UPPSALA
UNIVERSITET

IT Licentiate theses
2013-007

Meteorological Impact and Transmission Errors in Outdoor Wireless Sensor Networks

HJALMAR WENNERSTRÖM

UPPSALA UNIVERSITY
Department of Information Technology

Meteorological Impact and Transmission Errors in Outdoor Wireless Sensor Networks

Hjalmar Wennerström
hjalmar.wennerstrom@it.uu.se

December 2013

Division of Computer Systems
Department of Information Technology
Uppsala University
Box 337
SE-751 05 Uppsala
Sweden

http://www.it.uu.se/

Dissertation for the degree of Licentiate of Philosophy in Computer Science

ISSN 1404-5117
Printed by the Department of Information Technology, Uppsala University, Sweden

Abstract

Wireless sensor networks have been deployed outdoors ever since their inception. They have been used in areas such as precision farming, tracking wildlife, and monitoring glaciers. These diverse application areas all have different requirements and constraints, shaping the way in which the sensor network communicates. Yet something they all share is the exposure to an outdoor environment, which at times can be harsh, uncontrolled and difficult to predict. Therefore, understanding the implications of an outdoor environment is an essential step towards reliable wireless sensor network operations.

In this thesis we consider aspects of how the environment influence outdoor wireless sensor networks. Specifically, we experimentally study how meteorological factors impact radio links, and find that temperature is most significant. This motivates us to further study and propose a first order model describing the impact of temperature on wireless sensor nodes. We also analyze transmission errors in an outdoor wireless sensor networks, identifying and explaining patterns in the way data gets corrupted. The findings lead to a design and evaluation of an approach for probabilistic recover of corrupt data in outdoor wireless sensor networks. Apart from the experimental findings we have conducted two different outdoor deployments for which large data sets has been collected, containing both link and meteorological measurements.

Acknowledgements

I would like to thank my two supervisors Lars-Åke Nordén and Christian Rohner. Their team effort has helped me along the path to this licentiate thesis. They have given me the opportunity and support to choose my own research directions. I would also like to thank professor Per Gunningberg for creating an environment in which I was free, able, and encouraged to conduct research. A big thanks to all other members of the CoRe group, past and present, for their support and feedback on all things big and small, you are Fredrik Bjurefors, Volkan Cambazoglu, Laura Marie Feeney, Martin Jacobsson, Liam McNamara, Edith Ngai, Olof Rensfelt, Ioana Rodhe, and especially Frederik Hermans who taught me a lot about how research is done. A special thanks also to Carlo Alberto Boano and Thiemo Voigt who invited me into a successful collaboration together.

I would also like to acknowledge my second (interdisciplinary) research family, namely all the people within CNDS. A special thank you to all CNDS PhD students, who I shared the experience of going through the first two years with, you provided a broader context in which to view my own work.

I acknowledge CNDS, WISENET and INTERACT for the financial support of the research in one way or the other.

Finally I would like to thank family and friends, especially my wife Enni, for your enormous support and sacrifices you have done in order for me to continue with research with you by my side, without it there would be no thesis.

Included papers

Paper I: **A Long-Term Study of Correlations between Meteorological Conditions and 802.15.4 Link Performance**
Hjalmar Wennerström, Frederik Hermans, Olof Rensfelt, Christian Rohner, and Lars-Åke Nordén
In Proceedings of the Tenth IEEE International Conference on Sensing, Communication and Networking (SECON), June 2013, New Orleans, USA.

Paper II: **Hot Packets: A Systematic Evaluation of the Effect of Temperature on Low Power Wireless Transceivers**
Carlo Alberto Boano, Hjalmar Wennerström, Marco Antonio Zúñiga, James Brown, Chamath Keppitiyagama, Felix Jonathan Oppermann, Utz Roedig, Lars-Åke Nordén, Thiemo Voigt, and Kay Römer
In Proceedings of the 5th Extreme Conference on Communication (ExtremeCom), August 2013, Thorsmork, Iceland.

Paper III: **Transmission Errors in a Sensor Network at The Edge of The World**
Hjalmar Wennerström, Liam McNamara, Christian Rohner, and Lars-Åke Nordén
In Proceedings of the 5th Extreme Conference on Communication (ExtremeCom), August 2013, Thorsmork, Iceland.

Paper IV: **All is not Lost: Understanding and Exploiting Packet Corruption in Outdoor Sensor Networks**
Frederik Hermans, Hjalmar Wennerström, Liam McNamara, Christian Rohner, and Per Gunningberg
To be published at the 11th European Conference on Wireless Sensor Networks (EWSN), February 2014, Oxford, England.

List of work not included in this thesis

A: **A Study of Packet Errors on Outdoor 802.15.4 Links**
Hjalmar Wennerström, Liam McNamara, Christian Rohner, and Lars-Åke Nordén
In Proceedings of the Ninth Swedish National Computer Networking Workshop (SNCNW 2013), June 2013, Lund, Sweden.

B: **A Long-Term Study on the Effects of Meteorological Conditions on 802.15.4 Links**
Hjalmar Wennerström, Frederik Hermans, Olof Rensfelt, Christian Rohner, and Lars-Åke Nordén
In Proceedings of the Eight Swedish National Computer Networking Workshop (SNCNW 2012), June 2012, Stockholm, Sweden.

Contents

Part I

Thesis

Chapter 1

Introduction

1.1 Wireless Sensor Networks

Automated monitoring of an environment is essential to many areas of research, industry, and daily life. It can range from monitoring the soil moisture levels in the ground [43], vibrations in oil rig equipment [24] or the room temperature in your home [15]. These processes are measured and understood through the sensors that perceive the surroundings. As a result of developments in electronic miniaturization, a networking paradigm called wireless sensor networks (WSN) has been established. It further enhances the monitoring capabilities by decreasing the complexity of installation and maintenance while at the same time allowing for an increasing number of sensors that can be remotely accessible.

A WSN is composed of a network of *sensor nodes*, which typically are small resource constrained devices, equipped with sensors as well as wireless communication and computation capabilities. The key features shaping such devices, and in turn the whole network, are low energy consumption and low cost. The implications are that sensor nodes can have a small form factor and run on battery, facilitating monitoring of processes that previously was not feasible or economically justifiable. However, these key features also generate constraints, which in turn influence all aspects of research, design, and implementation of WSNs.

When dealing with such constraints, one of the biggest challenges from a communication perspective lies in providing reliable data collection while

consuming as little energy as possible. The fact that sensor nodes are wireless and use radio technology to transmit data challenges this reliability. This is because wireless data transmissions are inherently unstable and error prone, in turn requiring the sensor networks to be more resilient, which comes at the cost of more energy being used.

One way to ensure the performance of a WSN is to identify, study, and understand the processes challenging the reliability. Ideally, to maintain reliability the resilience of the network should be 'good enough' to protect against these challenges. However, since energy is a scarce resource, the better we understand these challenging processes and their implications, the more fine-tuned mitigating strategies we can use, prolonging the lifetime of the network.

One of the more evident threats to the reliability is the environment in which the network is deployed, with numerous potential effects where some are easier to predict and manage than others. Specifically, dynamic and ever changing processes, such as interference and climate for example, are more complex since the network needs to be reactive or even proactive to such changes. Failing to do so can lead to transmission errors and loss of data, impacting the performance in a negative way. These transmission errors, occurring between the sending and reception of data, arise due to an incorrect radio transmission. This can for example happen due to a weak signal where the receiver cannot distinguish what was sent from the background noise.

In this thesis, we focus on aspects that characterize outdoor WSNs. By this we do not mean that all outdoor WSNs are created equal but that there are distinct general features that differentiates them from indoor WSNs. Therefore, this thesis aims at highlighting findings from experiments in outdoor WSNs that typically cannot be studied in an indoor environment alone. The focus lies on studying how the performance is influenced by meteorological factors and transmission error characteristics in these networks.

1.2 Outdoor Wireless Sensor Networks

Already from the inception of wireless sensor networks, many envisioned application areas meant that the network would be deployed outdoors. In fact, one of the very first deployments, monitoring the micro climate of birds' nests on Great Duck Island, was placed outdoors [26]. Since then, deployment scenarios of outdoor WSNs have grown and example areas today

include environmental monitoring [10, 27, 41], habitat monitoring [14, 22, 26], agriculture [32, 39], disaster relief [16, 17], structural monitoring, and logistics [7, 29, 44].

Although there are many examples of sensor networks outdoors, it is still considered a challenge to successfully deploy and run such a network. This difficulty can be seen in publications detailing experiences from such deployments, often containing helpful insights into common pitfalls and unexpected behavior [6, 24, 25, 40]. Reasons for experiencing difficulties in these deployments comes from the fact that they are typically remote and inaccessible, making it harder to correct any mistakes once deployed. Another complicating factor is the difficulty to understand the environment and the implications it will have on the successful operation of the WSN.

Naturally, any outdoor deployment is exposed to an ever-changing meteorological environment potentially impacting the performance of the WSN. If the meteorological environment is harsh then the physical environment can be more compliant, typically with more open spaces. This in turn enables wireless sensor nodes to be in line of sight of one another, allowing for longer distance links.

Another influential factor is the fact that many sensor nodes use the unlicensed 2.4 GHz ISM band to transmit data. Due to it being unlicensed, it is shared among many devices (WiFi-routers, wireless phones, baby-monitors, microwaves etc.), and as a result interference can be a problem. However, in many outdoor scenarios this might not be a big issue, especially if they are deployed in an non-residential environment where such devices are less common. This implies that interference is likely less prevalent in outdoor WSNs, creating a radio environment that differs to that of the indoor scenario.

Additional features of outdoor WSNs include providing power to sensor nodes, the possibility to harvest energy from the environment, problems with plants and wildlife (bio fowling), and how localization can be done. Although these aspect are not adressed in this thesis they contain further implications that need to be understood in order to operate WSNs outdoors. Specifically batteries have an important role since most batteries are affected by temperature [31]. This means that batteries for an outdoor deployment are likely to fluctuate in voltage [6] and likely have a shorter lifetime compared to if they were used indoors, further motivating the need reduce the power supply in any outdoor WSN.

These highlighted aspects illustrate ways in which outdoor wireless sensor

networks can differ from their indoor counterparts. Therefore, in this thesis we focus on examining and exploiting some of these differences through experiments and findings from wireless sensor networks deployed outdoors.

