be used.

To aggregate across all the executed tests, the same statistical analysis that is applied in step 2 is used, but this time on a higher layer. For example, regarding the reliability metric, each one of the average PRR values that were previously extracted per test can be seen as a new sample, and based on these new samples a CI can be estimated for the median. If more tests are executed, then some of the extreme samples will have less impact on the bounds of the CI of the median and possibly make the width of the CI of the median smaller. This is illustrated in figure 2.4, where it is assumed that each sample stems from a different test. In this figure, the incentive to run more tests is clearly depicted.

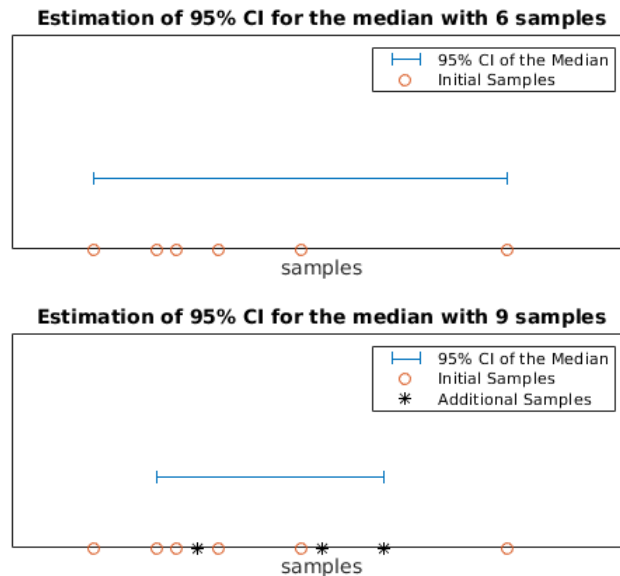


Figure 2.4: Estimation of the 95% CI for the median of two sets of samples with different size. *More samples can lead to tighter CI*

The same argument to calculate the minimum required number of tests can be stated also for the energy efficiency metric. More specifically, by applying the mentioned statistical analysis, a lower and upper bound were estimated per test for the maximum DC across all nodes per protocol period. Then, the set of upper bound values can be thought as a new set of data and by applying the

same statistical analysis, a new confidence interval for the median of the upper bounds can be estimated, with a given confidence level. However, the actual number is still the same, since the statistical method is metric independent.

2.7 Step 5: Results presentation

At step 3 it is explained how the processed data of a test, after applying a statistical analysis, can be aggregated in a vector that contains some information about the defined metrics. Having one such vector per test and after executing a specific number of tests, an interesting approach would be to represent each vector as a point into a n -dimensional space, where n is the number of the used indicators. This representation can be then used to formulate and propose definitions for repeatability, replicability and reproducibility. Such definitions are proposed in chapter 4.

Case study: Crystal protocol

In this chapter, the described methodology is used to evaluate the performance of a state-of-the-art low-power networking protocol which is called Crystal [8]. Crystal aims to achieve per-mille duty cycle with perfect reliability and very small latency.

3.1 Background

In wireless communication systems, the presence of two different signals at the same frequency and at the same time implies that there is interference. It is also called destructive interference, since it reduces the probability of a correct reception of any of the interfering signals at the receiver. However, due to the physical properties of the symbols used in IEEE 802.15.4, a phenomenon called *capture effect* occurs. In this case, a node can receive a packet despite interference from other transmitters under certain conditions:

1. the strength of that packet carrier signal has to be larger than the sum of the strengths of the interfering signals roughly by 3 *dB* and
2. the time difference between the arrivals of that signal and the interfering signals has to be smaller than the reception of the packet preamble. This time difference is in the order of 100 μs .

In addition, if the interfering signals are identical and have a tiny temporal difference i.e less than 0.5 μs , then constructive interference occurs, which significantly increases the probability of successful reception.

A novel protocol called Glossy was introduced in 2011 [6]. It leverages constructive interference to realize simultaneously fast network flooding and accurate time synchronization. In Glossy, one node initiates a flood with a single transmission. All neighboring nodes that receive the packet retransmit it immediately

and synchronously. With this flooding process, any packet sent by one node is eventually by all the other nodes. Then, in turn, another node can initiate a new flood. Because the retransmissions within a single flood are tightly timed, constructive interference is exploited and very good reliability is achieved. Experimental results indicate that a Glossy flood can lead to a successful packet reception with probability higher than 99.99%. In essence, it can be stated that Glossy protocol converts a multi-hop topology into a single hop network.

3.2 Description of Crystal

Crystal is a periodic protocol targeted for periodic data collection applications. It uses the Glossy protocol as a primitive to build reliable data collection. The period at which data are aggregated at the sink is called the *epoch*. Each epoch consists of a very short active portion, in which all nodes participate in data collection and a much longer sleep portion, when nodes consume very little power. The basic structure of an epoch in Crystal protocol can be seen in figure 3.1.

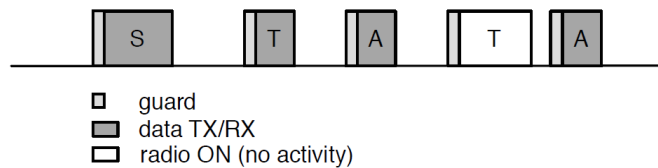


Figure 3.1: The basic structure of a Crystal epoch. *As long as there is sufficient time, source nodes retransmit their packets until they are acknowledged by the sink. Figure is adapted from [8]*

Each Crystal epoch contains a slot of S type. In the S slot, the sink initiates a Glossy flood in order to synchronize the nodes of the network. After that, successive pairs of transmitting and acknowledging slots, namely T and A slots, are repeated. More specifically, any source node that wants to send a packet initiates a Glossy flood at the sending slot. Therefore it is possible that many source nodes will transmit at the same slot. However, due to the capture effect, it is highly likely that at least one of these packets will be received successfully from the sink. In such case, the sink sends an acknowledgment, at the A slot, via a Glossy flood, where it mentions the ID of the sender of the successfully received packet. Due to Glossy reliability, with very high probability all nodes will receive this acknowledgment. In the subsequent T slot, the node with the acknowledged ID will not attempt to send his own packet anymore, but he will remain active to relay packets from other nodes. All the source nodes that still have a packet to sent, they will try again at the next T slot. This procedure repeats until all nodes have sent their packets. If the sink is unable to receive successfully a packet, then the A slot contains a negative acknowledgment.

The negative acknowledgments are also used to implement a distributed termination condition, so that nodes can return to sleep mode. More precisely, if a sink does not receive packets for a predefined number of slots, it goes to sleep. In addition, a node that receives R consecutive negative acknowledgments, it also goes to sleep, in order to save energy. As it can be observed from its operating principle, Crystal is a very reliable protocol, as nodes will constantly try to send their own data until there is no chance of a successful packet reception from the sink, due to time constraints.

3.3 Application of methodology on Crystal

In order to illustrate the use of the proposed methodology, we use a case study, where we apply it on Crystal protocol.

3.3.1 Inputs

The inputs of the methodology are the protocol code and the testbed. Regarding the protocol code, it should be mentioned that the original Crystal code is publicly available. However, the actual code that we use in this case study is not the original. We use instead a code that has the same functionality and is an implementation of Crystal protocol in Baloo [9]. Baloo is a generic network stack that is flexible and allows the implementation of a wide variety of network layer protocols, while introducing only limited memory and energy overhead. This choice was made because it was easier to modify the instrumentation of the Crystal implementation in Baloo, for raw data collection.