1.3 Contributions

The thesis focus is on the performance impact due to meteorological conditions, and characteristics of the consequences due to transmission errors in outdoor wireless sensor networks.

The main contributions of the thesis is,

I) Collection of two extensive datasets of link measurements from two outdoor WSNs over a long period of time. One of the dataset also includes high-quality meteorological measurements for the entire one and a half year measurement.

II) Through analysis of collected dataset, identify temperature as opposed to humidity or precipitation as the highest correlating meteorological factor in an outdoor WSN.

III) Provide an analytic model describing the impact of temperature on low-power wireless transceivers through conducting controlled experiments as well as outdoor measurements.

IV) Identify and explain biases in the way payload bit errors occur due to 802.15.4 symbol coding. Propose a method for statistically reconstructing corrupt packets in an 802.15.4 outdoor WSN leveraging the discovered biases.

Chapter 2

Meteorological Impact

2.1 Meteorological Factors

Meteorological conditions are known to influence radio signals. However the impact and vulnerability depends largely on the frequencies used, where higher frequencies (>10 GHz) are more susceptible to attenuation of the signal due to rain for example [30, 13]. Wireless sensor nodes use frequencies in the 2.4 GHz ISM band or less, which should make them impervious to any impact of such conditions. However, experiences from real world deployments tell us something different. There are numerous cases where sometimes, unexpected performance variations have been attributed to changing meteorological factors [2, 9, 29, 36, 39].

This shows that although radio waves with frequencies of 2.4 GHz may likely not be affected there are other factors impacting the ability of a WSN to transmit data. It could be due to the sensor node itself being exposed and affected, the surrounding environment changing or other aspects that changes due to a change in the meteorological conditions. A few illustrative examples include,

- the sensor node experiencing different temperatures and humidity during a day
- rainfall, snow, or wind changing the physical environment by causing pools of water, snow cover, or debris blowing around affecting the signal propagation

- secondary effects where spring and summer generate more foliage, changing the physical environment leading to fading and shadowing

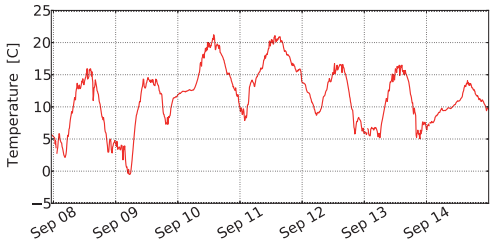
where the latter two are more specific to the local environment in which the sensor network is deployed. One can indeed study such local effects, yet they can be difficult to generalize to more than one deployment since the exact physical environment is so different between deployment sights.

Instead, what we focus on in this thesis is the question of what meteorological factors, such as temperature, humidity and precipitation, impacts the sensor node and its ability to transmit data. This has the potential of being more general, and as such the findings should be transferable to new deployments.

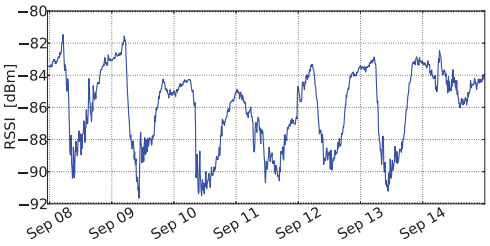
2.2 Measuring Meteorological Impact on 802.15.4 Links

Measuring how the quality of a link changes over time and correlating that to meteorological conditions provides insights into what are the dominating factors. The basis of any meteorological impact on the performance of a sensor node or network needs to be explained in terms of a physical phenomenon. Therefore, from a communication perspective, it is natural to start measuring the impact on the lowest level of the communication stack, the physical layer. A common standard in WSNs today is the IEEE 802.15.4 [20] standard, which defines the physical and medium access control (MAC) layer.

At the physical layer, the two most common metrics of signal quality includes received signal strength indicator (RSSI) and link quality indicator (LQI). RSSI measures the strength, in dBm, of the incoming packets radio signal as perceived by the transceiver and LQI measures the amount of corruption in the preamble, i.e. the start of a packet. They both give an indication on a per packet basis of how 'good' the reception was, where the values are comparable in most, but not all, cases. These metrics provide a snapshot measure of the quality of a link between two sensor nodes. One can then compare changes in the quality of the link to changes in meteorological variables to work out how they are correlated. An example of how RSSI can vary over time, and the corresponding temperature variation, can be seen in Figure 2.1. The major drawback of looking at these values, indicating the quality of the link, is the fact that they do not describe when data packets are lost, since they are only computed on received packets. Additionally, a change in RSSI or



(a) Ambient temperature

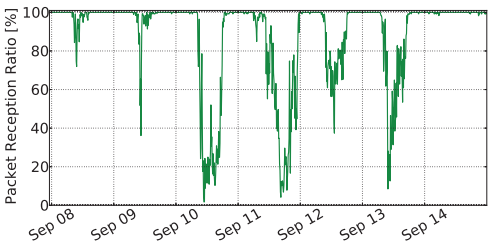


(b) RSSI for one link

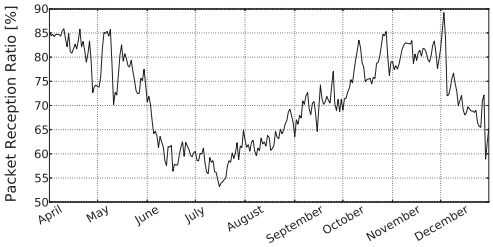
Figure 2.1: Example of how temperature and RSSI measurements for one link can look like. There is a visible pattern in both graphs, the inverse pattern in RSSI indicates a negative correlation. Measurements from September 2012.

LQI might not signify that things are better or worse in terms of successfully receiving data packets.

To that end, a complementary way is to look at how well a link is able to successfully transmit data packets. This is more useful from an application perspective since what is important is if the data arrives successfully or not, which is a more binary metric. If the quality of a link varies but all packets get successfully delivered, the application still works as it should. It is only when packet transmissions start to fail that there is a problem (which is of course coupled to the quality of the link). This can be measured by computing the packet reception ratio (PRR), which is the ratio between the number of successfully received packet and the number of sent packets measured over some time interval. A common issue with such a metric is the somewhat binary behavior of links where they typically either work very well or very poor [3, 35]. An example of this can be seen in Figure 2.2a where



(a) PRR computed over 10 minutes for one link



(b) Average daily PRR for all links in a network

Figure 2.2: Two examples of how successful packet transmissions can be measured over time. Comparing (a) to plot in Figure 2.1 indicate some correlation but with more sudden fluctuations.

the PRR for one link is shown. Since most meteorological measurements are continuous values, the mapping to an almost binary value might be misleading.

Lastly, the performance of the whole network can be studied and how it coincides with meteorological factors. The network performance can for example be expressed as an aggregate of the performance of all links in the network and provide a macro view of the processes. An illustrative example of the performance variation can be seen in Figure 2.2b. This is far away from the low level RSSI or LQI measurements, which are more coupled to the physical event, yet it can describe a smoother average of all the links, showing a system wide impact. A downside to this approach is that some links always get all the data through or never get any data through no matter the meteorological conditions, effectively obfuscating the aggregated result.

We find that these three levels are complementary and illustrate link changes with different perspectives. All three levels are used in Paper I and in Paper II we focus on individual links.

2.3 Findings and Insights from Previous Studies

The meteorological impact on outdoor WSNs is measured by collecting trace data from a deployment and collecting information regarding the meteorological conditions during the time and location of the trace data collection. Findings from previous studies are sometimes contradictory and hard to compare since the experimental measurements are done in a variety of ways. However, there are a few valuable, general findings outlined here.

Temperature has been studied the most [5, 8, 19, 39], and although some results are conflicting the dominating result is the negative impact of temperature on RSSI [5, 8]. Implying that as temperature rises the RSSI decreases, making the link worse. The explanation of this effect comes from temperature sensitive hardware components. Specifically, semiconductors leak current and in the transceiver this results in a drop in signal strength. Related studies on the impact of water content in the air, measured as humidity, rain and fog, also have discrepancies between results [2, 8, 36, 39]. Some observe that humidity improves RSSI [39], others that rain and fog reduces the amount of packets received [2, 36] or that rain has simply no observable impact at all [8].

Some of the contradictory results in previous work are likely due to a difference in methodology. It can be understood through the way measurements are made and the quality of the meteorological data that was used. Experiments typically run from a few hours to a few days with the exception of [8] who conduct measurements between January and March, capturing more rare events such as fog. The other aspect is how the meteorological data was obtained. Some use the on board sensors [4, 8, 19], other deploy a small weather station next to the deployment [2, 39] and lastly some use the public weather service for the area in which the deployment resides [8, 36].

These differences and shortcomings in how the measurements were collected strongly motivated our experimental design in Papers I and II. There we deploy a WSN next to a professional weather station [1], equipped with several high quality meteorological sensors. We also collect the sensor readings from the sensor nodes themselves. This allows us to get very good meteorological

measurements with high accuracy in time, place and sensing. Furthermore our experiment stretches over a long period of time, months and years, to provide a richer data set of different conditions, including seasonal variation.

The fact that several meteorological measurements correlate to a high extent (consider a scenario when it starts to rain, the temperature drops and the humidity rises) further complicates any analysis and conclusion. We observe this difficulty in the findings regarding relative humidity and temperature. Where a change in temperature, by definition, causes a change in relative humidity without the water content in the air changing, leading to observations of the impact of relative humidity. We therefore focus or analysis of absolute rather than relative humidity in Paper I.

However, on a general note such meteorological correlations are inherently difficult to address, given that we run real-world experiments. We provide two different approaches in Papers I and II to try and tackle this issue. In Paper I we decouple temperature and humidity measurements by fixing one of the values. For example, we only consider the measurements where the temperature is 25 degrees, using these measurements we can the compute a correlation between RSSI and humidity. The other approach in Paper II, is to have a supervised testbed where you are able to fully control the temperature, leaving other parameters unchanged.

Chapter 3

Transmission Errors

3.1 Packet Errors in 802.15.4

In the previous chapter, we discussed the impact of meteorological conditions and how they can influence the performance of a WSN. It is clear that a degraded performance leads to errors during transmission, something that is typically measured as packet loss. In this chapter, we shift focus toward these *transmission errors* and how they corrupt the payload of 802.15.4 packets.