The examined application scenario is the typical case of non real-time periodic data collection. The used testbed is FlockLab [10]. The used platform is the *Tmote sky* [4]. Tmote sky is an ultra low power wireless module for use in sensor networks, monitoring applications, and rapid application prototyping. It leverages emerging wireless protocols and the open source software movement and it has been a popular choice for use in various applications. The network that is used for running the tests consists of 20 such nodes, which are randomly selected from the available set of nodes in FlockLab. Therefore, 19 source nodes generate data and the goal of the application is to collect the data from these source nodes at the sink node. Finally, in this application scenario the epoch duration is chosen to be 2 *sec*.

3.3.2 Step 1: Definition of output variables

The dimensions that we consider to be important in this setting are the energy efficiency and the reliability. As a metric for the former we choose the radio DC

and for the latter the PRR. At the end of each epoch, the DC value of each node during that epoch is reported. An example of recorded radio DC across different epochs for a certain node is illustrated in figure 3.2. In order to compute the PRR, we record all the packets that are generated at each node as well as all the packets that arrive at the sink node.

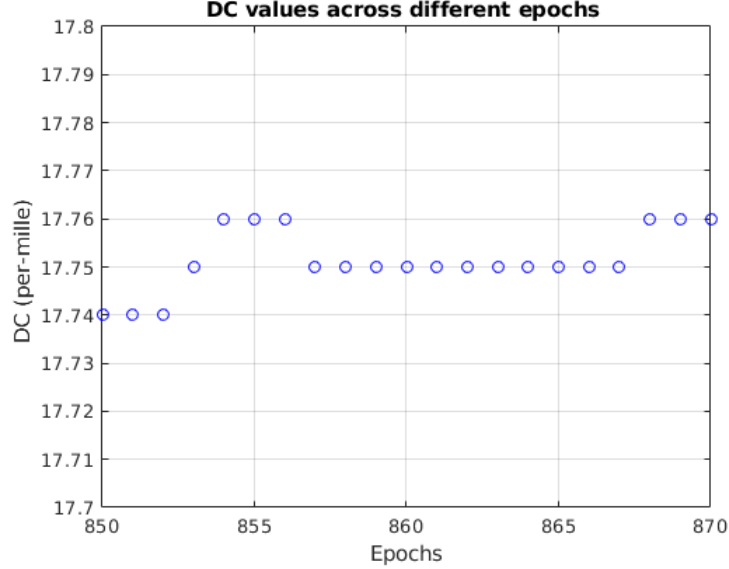


Figure 3.2: *The radio DC of a node shows small variations across different application periods.*

3.3.3 Step 2: Time duration of a single test

As it is explained in the methodology, the time duration of each test has a direct impact on the number of the collected raw data. In addition, we can assume that the Crystal protocol has the same period as the application scenario and as a result the parameter p has value $p = 1$. In other words, one application sample is also one protocol sample and hence there is no need for aggregation.

According to the statistical analysis in [12], the confidence level that is chosen is related to the number of samples that are available. Intuitively, the more samples we have, the more information we get about the underlying distribution. As a result, to achieve a given confidence level (e.g 95%) there is a minimum number of samples that is required.

In this case study, we choose to collect as much raw data as possible, with the hope of getting confidence intervals for the true median that are tight. The maximum test duration for one test in FlockLab is one hour. Therefore, we choose to use this maximum test duration. Since the epoch is 2 *sec* and we

collect one sample per epoch for the DC value per node, we end up with 1800 DC samples per node.

3.3.4 Step 3: Data processing

After running the experimental campaign for one test, all the raw data have been collected. At this point, the data processing begins. As an indicator for the PRR we use the average PRR across all nodes and we compute it by dividing the total number of packets that reached the sink by the total number of generated packets across all nodes.

As an indicator for the DC, we use the maximum DC. More specifically, for each epoch, we find the maximum DC across all nodes. This results to one value per epoch, i.e to 1800 samples of maximum DC per epoch. The next step is to compute the 95% confidence interval for the median (50th percentile) of the maximum DC per epoch. We choose the 95% as the desired confidence level, because it is one of the most popular choices across the scientific community. At this point, we apply the method proposed in [12]. The sample values are sorted in ascending order and then they are inserted into the vector x . At the relationship (9) of this paper, we substitute N by 1800, which is our sample size, and we try to find the largest value of the integer m , such that the right side is larger or equal than the desired confidence interval, which is represented in the left side. This integer value m is the index, based on which the confidence interval is estimated. The larger value m has, the more likely is that the confidence interval will be tighter, as more extreme values are not included for estimating both the upper and the lower bound.