Successfully transmitting data between sensor nodes is central to any wireless sensor network. However, wireless channels are prone to errors and more unreliable than a wired connection. Therefore, studying error processes in the network becomes relevant in order to better understand and combat such unwanted behavior leading to wasted energy. Specifically, studying packet corruption during transmission in IEEE 802.15.4 based networks can provide insights to WSN developers.

Transmission errors occur between the sending and the reception of data, where errors can manifest at different levels. There are several possible reasons for data corruption and packet loss, where either a weak or interfered signal are common causes. In an outdoor WSN we observe that it is primarily a weak signal that causes such issues. The standard way of combating transmission errors is either by simply re-sending the data, so called automatic repeat request (ARQ), or by forward error correction (FEC) where redundant data is encoded into a packet in order to be able to detect and correct corrupt data segments. The choice of strategy depends on the condi-

tions of the channel, where the more errors there are, the more difficult and costly it becomes to successfully transmit data.

Transmission errors ultimately manifest themselves as either missed or corrupt data packets. A missed packet is one in which the receiver never even starts to receive, due to failure to synchronize the transmission with the sender, resulting in a lost packet. A corrupt packet on the other hand is received but found to contain errors upon completed reception. Any packet that contains errors is discarded by the receiver and therefore also considered lost. The difference is that the receiver spends energy (a scarce resource in any WSN) on receiving a whole packet that is then found to contain errors and consequently discarded. We find the idea of utilizing corrupt packets interesting given that they are received anyway. This idea, inspired by Postel's law which proclaims "*be conservative in what you do, be liberal in what you accept from others*", is something we explore in Paper IV. There we suggest and implement one way in which to use corrupt data packets to do a probabilistic reconstruction of data.

3.2 Analyzing Transmission Errors

Transmission errors can be studied at different levels of granularity or abstraction, where making an appropriate choice depends on what is of interest. Here we highlight four different levels at which transmission errors can be studied. The way to perform such analysis is to capture corrupt packets, containing what was actually received and compare that to what was sent. The levels are:

- **Packet Level -** Measuring the amount of successful, corrupt and missed packets. This shows the prevalence of transmission errors on a per packet level. This should be most interesting from an application perspective.
- **Symbol Level -** Measuring the amount of corruption per symbol, a four bit data value. There are 16 symbols defined in the 802.15.4 standard (when using 2.4 GHz frequencies), and all transmitted data is cut up into symbols. Conversely, received data is combined into symbols by the receiver, making it a good metric for studying errors closely related to the actual transmission of data.

- **Bit Level -** A common metric is to compute the bit error rate (BER), indicating the probability of bits getting corrupted. It has strong roots in digital communication and relates to SNR measurements of a channel, making it a useful metric. However, in 802.15.4 data is not transmitted as bits but rather as symbols, making the BER metric misplaced for analysis where the origin of the errors are studied.
- **Chip Level -** At this the lowest level, chips are the binary values that are actually modulated onto the carrier. A total of 32 chips are used to represent one symbol in 802.15.4 (using 2.4 GHz). Received chips are mapped to symbols in the transceiver hardware, meaning that in most sensor nodes, the information of what chips were received is not available. So far, the way to get access to the chips is by using a software defined radio, providing complete access.

Out of these four, the symbol and chip levels illustrate they way in which data is *transmitted* in an 802.15.4 network and where the bit level illustrate the way in which data is *represented* in the payload. The use of these different levels can be seen in other works where chip errors [21, 42], symbol errors [34], and bit errors [18] are analyzed. In papers III and IV we study and discuss the impact of symbol and chip errors, and how they relate to bit errors in the payload, as well as looking at these error processes on a per packet level. We find that in an outdoor WSNs where there is little or no interference, the position of symbol errors within packets are evenly distributed, indicating that they are independent. Furthermore, we find that the same is not true for bit errors in the payload. Through analysis of the corruption we conclude that when symbol errors occur, the resulting bit patters have a visible structure, due to the way symbols gets confused as other symbols. This in turn explains one of the reasons why bit errors are not independent.

3.3 Dealing with Transmission Errors

Transmission errors and lost data packets are an unwanted yet occurring phenomenon in almost all WSNs. Therefore multiple strategies of combating such effects have been proposed in the research literature.

As mentioned earlier in this chapter, a common approach is to use error-correcting codes, possibly in combination with retransmission techniques, to

improve error resilience [21, 33]. These approaches typically introduce overhead in the size of the packet, or the number of transmissions, or both. For a wireless channel where errors are rare and not consecutive, retransmitting data might prove a good choice since very little overhead is added when everything is working fine. On the other hand when errors are more common, yet not severe, applying an FEC might be the better choice, since all packets will be more 'protected'. An issue with FECs is that they often work under a given channel condition. If the channel conditions change, for better or worse, the FEC will either not protect enough, resulting in broken packets, or overprotect the data, which wastes energy.

In an outdoor scenario the chances of being interfered by other devices is less likely. Instead, as mentioned earlier, what we have observed is that packet corruption is due to a weak signal. This has implications on the way packets get corrupted, where during interference a large consecutive block of data gets lost where as for a weak signal the process is more random and errors can be independent. This favors the use of an FEC, which typically perform better when errors are independent and less bursty.

An alternative approach is to accept transmission errors instead of trying to combat them, as a way to not increase energy consumption. Compressive sensing is one such approach, where a sparse number of sensing measurements are used to reconstruct the 'true' measurement [11, 43]. This is primarily used as a way to reduce the sampling and transmissions in a network and not as a way to reconstruct measurements from lost packets, however such examples exist [23]. Paper IV presents a complementary approach to allow transmission errors, where we try and reconstruct the original data based on the underlying error characteristics in an 802.15.4 WSN.

Chapter 4

Experimental WSN Deployments

4.1 Marsta

The first deployment is located in the outskirts of Marsta, a small village north of Uppsala in Uppland, Sweden. The site has a weather station, equipped with several professional grade sensors, operated by the department of earth sciences at Uppsala University. The area is an open grass field with no large vegetation. During summers the temperature reaches 30 degrees at most and during winter it goes as low as around -20 degrees.

The sensor network deployment consist of 16 TelosB sensor nodes [12] mounted on 4 poles along a straight line running 80 meters. The nodes are powered via USB and are connected to a testbed that can be used to perform controlled experiments. The sensor network has been running more or less continuously, collection link measurements since March 2012. A picture of the site and the weather station can be seen in Figure 4.1a.

The collected link measurements include information on a per packet level of RSSI, LQI, noise floor, time stamp, and payload. The measurements from the meteorological station, measured over 10 minutes, includes ambient temperature, humidity, precipitation, wind speed, wind direction, radiation etc. Additionally, we have also performed channel measurements of the 2.4 GHz ISM band using a Wi-Spy [28].

(a) Aerial view of the Marsta deployment with weather station.

(b) A pole with nodes and testbed equipment in the Abisko deployment

Figure 4.1: Pictures from the two deployments

4.2 Abisko

The second deployment is located in Abisko, a remote village above the arctic circle in northern Sweden. The network is deployed in the vicinity of an arctic research station [38]. The area is categorized as sub-arctic with long winters and short summers, as well as having very little precipitation with only 400 mm per year on average. The deployment sits in a field with arctic birch trees that bloom during summer.

The sensor network is deployed in a similar fashion to that in Marsta, with the difference that only 12 TelosB sensor nodes in total are used, with four poles and three nodes per pole. Figure 4.1b shows one of the poles with the three nodes mounted at different heights. The sensor network has been running since March 2013, with periods of downtime due to failures. The collected link measurements contain the same information as those in the Marsta deployment.

4.3 Lessons from Deploying Experimental Outdoor WSNs

Through experience of deploying two outdoor WSNs in remote locations we have had to address some of the challenges that most WSN developers would face during such deployments. Here we highlight two of the main issues that ultimately lead to the success of the deployments.

The first issue we came across is how to protect and encapsulate all the equipment so that it does not break while being deployed. Here is it essential to have a waterproof system, where everything is built to handle the outdoors for a long time. For us this meant special equipment, with so called IP68 classification which translates into dust and waterproof. Furthermore vegetation can pose a problem, especially in our case since we use testbed equipment connected through cables, making them vulnerable to any strain caused by growing plants. Our solution was to use cables with strain relief, and sometimes partition the cables in order to reduce the problem.

The second issue we faced was to monitor and being able to control the experiment remotely. To have the capability to know when things are working and when they are not, without having to go there, was essential to us since the deployments were far away and running over long periods of time. Our solution was to connect the testbed to the Internet, allowing full access to the system from our office. This also enabled us to have automated monitoring so that whenever something went wrong, we would receive an email detailing the problem. Given that the deployments were experimental, and prone to failures every now and then, this proved a very useful tool to have. It also allowed us easy access to the tracedata, which could be collected on a daily basis.

The deployment of any such experimental network requires a lot of planning and testing. We found that even though you do these things there are always going to be unexpected challenges or problems that need to be addresses 'on site'. We found that what it comes down to is reducing the number of unexpected issues as much as possible by thoroughly examining and testing the setup. Naturally there will always be unforeseen issues in these types of deployments since they are always the first of their kind. We learned a lot about the Marsta deployment that we were able to use when planning and setting up the Abisko deployment. Yet, the Abisko deployment, described in Paper III, came with its own set of issues.

Chapter 5

Summary of Papers

Paper I

A Long-Term Study of Correlations between Meteorological Conditions and 802.15.4 Link Performance
Hjalmar Wennerström, Frederik Hermans, Olof Rensfelt, Christian Rohner, and Lars-Åke Nordén. In Proceedings of the Tenth IEEE International Conference on Sensing, Communication and Networking (SECON), June 2013, New Orleans, USA.

Summary. The fact that 802.15.4 wireless links are affected by meteorological conditions is known, yet the complexity and interplay between meteorological conditions makes any short term ad-hoc measurement difficult to interpret. To this end, our paper aims at systematically and over long periods of time evaluate different meteorological conditions and how they correlate with link measurements.

The paper presents an analysis of six months of data from an outdoor WSN collocated with a high quality meteorological weather station. The variation in packet reception rate (PRR), both over long and short term, is shown as well as how RSSI can vary over a few days. Five meteorological factors and their correlation with RSSI and PRR are presented. The two most dominating factors studied, temperature and humidity, are decoupled to further implicate which has the bigger impact.

Contribution. The paper systematically and over long time studies the correlation to some of the most common meteorological factors that are

thought to influence WSN performance. We find that temperature correlates the most with RSSI and PRR and by decoupling temperature from absolute humidity we still see a stronger correlation to temperature in regards to RSSI.

My Contribution. I am the main author of this paper. I developed the initial idea and was responsible for the deployment and maintenance. I performed all the data analysis and wrote most parts of the paper in collaboration and discussion with the co-authors. I presented the paper at SECON 2013.

Paper II

Hot Packets: A Systematic Evaluation of the Effect of Temperature on Low Power Wireless Transceivers
Carlo Alberto Boano, Hjalmar Wennerström, Marco Antonio Zúñiga, James Brown, Chamath Keppitiyagama, Felix Jonathan Oppermann, Utz Roedig, Lars-Åke Nordén, Thiemo Voigt, and Kay Römer. In Proceedings of the 5th Extreme Conference on Communication (ExtremeCom), August 2013, Thorsmork, Iceland.

Summary. The paper is a focused study on how temperature affects sensor nodes, which is a large impacting factor in any sensor networks where temperature varies. Motivated by the observations in an outdoor sensor network an analysis in a controlled testbed is performed. The controlled setting agrees with the observed behavior outdoors and highlights additional aspects related to the temperature dependence of specific transceiver models. Based on the measurements both from the outdoor and controlled experiment an analytical model is derived to determine the Signal-to-Noise (SNR) ratio at at given temperature for a given transceiver model.

Contribution. The paper describes in detail the impact of temperature on RSSI using real world traces, controlled traces, and an analytical model to model SNR.

My Contribution. I provided the data and analysis for the outdoor deployment. I participated in the general discussions together with Carlo, where we analyzed the data, comparing the controlled experiments to the observation outdoors. I helped write the parts of the paper detailing the analysis of the outdoor deployment (Section 3.1).

Paper III

Transmission Errors in a Sensor Network at The Edge of The World
Hjalmar Wennerström, Liam McNamara, Christian Rohner, and Lars-Åke Nordén. In Proceedings of the 5th Extreme Conference on Communication (ExtremeCom), August 2013, Thorsmork, Iceland.

Summary. The paper presents a sensor network deployment in a sub-arctic environment. The specific environment with little or no wireless interference is the basis for an analysis of transmission errors in such a network. The paper details the occurrence of corrupt packets over weak links and the amount of corruption in each packet.

Contribution. This paper shows that bit error are not equally likely on all position within a packet even though symbol errors are evenly spread over all positions. The reason for this being a bias in the way the intended symbols are incorrectly mapped to other symbols.

My Contribution. I am the main author of this paper and was responsible for the deployment and maintenance in collaboration with Liam. I performed the data analysis and wrote most parts of the paper in collaboration and discussion with the co-authors. I presented the paper at ExtremeCom 2013.

Paper IV

All is not Lost: Understanding and Exploiting Packet Corruption in Outdoor Sensor Networks
Frederik Hermans, Hjalmar Wennerström, Liam McNamara, Christian Rohner, and Per Gunningberg. To be published at the 11th European Conference on Wireless Sensor Networks (EWSN), February 2014, Oxford, England.

Summary. Here we further investigate corrupt 802.15.4 packets and study in detail symbol corruption. We find that there is an inherent structure in the way symbols get corrupted and attribute that to implementation details in low power wireless transceivers. We propose a probabilistic method to infer what symbols were sent, given a received symbol. It is based on the information that symbol errors are uncorrelated, and by having an estimate of the probability of a broken chip. The result is a ranked list of possible sent values. We show that the ranking is sound as it ranks the correct symbol with a high probability in 95% of cases.

Contribution. The paper proposes a scheme to probabilistically infer what data was sent for a broken 802.15.4 packet. It highlights and explains the origin of patterns in the way symbol error occur that was previously not known.

My Contribution. I contributed with the experiential setup, data collection and initial analysis. I wrote minor parts of the paper together with Frederik and Liam, and took part in discussions of the work.

Chapter 6

Conclusions and Future Work

6.1 Meteorological Impact

In this thesis we show how the performance of an outdoor WSN fluctuates over time. By correlating the performance variation to meteorological factors in Paper I we find that temperature has the highest correlation. We then further study and determine the precis impact of temperature in Paper II, resulting in a generic first-order model describing the impact of temperature on a few common sensor nodes. These findings show how important it is to understand what temperature ranges the sensor nodes in a deployed network might be exposed to. They also provide a way to model how temperature can impact the radio transmissions, something that can have potential uses in several areas. The underlying cause of this effect originates in temperature sensitive hardware, which experience voltage drops and possibly higher noise levels as temperature rises. The focus of this thesis has not been to study these processes in detail but rather to investigate and determine their impact on the communication between wireless sensor nodes.

Future directions of this work could include studying and predicting the impact at different levels in a network. Questions such as how does the network topology change, how does this impact routing, can we include temperature in a link quality metric, is it possible to predict the impact of temperature 10 seconds, 10 minutes, 1 hour, or 1 day in advance. These questions could lead to findings that would enable networking strategies to be more proactive and not only reactive. Another aspect is to study how clock drift, and services that rely on synchronized time, are affected by this.

This is a well known area of research yet long term outdoor measurements might reveal new insights. It is also important to note that this negative impact of temperature is something that is likely reduced in future hardware components (although newer transceiver chips such as the CC2520 [37] are still affected). One of the reasons that it is so visible on a sensor node is that fact that is uses very low-cost components that are not as resistant to this effect as their more expensive counterparts.

Although the focus has been dominated by temperature, it is not the only factor. Naturally there are other aspects that can influence the performance as well, especially since the microclimate and surroundings of any location of a WSN is unique. Although further exploring additional factors in more detail is tempting, caution towards the generality of the findings is important. Therefore we think that investigating things like rainfall, humidity or snow depth in more detail might prove more challenging than temperature since they are more variable phenomenon. As an example, humidity should be studied at different levels of temperature and rainfall intensity to provide more qualified insights.

6.2 Transmission Errors

We depict the error characteristics of lost packets in two different networks in Papers III and IV and show how the modulation scheme and code words used impact the way errors manifest themselves. These findings helps to better understand corruption of packets and as we show in Paper IV, can be used to for example probabilistically reconstruct data. The idea of reconstructing data is one of the ways in which this information could be leveraged. The findings could for example also influence the design of FECs or other coding techniques to improve error resilience.

Continued work in this area includes showing the applicability of such a probabilistic reconstruction process by coupling it with a real sensing application. To better motivate any approach where corrupt packets are utilized one thing that should be investigated is the occurrence of them. Although some early analysis of this is presented in Paper III, there are more questions to be asked such as, do corrupt packets come in burst or interleaved with other packets, how large part out of all packet loss could be expected to be corrupt, could this be coupled to RSSI or temperature. The key aspect that needs to be further shown is in what scenarios, if any, would it be useful to consider corrupt packets.

Bibliography

[1] Marsta weather station. `http://celsius.met.uu.se/?pageid=12`.

[2] G. Anastasi, A. Falchi, A. Passarella, M. Conti, and E. Gregori. Performance measurements of motes sensor networks. In *Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, MSWiM '04, New York, NY, USA, 2004. ACM.

[3] N. Baccour, A. Koubâa, L. Mottola, M. A. Zúñiga, H. Youssef, C. A. Boano, and M. Alves. Radio link quality estimation in wireless sensor networks: A survey. *ACM Trans. Sen. Netw.*, 8(4):34:1–34:33, Sept. 2012.

[4] K. Bannister. Impacts of thermal reduction in transceiver performance on outdoor sensing networks. Master's thesis, Arizona State University, Phoenix, AZ, USA, 2009.

[5] K. Bannister, G. Giorgetti, and S. K. S. Gupta. Wireless sensor networking for hot applications: Effects of temperature on signal strength, data collection and localization. In *Proc. of the 5th Workshop on Embedded Networked Sensors (HotEmNets)*, 2008.

[6] G. Barrenetxea, F. Ingelrest, G. Schaefer, and M. Vetterli. The hitchhiker's guide to successful wireless sensor network deployments. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, SenSys '08, pages 43–56, New York, NY, USA, 2008. ACM.

[7] L. Benini, D. Brunelli, C. Petrioli, and S. Silvestri. Genesi: Green sensor networks for structural monitoring. In *Sensor Mesh and Ad Hoc Communications and Networks (SECON), 2010 7th Annual IEEE Communications Society Conference on*, pages 1–3, 2010.

[8] C. Boano, J. Brown, Z. He, U. Roedig, and T. Voigt. Low-power radio communication in industrial outdoor deployments: The impact of weather conditions and ATEX-compliance. In *Proc. of the* 1^{st} *International Conference on Sensor Networks Applications, Experimentation and Logistics (SENSAPPEAL)*, 2009.

[9] B. Capsuto and J. Frolik. Demo abstract: A system to monitor signal fade due to weather phenomena for outdoor sensor systems. In *Proceedings of the Fifth International Conference on Information Processing in Sensor Networks (IPSN'06)*.

[10] R. Cardell-Oliver, M. Kranz, K. Smettem, and K. Mayer. A reactive soil moisture sensor network: Design and field evaluation. *International Journal of Distributed Sensor Networks*, 1(2):149–162, 2005.

[11] C. T. Chou, R. Rana, and W. Hu. Energy efficient information collection in wireless sensor networks using adaptive compressive sensing. In *Local Computer Networks, 2009. LCN 2009. IEEE 34th Conference on*, pages 443–450, 2009.

[12] Crossbow Inc. TelosB datasheet. `http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/TelosB_Datasheet.pdf`.

[13] A. Dissanayake, J. Allnutt, and F. Haidara. A prediction model that combines rain attenuation and other propagation impairments along earth-satellite paths. *Antennas and Propagation, IEEE Transactions on*, 45(10):1546–1558, 1997.

[14] V. Dyo, S. A. Ellwood, D. W. Macdonald, A. Markham, C. Mascolo, B. Pásztor, S. Scellato, N. Trigoni, R. Wohlers, and K. Yousef. Evolution and sustainability of a wildlife monitoring sensor network. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 127–140, New York, NY, USA, 2010. ACM.