Following this procedure, we search iteratively for the largest value of m that is suitable and we find m to be $m = 245$. Therefore, the confidence interval of the median for the maximum DC per epoch across all nodes is the interval $[x_{245}, x_{1566}]$. In other words, after processing the raw data for the energy efficiency for one test, the result is an upper and a lower value of a confidence interval. Overall, each test is described by these two values and one additional value for the reliability - the average PRR value across the test duration.

The same processing is repeated for all the executed tests. The resulting confidence intervals for all these tests are illustrated in figure 3.3. It should be noted here that the average PRR for all the executed tests was found to be 100%. This was not surprising, because, as described before, the Crystal protocol is very reliable, due to its operating principle.

3.3.5 Step 4: Number of tests

After finishing the data processing for one test, the 95% CI for the median of the maximum DC per epoch and the average PRR are calculated. In order to

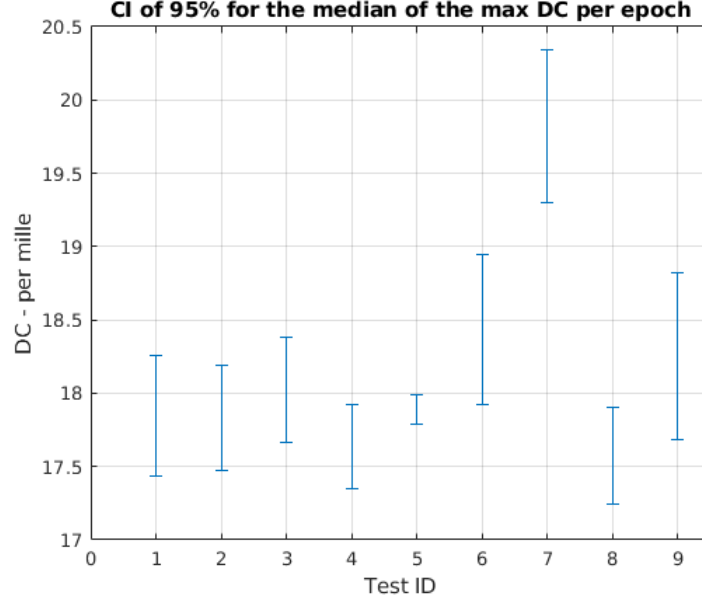


Figure 3.3: The 95% confidence interval for the median of the max DC per epoch across all nodes. *Due to variability some confidence intervals are tighter than others.*

make a rather conservative estimation about the DC, we choose to consider only the upper bound of the CI of the median for the maximum DC per epoch. Each of these upper bounds can be seen as a new sample point. Therefore we can use the same statistical method as in step 2 on a higher layer to compute a 95% CI of the median of those upper bounds.

One such upper bound is obtained from a single test. As it has been explained in chapter 2, having more tests and hence more of those upper bounds is better. However, we have already decided to run each test for a long time duration to have tighter 95% confidence interval per test. Therefore, it is not feasible to run a very large number of tests. From the Table 1 in [12], it can be seen that the minimum required number of samples to achieve a 95% CI for the median is six. However, we decided to run nine tests. For each of these nine tests, one upper bound is derived and it is inserted in vector y . The reason for choosing nine tests is that the 95% CI for the median of the samples in vector y can be defined as $[y_2, y_8]$. In other words, the CI for the median is not affected by the best and worst performing tests. In this example, as it can be seen from figure 3.3 test 8, which has the smallest upper bound and test 7, which has the highest upper bound are not influencing the estimation of the CI.

It should be mentioned here that for the estimation of the minimum number of required tests, the average PRR and its 95% CI could be used. However,

this minimum number would still be the same, as the used statistical analysis is independent from the chosen metrics.

3.3.6 Step 5: Results presentation

As described in the proposed methodology, each test is represented as a point into a n -dimensional space, where n is the number of the used indicators. In this case study, the dimension of the hyperspace is 2, since we have two different dimensions that we care about. As it has already been discussed, one dimension represents the energy efficiency. In order to be rather conservative in the estimation of DC, we choose the upper bounds of the 95% confidence interval for the median as a performance indicator. This choice, although justified, is not the only one possible. The other dimension is depicted by the use of the average PRR across the test. The two-dimensional plot is illustrated in figure 3.4.

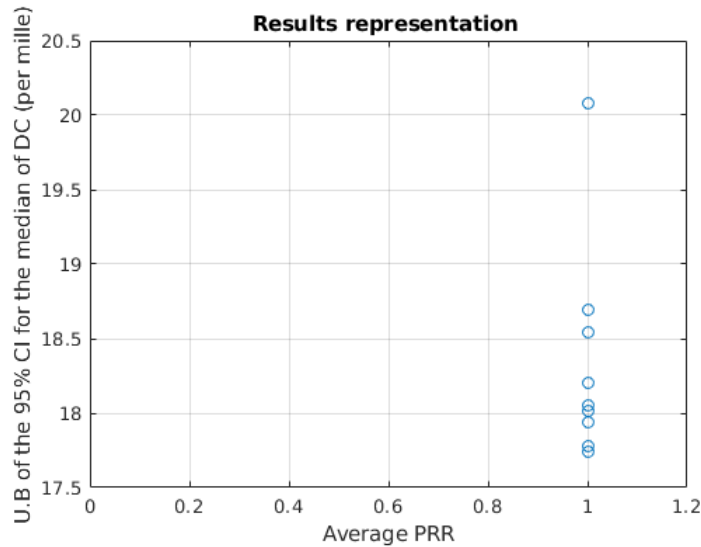


Figure 3.4: Results presentation at the last step of the proposed methodology. The horizontal axis represents the average PRR across each test. The vertical axis represents the upper bound of the CI of the 95% of the median of the maximum DC per epoch across all nodes.

At this point, the application of the proposed methodology in the case study has been completed.

Proposal of definitions

As it has been already stated, one of the main objectives of this semester project is to propose definitions for repeatability, replicability and reproducibility that are targeted to the field of low-power wireless networking, by refining the general ACM definitions that were presented in the introduction. To achieve that, the methodology presented in chapter 2 is utilized.

4.1 Repeatability definition

The repeatability of a protocol can be examined if the same team of researchers uses the proposed methodology and runs multiple tests. In the repeatability context, it is assumed that the inputs of the proposed methodology (protocol code, testbed and environment) have to be static across all different test executions. However, this hypothesis is difficult to hold and also to verify, since the wireless environment is usually characterized by a dynamic behavior, which may cause some variability at the results of each test.

At the very last step of the proposed methodology, the results are represented as points in the n -dimensional space. The intuition is that the closer these points are to each other, the more repeatable is the protocol. To illustrate this, the case study of the Crystal protocol will be continued here, by utilizing the same results representation.

In order to produce a qualitative result to resemble the repeatability, the concept of repeatability level can be defined. To achieve that, a way to measure distances in the n -hyperspace should be defined. After this step, the distances can be normalized. In our case, for example, both axes are normalized and hence, to normalize distances it is needed to divide them by a factor of $\sqrt{2}$. A final step would be to define a measure of the closeness of the points. One way to do that would be to use the average of the pairwise distances. Another way would be to use the average or the median distance of the points to the centroid. What is the most appropriate way is not obvious and remains an open question.