[15] C. Ellis, M. Hazas, and J. Scott. Matchstick: a room-to-room thermal model for predicting indoor temperature from wireless sensor data. In *Proceedings of the 12th international conference on Information processing in sensor networks*, IPSN '13, pages 31–42, New York, NY, USA, 2013. ACM.

[16] T. Gao, C. Pesto, L. Selavo, Y. Chen, J. Ko, J. H. Lim, A. Terzis, A. Watt, J. Jeng, B. rong Chen, K. Lorincz, and M. Welsh. Wireless medical sensor networks in emergency response: Implementation and pilot results. In *Technologies for Homeland Security, 2008 IEEE Conference on*, pages 187–192, 2008.

[17] S. George, W. Zhou, H. Chenji, M. Won, Y. O. Lee, A. Pazarloglou, R. Stoleru, and P. Barooah. Distressnet: a wireless ad hoc and sensor network architecture for situation management in disaster response. *Communications Magazine, IEEE*, 48(3):128–136, 2010.

[18] B. Han, L. Ji, S. Lee, B. Bhattacharjee, and R. R. Miller. Are All Bits Equal? Experimental Study of IEEE 802.11 Communication Bit Errors. *IEEE/ACM Trans. Networking.*, 20(6):1695–1706, 2012.

[19] M. Holland, R. Aures, and W. Heinzelman. Experimental investigation of radio performance in wireless sensor networks. In *Wireless Mesh Networks, 2006. WiMesh 2006. 2nd IEEE Workshop on*, sept. 2006.

[20] IEEE Computer Society. *802.15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs).*

[21] K. Jamieson and H. Balakrishnan. PPR: partial packet recovery for wireless networks. In *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, ACM SIGCOMM '07, pages 409–420, New York, NY, USA, 2007. ACM.

[22] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet. *SIGOPS Oper. Syst. Rev.*, 36(5):96–107, Oct. 2002.

[23] L. Kong, M. Xia, X.-Y. Liu, M.-Y. Wu, and X. Liu. Data loss and reconstruction in sensor networks. In *Proceedings of the 32nd IEEE International Conference on Computer Communications*, IEEE INFOCOM '13, 2013.

[24] L. Krishnamurthy, R. Adler, P. Buonadonna, J. Chhabra, M. Flanigan, N. Kushalnagar, L. Nachman, and M. Yarvis. Design and deployment of industrial sensor networks: experiences from a semiconductor plant and the north sea. In *Proceedings of the 3rd international conference*

on Embedded networked sensor systems, SenSys '05, pages 64–75, New York, NY, USA, 2005. ACM.

[25] K. Langendoen, A. Baggio, and O. Visser. Murphy loves potatoes: experiences from a pilot sensor network deployment in precision agriculture. In *Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International*, pages 8 pp.–, 2006.

[26] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, WSNA '02, pages 88–97, New York, NY, USA, 2002. ACM.

[27] K. Martinez, R. Ong, and J. Hart. Glacsweb: a sensor network for hostile environments. In *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, pages 81–87, 2004.

[28] Metageek. Wi-Spy 2.4i, Spectrum Analyzer for 2.4GHz ISM Band. `http://www.metageek.net/products/wi-spy`.

[29] L. Mottola, G. P. Picco, M. Ceriotti, c. Gună, and A. L. Murphy. Not all wireless sensor networks are created equal: A comparative study on tunnels. *ACM Trans. Sen. Netw.*, 7(2):15:1–15:33, Sept. 2010.

[30] A. Panagopoulos, P.-D. Arapoglou, and P. Cottis. Satellite communications at ku, ka, and v bands: Propagation impairments and mitigation techniques. *Communications Surveys Tutorials, IEEE*, 6(3), 2004.

[31] C. Park, K. Lahiri, and A. Raghunathan. Battery discharge characteristics of wireless sensor nodes: an experimental analysis. In *Sensor and Ad Hoc Communications and Networks, 2005. IEEE SECON 2005. 2005 Second Annual IEEE Communications Society Conference on*, pages 430–440, 2005.

[32] F. Pierce and T. Elliott. Regional and on-farm wireless sensor networks for agricultural systems in eastern washington. *Computers and Electronics in Agriculture*, 61(1):32 – 43, 2008.

[33] D. Schmidt, M. Berning, and N. Wehn. Error correction in single-hop wireless sensor networks: a case study. In *Proceedings of the Conference on Design, Automation and Test in Europe*, DATE '09, pages 1296–1301, 3001 Leuven, Belgium, Belgium, 2009. European Design and Automation Association.

[34] F. Schmidt, M. Ceriotti, and K. Wehrle. Bit Error Distribution and Mutation Patterns of Corrupted Packets in Low-Power Wireless Networks. In *Proceedings of the 8th ACM WiNTECH Workshop*, WiNTECH '13, Miami, Florida, USA, September 2013.

[35] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis. Understanding the causes of packet delivery success and failure in dense wireless sensor networks. In *Conference On Embedded Networked Sensor Systems: Proceedings of the 4 th international conference on Embedded networked sensor systems*, volume 31, 2006.

[36] J. Sun and R. Cardell-Oliver. An experimental evaluation of temporal characteristics of communication links in outdoor sensor networks. In *Proc. of the 2th International ACM Workshop on Real-World Wireless Sensor Networks (REALWSN)*, 2006.

[37] Texas Instruments. *CC2520 datasheet - 2.4 GHz IEEE 802.15.4 / ZigBee RF Transceiver*, revision swrs068 edition, dec 2007.

[38] The Swedish Polar Research Secretariat. Abisko Scientific Research Station. `http://www.polar.se/abisko`.

[39] J. Thelen, D. Goense, and K. Langendoen. Radio wave propagation in potato fields. In *Proc. of the 1st Workshop on Wireless Network Measurement (WiNMee)*, 2005.

[40] T. Voigt, F. Osterlind, N. Finne, N. Tsiftes, Z. He, J. Eriksson, A. Dunkels, U. Bamstedt, J. Schiller, and K. Hjort. Sensor networking in aquatic environments - experiences and new challenges. In *Local Computer Networks, 2007. LCN 2007. 32nd IEEE Conference on*, pages 793–798, 2007.

[41] G. Werner-Allen, J. Johnson, M. Ruiz, J. Lees, and M. Welsh. Monitoring volcanic eruptions with a wireless sensor network. In *Wireless Sensor Networks, 2005. Proceeedings of the Second European Workshop on*, pages 108–120, 2005.

[42] K. Wu, H. Tan, H.-L. Ngan, Y. Liu, and L. M. Ni. Chip Error Pattern Analysis in IEEE 802.15.4. *Mobile Computing, IEEE Transactions on*, 11(4):543–552, 2012.

[43] X. Wu and M. Liu. In-situ soil moisture sensing: measurement scheduling and estimation using compressive sensing. In *Proceedings of the 11th*

international conference on Information Processing in Sensor Networks, IPSN '12, pages 1–12, New York, NY, USA, 2012. ACM.

[44] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin. A wireless sensor network for structural monitoring. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, SenSys '04, pages 13–24, New York, NY, USA, 2004. ACM.

Part II

Papers

Paper I

Paper I
A Long-Term Study of Correlations between Meteorological Conditions and 802.15.4 Link Performance

Hjalmar Wennerström, Frederik Hermans, Olof Rensfelt, Christian Rohner, and Lars-Åke Nordén. *A Long-Term Study of Correlations between Meteorological Conditions and 802.15.4 Link Performance*
In Proceedings of the Tenth IEEE International Conference on Sensing, Communication and Networking (SECON), June 2013, New Orleans, USA.

A Long-Term Study of Correlations between Meteorological Conditions and 802.15.4 Link Performance

Hjalmar Wennerström, Frederik Hermans, Olof Rensfelt,
Christian Rohner, Lars-Åke Nordén

Department of Information Technology, Uppsala University, Sweden

Abstract

Outdoor wireless sensor networks are all exposed to a constantly changing environment that influences the performance of the network. In this paper, we study how variations in meteorological conditions influence IEEE 802.15.4 links. We show that the performance varies over both long and short periods of time, and correlate these variations to changes in meteorological conditions.

The case study is based on six months of data from a sensor network deployed next to a meteorological research station running a continuous experiment, collecting both high-quality link and meteorological measurements. We present observations from the deployment, highlighting variations in packet reception ratio and signal strength. Furthermore, we show how the variations correlate with four selected meteorological factors, temperature, absolute humidity, precipitation and sunlight.

Our results show that packet reception ratio and signal strength correlate the most with temperature and the correlation with other factors are less pronounced. We also identify a diurnal cycle as well as a seasonal variation in the packet reception ratio aggregated over all links. We discuss the implication of the findings and how they can be used when designing wireless sensor networks.

1 Introduction

Changes in the environment can heavily affect the performance of wireless sensor networks (WSNs) that are deployed outdoors. In particular, variations in weather, such as changes in temperature or precipitation, have been

reported to impact radio communication, leading to packet loss [1, 13] and affecting link quality metrics [3, 2]. Understanding the effects of meteorological conditions on sensor network communication will help to take such performance issues into account during design and deployment.

Earlier work has analyzed the effects of some meteorological parameters on radio links in sensor networks. However, despite the large body of work on the subject, no community-wide consensus has been reached, and a number of studies arrive at incompatible conclusions. For example, Anastasi et al. report that rain coincides with a decrease in packet reception [1]. Boano et al. find that it does not have a significant impact on network performance [3]. Thelen et al. report improved link quality during times of rain [15]. We believe that such discrepancies are largely due to differences in methodology. Whereas some work considers measurements from on-board sensors, others rely on very coarse data provided by public weather services. Similarly, the duration of observation differs from a few hours to up to a few days.

We address these shortcomings by conducting a six months experiment of a 802.15.4-based sensor network that is collocated with a meteorological research station. The station provides high-quality weather data that is accurate in terms of sensing, time and location. We correlate this data with continuous measurements on the network's communication performance. The experimental design allows us to study the effects of seasonal weather changes in addition to the short-term effects.

Our results characterize distinct short-term and long-term performance cycles that correspond to a diurnal pattern and some seasonal effects. We assess the underlying causes of these variations by showing how the received signal strength (RSSI) and packet reception ratio (PRR) correlate with meteorological conditions. We find that temperature has the strongest correlation with both RSSI and PRR among the studied factors. We observe significant drops in PRR even on strong links with an average PRR above 90%.