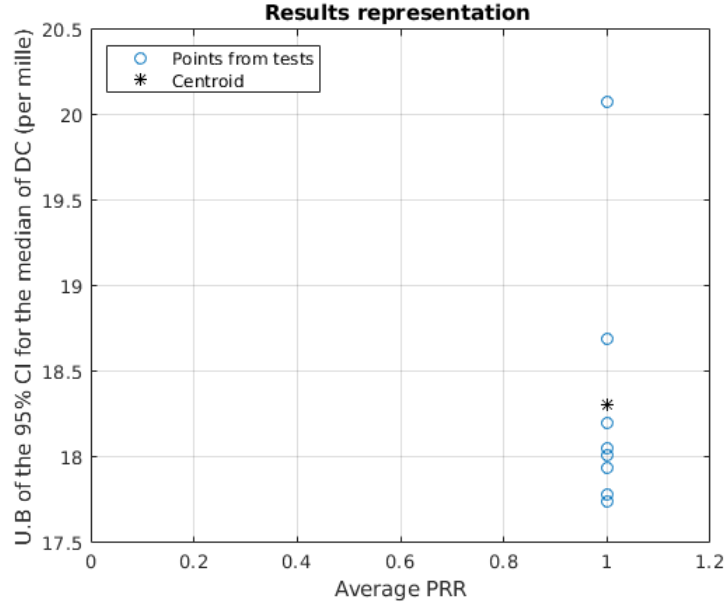


Figure 4.1: The results presentation and the centroid of the points. *Intuitively, the closer the points, the higher the repeatability.*

4.2 Replicability definition

The replicability of a protocol can be examined if a different team of researchers uses the proposed methodology. There can be defined two different types of replicability, based on the attempted way to replicate the results. These two types are the weak replicability and the strong replicability.

Weak Replicability: An experiment is defined to be weakly replicable if a different research team follows the proposed methodology by reusing the same artifacts, i.e. the same raw data with the original research team. Since the raw data are the same, and the proposed methodology is well defined and deterministic, all performance results should be (at least) weakly replicable. If the latter does not hold, two events are likely to happen. In the first place, it could have been that the data processing method is not documented well-enough. This may lead to different processing of the raw data and hence to different results. In a second place, it could have been errors in the data processing (i.e. not using all the available data, typos, etc.).

Strong Replicability: An experiment is defined to be strongly replicable if a different research team follows the proposed methodology, but reruns the experimental campaign from scratch. This means that there will be a new - and most likely different - set of raw data. By following the same way of

results representation, two different clusters of points in the n -hyperspace are created. The first cluster represents the results from the original team and the second cluster represents the results produced by using the newly collected raw data. It is intuitive that the closer these clusters are, the more strongly replicable the protocol is. In order to quantify the strong replicability, a metric needs to be defined. There are many possible choices for that. One simple but very intuitive example would be to use the L_2 norm for measuring distances in the n -hyperspace. After that, normalization can follow and then the centroid of each cluster can be computed. Then it is straightforward to compute the distance, d , between the two centroids and define, for example, the strong replicability level, SRL as $SRL := 1 - d$

According to this definition, it is clear that the higher the SRL, the more strongly replicable is the protocol. However, it should be noted here that this metric for estimating the replicability level is only one of the many possible. Whether this is a good choice it is an open question.

4.3 Reproducibility definition

The reproducibility of a protocol can be examined if a different team of researchers repeats the proposed methodology but it uses a different input. More specifically, to assess if a protocol code is reproducible, then this input should remain static. However the other input of the methodology can change. For example, the environment can change, by differentiating the time of execution between day and night time and hence by changing the interference level. A change can also happen by using a different testbed or by using the same testbed but a different set of nodes.

After specifying the new inputs, the experimental campaign is executed. Similarly, there will be a new - and most likely different - set of raw data. By following the same procedure as in the strong replicability case two clusters of points are generated. Again, according to our intuition, the closer these clusters are, the more reproducible the protocol is. A reproducibility level can be defined in a similar way as in the strong replicability case. Although this reproducibility level can be used to compare between different protocols, providing a threshold value, above which a protocol can be considered reproducible is a difficult task. In addition, it is likely that such threshold will depend on the application scenario.

Conclusion and Future Work

In this semester thesis, a methodology for evaluating low-power wireless networking protocols that serve a non real-time periodic data collecting application has been introduced. This methodology is composed of well-defined steps and it is based on a statistical analysis. The purpose of the methodology is to provide statistically sound arguments to answer questions that occur very often such as what should be the test duration and how many tests should be executed. The proposed methodology does not depend on the desired metrics.

In order to illustrate the use of the proposed methodology we evaluate the performance of the Crystal protocol, as a case study. As a last step, based on the proposed methodology we refine the terms of repeatability, replicability and reproducibility in the context of the low-power wireless networking.

This project provides many opportunities for further research. More specifically, in the second step of the methodology, where an aggregation method is needed to compose protocol samples from application samples. Hence, it may be of interest to further investigate which method is better suited for that purpose.

In addition, more advanced statistical methods and data mining techniques could be used to extract more information from the collected raw data. One example would be to investigate whether a regression model, based on the collected data, would produce better predictions for the evolution of the desired metrics. If that is the case, the number of the samples that are required to make meaningful statements has to be defined. A second example would be to investigate whether the use of the autocorrelation could lead to identification of patterns among the data. Such patterns could offer an added value and contribute in understanding better the behavior of the examined protocol.

Finally, finding a statistically meaningful way to quantify the repeatability, replicability and reproducibility levels is also of interest. One approach to do that could be to use advanced data mining techniques, such as clustering and pattern recognition in order to return numerical values that will indicate how well a protocol can be repeated, replicated and reproduced.

Bibliography

- [1] ACM Digital Library-Curation Platform Integrations. <https://www.acm.org/publications/dl-pilot-integrations>.
- [2] Artifact Review and Badging. <https://www.acm.org/publications/policies/artifact-review-badging>.
- [3] The ACM Task Force on Data, Software, and Reproducibility in Publication. <https://www.acm.org/publications/task-force-on-data-software-and-reproducibility>.
- [4] Tmote sky.
- [5] Manjunath Doddavenkatappa, Mun Choon Chan, and Akkihebbal L Ananda. Indriya: A low-cost, 3d wireless sensor network testbed. In *International conference on testbeds and research infrastructures*, pages 302–316. Springer, 2011.
- [6] Federico Ferrari, Marco Zimmerling, Lothar Thiele, and Olga Saukh. Efficient network flooding and time synchronization with glossy. In *Information Processing in Sensor Networks (IPSN), 2011 10th International Conference on*, pages 73–84. IEEE, 2011.
- [7] John PA Ioannidis. Why most published research findings are false. *PLoS medicine*, 2(8):e124, 2005.
- [8] Timofei Istomin, Amy L Murphy, Gian Pietro Picco, and Usman Raza. Data prediction+ synchronous transmissions= ultra-low power wireless sensor networks. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, pages 83–95. ACM, 2016.
- [9] Romain Jacob, Jonas Bächli, Reto Da Forno, and Lothar Thiele. Synchronous transmissions made easy: Design your network stack with baloo. 02 2019.
- [10] Roman Lim, Federico Ferrari, Marco Zimmerling, Christoph Walser, Philipp Sommer, and Jan Beutel. Flocklab: a testbed for distributed, synchronized tracing and profiling of wireless embedded systems. pages 153–166, 04 2013.
- [11] Karl R Popper. The logie of scientific discovery. *Hutchinson, London*, 1959.

- [12] Hanspeter Schmid and Alex Huber. Measuring a small number of samples, and the *3sigma* fallacy: Shedding light on confidence and error intervals. *Solid-State Circuits Magazine, IEEE*, 6:52–58, 04 2014.
- [13] Angus Stevenson. *Oxford Dictionary of English*. 01 2010.