Our findings have implications for the design of outdoor sensor networks. We highlight the importance of node placement and what variations to expect in PRR. Furthermore, we believe that our results can inform the design of strategies for mitigating weather effects that deteriorate network performance.

In summary, our three key contributions are:

- Based on a long-term measurement, we show that the overall network performance of our outdoor open space sensor network contains a diurnal cycle and a slower moving seasonal change.

- We show correlations between meteorological factors and the 802.15.4 link metrics RSSI and PRR. By decoupling temperature and humidity, we identify that temperature is the dominating factor. In contrast to previous work, we conclude that rain has no observable impact on either RSSI or PRR.
- Through a systematic analysis over different types of links we show that packet reception consistently maintains a negative correlation with temperature, with links at the edge of reception range showing a stronger correlation.

The rest of this paper is structured as follows. Section 2 describes related work on link measurements in outdoor sensor networks. Section 3 details our experimental setup. We describe the most important observations made over the six months period in Section 4 and provide an analysis of the correlations between link measurements and meteorological factors in Section 5. In section 6, we discuss the implication of our findings, and then conclude in Section 7.

2 Related work

Variations in 802.15.4 link measurements due to meteorological impact on outdoor wireless sensor networks have been studied by several researchers in the past. The results vary, and conclusions on which meteorological effects influence radio transmissions differ. Here we summarize the findings from the sensor network literature.

2.1 Temperature

Temperature has been the main focus in past research, but there are discrepancies in the findings. Work by Holland et al. [6] concludes that temperature has no impact on RSSI. Their view is also shared by Anastasi et al. who do not observe a change in packet receptions over different distances during varying environmental conditions [1]. In contrast, Boano et al. [3] and Bannister et al. [2] specifically show how higher temperature can reduce the received signal strength on a sensor node. Boano et al. set up a controlled experiment showing a decrease in RSSI as temperature increases [3]. They reason that changes in temperature affect crystal accuracy that induce frequency shifts, and thermal transceiver noise, that may degrade performance [14]. The same correlation between received signal strength and temperature is also reported by Thelen et al. [15].

2.2 Air Water Content

The influence of the amount of water content in the air has been hypothesized both to improve and hinder radio communication in WSNs. Thelen et al. [15] conclude that a higher relative humidity improves the received signal strength and attribute the enhancement to a change in the reflection coefficient on top of the plant foliage at their deployment site.

On the other hand, Anastasi et al. [1] and Sun et al. [13] report that rain and fog cause a decrease in packet reception ratios. Similarly, Capsute et al. [4] report a drop in signal strength during rain and snowfall. Interestingly, these findings contradict the fact that radio signals on frequencies below 11 GHz should be unaffected by rain and fog [9].

More recent work by Boano et al. [3] shows that rainfall, fog and snowfall have no severe impact on the received signal strength between two motes during non-extreme conditions. They explain the contradiction to earlier findings by arguing that it is the change in temperature that causes degradation in signal strength during times of rain and fog, rather than the amount of liquid water in the air.

A key observation is that the often discussed metric relative humidity can be misleading since it measures the amount of water vapor the air can hold at a given temperature. Thus, relative humidity is highly dependent on temperature, since changes in temperature also change the relative humidity, even though the amount of water vapor in the air stays the same.

2.3 Temporal Changes

The influences that the environment has on links has also been shown in the past by studying the temporal patterns describing the link quality.

Sun et al. [13] show how PRR fluctuates over a single link during a few days, suggesting the presence of a periodic pattern, following shifts in daytime and nighttime. Others have also noted that there can be a large variation in the received signal strength and radio link performance during daytime and nighttime [3, 2, 15]. These variations are most prominently explained by the changes in temperature.

2.4 Our Reflections

Based on previous work we identified two aspects that came to heavily influence our experimental design.

First of all, we argue that the type of weather data that is used for comparison has a major influence on the obtained findings. We note that

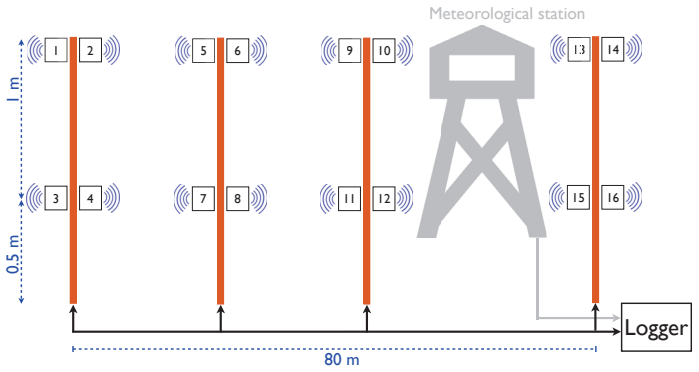


Figure 1: Setup of the outdoor sensor network. Sensor nodes are labeled 1 – 16. Nodes placed at 1.5m e refer to as high mounted and the ones at 0.5 meters as the low mounted.

meteorological measurements are obtained in different ways. They can be taken from the sensor board itself [3, 6], using publicly available weather data [13, 3, 7], or by deploying external sensors [1, 15]. The different ways of obtaining the data causes a large variety in the reliability of the measurements, in terms of location, timeliness and sensor accuracy. This also makes it more difficult to compare different findings.

Secondly, we note that previous work, except for [3], conduct experiments running from a few hours to a few days at most. This limits the possibility of experiencing a larger variety of meteorological conditions and capturing slow moving variations in the environment.

3 Experiment Overview

We designed an experiment where a WSN is deployed long-term collecting link measurements, while the environment is closely monitored during all times. We put up a sensor network at a meteorological research station that is equipped with professional-grade sensors for a variety of parameters. This enables us to study the effects of meteorological conditions on the sensor network's operation using high-quality meteorological data.

3.1 Sensor Network Deployment

The sensor network is comprised of 16 sensor nodes as shown in Figure 1. It is deployed on an open field with no trees or bushes in the surroundings. The sensors are running on a fixed power supply and do not depend on a battery, ensuring continuous operation over the entire long-term experiment. There are four 1.5 m high poles in total which are aligned along a 80 m straight line at distances of 0, 20, 40 and 80 meters respectively. We attach four nodes to each pole, allowing us to create links over a variety of distances. Two nodes are attached to the top of the pole (facing opposite directions), and two nodes are attached to the bottom, 0.5 m above the ground. This allows us to study how different heights and the reflection from the ground influence performance.

We use TelosB nodes [5], which are general purpose sensor nodes that are commonly used in the research community. The sensor nodes are equipped with 802.15.4-compatible CC2420 radio transceivers that operate in the 2.4 GHz ISM band [14]. Their prevalence in research projects and deployments makes them a natural choice for our study. The TelosB sensor node includes sensors to measure temperature, relative humidity and light.

Since the deployment site is at a remote location with very few people having access to it, interference caused by human activity is minimal.

3.2 Data Traffic Generation

The sensor nodes run a simple program we developed to periodically send packets along each radio link, i.e., to send packets between each possible pair of nodes. The transmission power of the nodes is set to the maximum of 0 dBm. One of the nodes acts as the designated sender, and sends a probing packet, 34 bytes in size, addressed to another node. Packets are sent with an inter-packet delay of 500 ms and each time the sender addresses a different node. If the addressed node receives a probing packet, it immediately sends back a response packet addressed to the sender. At the same time all other nodes overhear the packets being sent and are set to receive and log them accordingly. Every 30 seconds, the role of the sender is rotated among the sensor nodes in a round-robin fashion. This means that we have a total of 240 potential links in our network and the scheme generates at most 15 packet receptions (if all nodes can overhear it) per sent packet. On a typical day the experiment logs about two and a half million packet receptions.

For each received packet, a node logs a local timestamp, source and destination address, sequence number, the signal strength during packet recep-

tion, noise floor reading, checksum, payload and the Link Quality Indicator.

We use Sensei-UU, our relocatable wireless sensor network testbed [10], to monitor and control the experiment. All log messages from the sensor nodes are timestamped with a global time and stored for further processing.

3.3 Meteorological Station

The meteorological station is located near Uppsala, Sweden. The surrounding region is characterized by an average temperature of 16°C during summer and -5°C during winter with an annual precipitation around 450-650 mm. The station is operated by the Department of Earth Sciences at Uppsala University and all the meteorological data from the station can be viewed online [8]. It provides data for the following sensing modalities: temperature (at heights 0.84, 1.95, 4.78 meters), wind (direction and speed at heights 0.8, 1.7, 4.0 meters), precipitation, relative humidity, air pressure, snow depth. It also measures the incoming and outgoing long and short wave radiation that can be used to measure sunlight for example. Each of these values are sampled every ten seconds and an average is computed over ten minute intervals along with the standard deviation. Thus, the station provides measurements with a higher temporal resolution than usually available from public weather services. Since it is collocated with our sensor network, the measurements accurately reflect the meteorological conditions experienced by the network.

4 Observations

We present findings from six months of data collected between April 1st and September 30th, 2012. The experiment ran continuously during that period and generated approximately 475 million packet receptions. Due to such a large dataset being analyzed, the numbers presented here are based on averages over 10 minute intervals unless otherwise specified. This also matches the meteorological data which is obtained with 10 minute intervals. Figure 2 illustrates the change in mean and standard deviation of RSSI over one link for different time windows. It shows that the first sample stays within one standard deviation for a 10 minute window.

We start by making a number of observations based on the collected data in this section and in the next section analyze and draw conclusions based on correlations between meteorological factors and link measurements.

We first aggregate links to observe general trends over the entire experiment duration. This is followed by looking at one specific link over a shorter

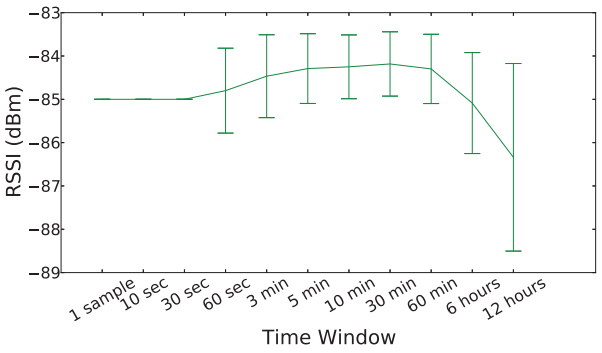


Figure 2: The RSSI of one link measured over different time windows with mean and standard deviation. The chosen 10 minute time window differs less than 1 dBm in RSSI for one standard deviation.

time span to highlight more fine-grained observations. In order to limit the scope, we have chosen to present four different meteorological variables based on previous work and our own insights on what is most interesting. They are temperature, absolute humidity, precipitation and sunlight.

4.1 Overall Packet Reception Ratio

Successful data delivery is an essential performance metric of any sensor network. Figure 3 shows the daily *overall mean PRR*, i.e., averaging all successful packet receptions from all nodes, for each day during the entire six months period. In Figure 3 the PRR changes over the months during the experiment. We observe the lowest average overall PRR on July 19th at 44.3% and the highest average PRR on May 7th at 92.8%. It illustrates how the ability of the sensor network to successfully deliver data can vary over long periods. A slow moving change is observed where the overall PRR decreases during the summer months and starts to recover during August.

Next we look at the stability of links and how it changes over time. Previous work such as Srinivasan et al. [12] has shown that PRR exhibits a cut-off behavior where it is typically either very good with a PRR above 90% or very poor with a PRR below 10%, and only a small portion of links have a PRR in between. Based on the same categorization, all the links in our deployment are plotted in Figure 4. First of all we see that

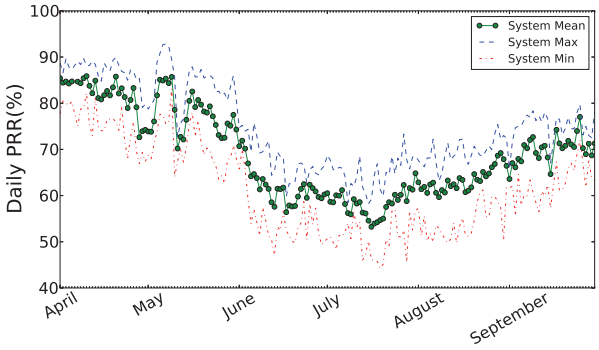


Figure 3: Daily PRR, computed as an overall average over all links during the entire six months period. Also shown is the minimum and maximum values of overall PRR over 10 minute intervals. There is a clear degradation in PRR during the months June, July and a slow recovery during August.

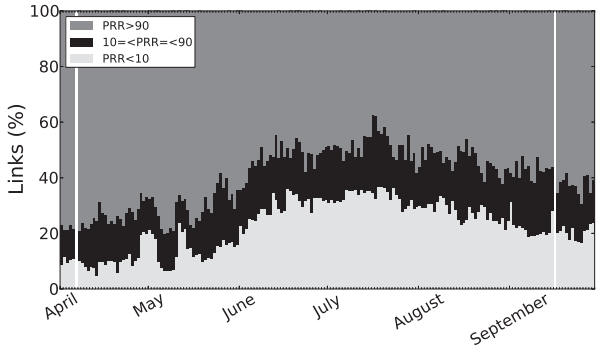


Figure 4: Percentage of links divided into three categories based on daily PRR average. High quality links decrease in early June and start to recover in August. The white stripes are two days when experiment was not running due to testbed failures.

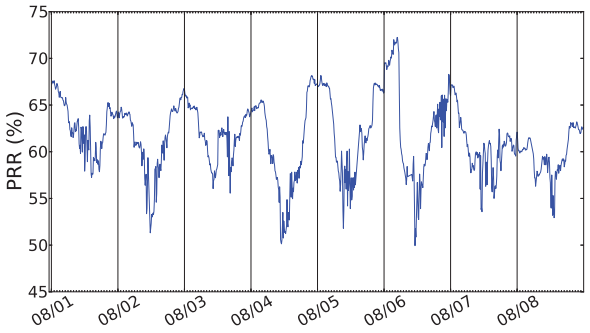


Figure 5: Overall mean PRR based on all links, computed over 10 minutes, during eight days in August showing a pattern of daytime lows and nighttime highs in PRR. The overall PRR can fluctuate more than 20% during one single day, seen during August 6th. Vertical lines indicate midnight.

the categorization matches with the observed daily PRR, where most links are either very strong (PRR $>$ 90%) or very weak (PRR $<$ 10%). Only about one fifth of the links have an intermediate PRR (90% $\leq$ PRR $\leq$ 10%) throughout the experiment. Note that it is the proportion of strong and weak links that changes throughout the experiment whereas the amount of intermediate links is fairly constant. This can be seen when strong links start to diminish in late May and at the same time the percentage of weaker links increase. This suggests that there are seemingly strong links, with PRR above 90% that can deteriorate over long time periods and become weaker. It implies that a high PRR during deployment is not necessarily a guarantee for a continued high PRR over time.

Figure 4 also shows that the amount of strong links fluctuates between 40-80% during the experiment. Analyzing this further reveals that 30% of all links are identified as being stable throughout the experiment, maintaining a PRR above 90% during all times. This means that about 50% of the links in our network are at some point strong with a daily PRR above 90% but deteriorate and become weaker. The 30% of links that remain stable throughout the experiment are seemingly unaffected by any changes in the environment. These links are categorized by the high-mounted short-distance links in our setup (see Figure 1).

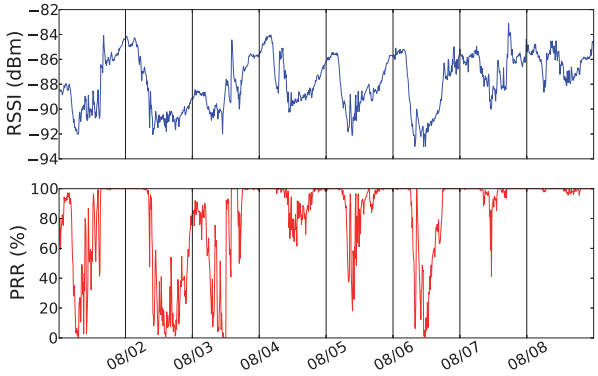


Figure 6: RSSI and PRR for the representative link during eight days in August. The link appears stable during nighttime but fluctuates during daytime. Vertical lines indicate midnight.

We have illustrated how the network's overall mean PRR changes over the different months of the experiment. Looking at a shorter timescale of a few days illustrates another characteristic of PRR. Figure 5 shows the mean overall PRR, this time computed over 10 minute intervals for eight days in early August. Here a diurnal variation in PRR is observed with a trend of daytime lows and nighttime highs. PRR can vary as much as 20% in one day, seen on August 6th.

4.2 Individual Link Performance

Up until now we have looked at how the aggregation of links in the network perform. The aggregated changes come from changes on individual links. It is therefore useful to look at how single links perform in order to get a more detailed view. To observe changes in individual links we have chosen to study one representative link. We selected the link between the high mounted nodes 1 and 13 that are 80 meters apart (see Figure 1), as our *representative link*. It is at the edge of the communication range while maintaining a high average PRR during large parts of the six months experiment. We argue that in a sensor network deployment, maximizing the distance between nodes while maintaining a high PRR is a desired feature and therefore studying

the performance of such a link is interesting. To further highlight observed variations we study the link over a shorter time window of eight days in early August.

To illustrate how an outdoor 802.15.4 link can vary over time, two link measurements of the representative link are presented in Figure 6. It shows the RSSI and PRR measurements over the eight day period. There are observable fluctuations in both parameters with a tendency to decrease during daytime and recover during nighttime.

The link readings in Figure 6 can be correlated with the four selected meteorological measurements in Figure 7. Figure 7a shows the temperature measured by the meteorological station and the computed absolute humidity. Figure 7b shows the measured precipitation over 10 minute intervals, as well as the amount of sunlight, measured as incoming shortwave radiation.

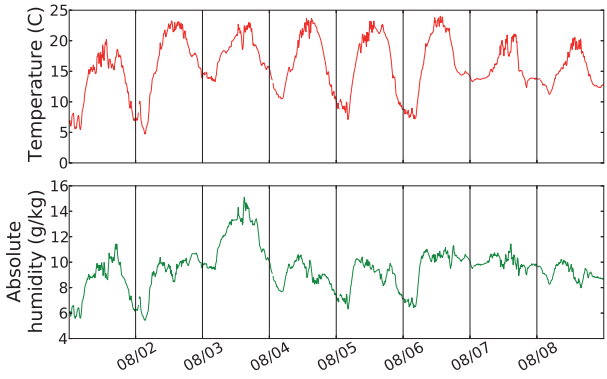
Looking at Figures 6 and 7 suggests that there are correlations between link measurements and meteorological factors. We explore these correlations further in the following section.

5 Analysis

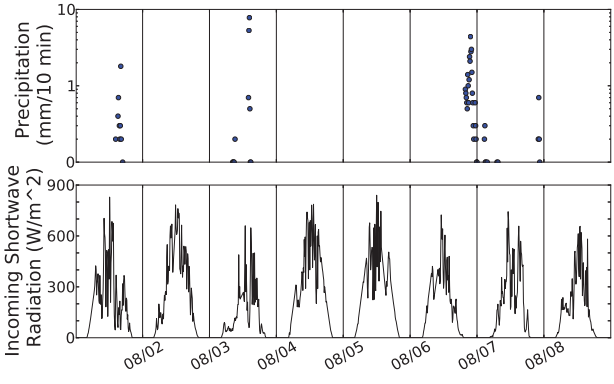
This section details the analysis of obtained measurements with a focus on correlations between the link measurements RSSI and PRR for the representative link and the four selected meteorological factors presented in Section 4.

To measure the correlation, we use Spearman's rank correlation [11] as our metric throughout the analysis. In the same manner as the more commonly known Pearson correlation, it is computed as a score between -1 and $+1$ where 0 indicates no correlation at all. It measures how well two variables *monotonically* increase (or decrease) in relation to one another. It does this by computing the linear dependence of the *ranked* variables as opposed to the variable values themselves. The metric emphasises a correlation where the change in one variable results in a change of the other variable, where the change rate might not be linear but steadily increasing or decreasing.

This section is divided into four parts, namely, temperature, air water content, temporal changes and further analysis. In each part we show correlations to link measurements and draw conclusions based on that and discuss how they compare to related work. An overview of the correlations for the representative link can be found in Table 1.



(a) Temperature and absolute humidity which is measured as grams of water vapor in kilograms of air.



(b) Precipitation in logarithmic scale and sunlight which is measured as the amount of incoming shortwave radiation in watts per square meter.

Figure 7: Measurements by the meteorological station during eight days in August. Vertical lines indicate midnight.

Table 1: Summary of the correlations between link metrics and meteorological factors for the representative link.

	RSSI	PRR
Temperature	-0.81	-0.54
Relative Humidity	0.12	0.21
Absolute Humidity	-0.72	-0.44
Precipitation	-0.13	0.06
Sunlight	-0.42	-0.33

5.1 Temperature

We sort all the link measurements for the representative link into buckets based on the temperature during the measurements. We generate one bucket for each degree of temperature. For each bucket the mean and standard deviation of RSSI and PRR is computed. Figure 8 shows the relationship between temperature and RSSI as well as temperature and PRR. There is a negative relationship between temperature and RSSI, similar to that reported by others [3], meaning that the RSSI decreases as temperature rises. We obtain a strong negative correlation between temperature and RSSI with -0.81 and -0.54 for PRR. For the representative link, when temperature rises above 16 °C the PRR goes below 90% on average. We expect that the increase in both PRR and RSSI around 25 °C in Figure 8 is due to a small dataset at those temperatures.

Now, we look at the overall mean PRR of all links in our deployment, and how it relates to temperature, shown in Figure 9. Again, measurements are sorted into buckets based on the temperature at the time of the reading. It illustrates that as temperature rises more links reach their sensitivity, and thus PRR decreases. The correlation between temperature and PRR can also be observed at this aggregated level, showing that it holds over a long period of time.

Conclusion

In accordance with previous work in [3, 2, 15] and opposed to [1, 6] we conclude that there exists a negative correlation between RSSI and temperature. In addition we have shown that there also exists a negative correlation between PRR and temperature in the aggregation of all links of our sensor network, which indicates a systematic degradation of successful packet reception as temperature increases. This was also observed for the

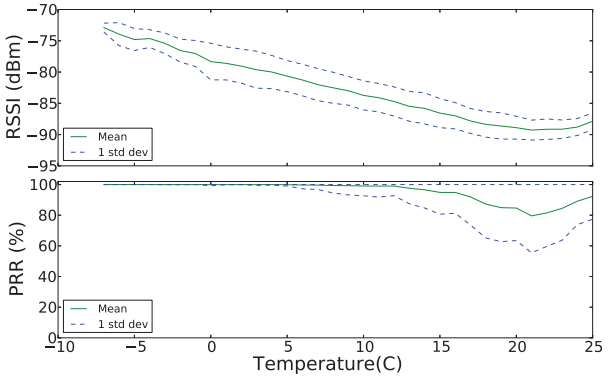


Figure 8: RSSI and PRR over the representative link and the relationship to temperature measured by the meterological station. Measurements from entire six months experiment. Spearman correlation: RSSI=-0.81, PRR=-0.54

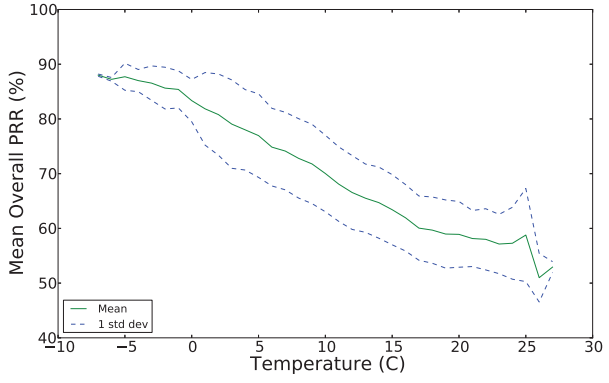


Figure 9: Relationship between overall mean PRR and temperature for our deployment. Measurements from entire six months experiment.

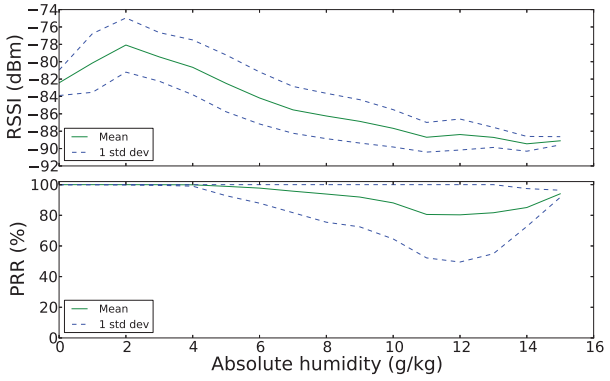


Figure 10: RSSI and PRR over the representative link and the relationship to absolute humidity in grams of water vapor per kilogram of air. Measurements from entire six months experiment. Spearman correlation: RSSI=-0.72, PRR=-0.44

representative link, a strong link at the edge of the communication range.

5.2 Air Water Content

Here we examine how two different measures of water content in the air influence the representative link. We begin with looking at absolute humidity that measures the amount of water vapour in the air. Then, we look at precipitation, in the form of rain as a measure of the amount of liquid water in the air. Relative humidity (Table 1) is a is a skewed metric, motivated in Section 2.2, therefore we do not study it further.

In the same fashion as before, link measurements are divided into buckets based on the absolute humidity when they were obtained. The relationship between absolute humidity to RSSI and PRR is demonstrated in Figure 10. There is a negative trend in both RSSI and PRR as the absolute humidity increases. The correlation is -0.72 for RSSI and -0.44 for PRR. The positive trend up until 2 grams of water vapor per kilograms of air in Figure 10 is likely due to a small measurement sample at those low levels.

In contrast, precipitation in Figure 11 shows no significant correlation with either RSSI or PRR. Here the correlation factors are -0.13 for RSSI

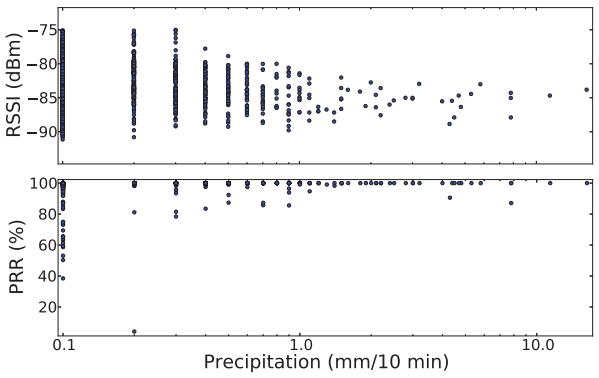


Figure 11: Scatter plot of RSSI and PRR for the representative link during precipitation. Note the logarithmic scale on the x-axis. Measurements from entire six months experiment. Spearman correlation: RSSI=-0.13, PRR=0.06

and 0.06 for PRR. Since the precipitation data is sparse the correlations and the plot only include data of when there was precipitation.

Conclusion

There exists a negative correlation between absolute humidity with RSSI and PRR. Our findings regarding RSSI and humidity contradict those of [15]. This is likely explained by the fact that we measure absolute humidity which is not directly dependent on temperature as is the case with relative humidity used in [15]. The implications are that as temperature drops, by definition the relative humidity increases, although the amount of water in the air is the same. This gives rise to a positive correlation between relative humidity and RSSI, also seen in Table 1.

However, we know from observing temperature and absolute humidity that they still correlate to a high degree, with a Spearman correlation of 0.75 for the six months experiment. It implies that changes in temperature and absolute humidity occur at the same time to a large extent Therefore it is not clear which correlation factor is most important. This motivated us to investigate further in Section 5.4.

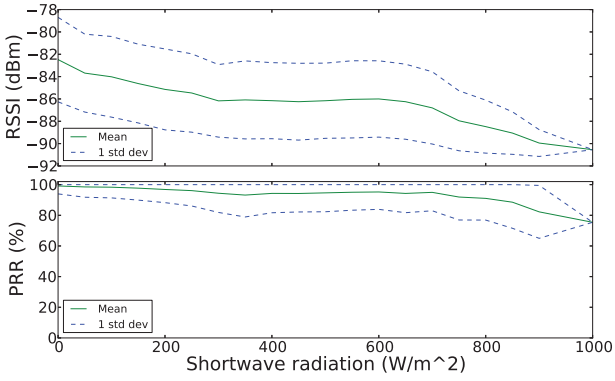


Figure 12: RSSI and PRR over the representative link and the relationship to sunlight measured by the meteorological station. Measurements from entire six months experiment. Spearman correlation: RSSI=-0.42, PRR=-0.33

Regarding precipitation, in contrast to [1, 13] no correlation between either RSSI or PRR was found. One explanation of the discrepancy could be that they look at precipitation over one or two individual days, and draw conclusions based on that. Despite the fact that we only look at the data during which there is precipitation the correlations remain low and without any clear indication. Our findings on the influence of precipitation on RSSI are however supported by the claims in [3]. One potential source for error is the fact that precipitation is one of the more difficult meteorological measurements to take. However we are confident that the meteorological station provides data with high quality precipitation sensors.

5.3 Temporal Changes

In Figures 5 and 6 we saw a tendency towards nighttime highs and daytime lows in both RSSI and PRR. One way to study the daily variations is to look at the amount of sunlight, measured as shortwave radiation. When the sun is not shining, i.e., during nighttime, the amount of incoming shortwave radiation is 0. We use this metric to look at how different amounts of sunlight influence RSSI and PRR for the representative link, shown in Figure 12. Here again we see a negative trend where more shortwave radiation results

Table 2: Average PRR and temperature during daytime and nighttime. For the aggregate of all links as well as the representative link.

	All Links PRR	Representative Link PRR	Avg. Temp (C)
Daytime	66.6%	92.8%	13.47
Nighttime	72.7%	98.4%	7.98

in lower RSSI and PRR. The correlation to RSSI is -0.42 and to PRR -0.33, which is lower than the ones for temperature and absolute humidity.

To further analyze the difference between day and night, we sorted all the PRR measurements into a daytime or nighttime category. The daytime category contains all the measurements taken between sunrise and sunset (determined by the shortwave radiation) and the nighttime category contains the measurements between sunset and sunrise. PRR is computed as an average over all links, as well as for the representative link for each of the two categories. The data was from the entire six months and the result is shown in Table 2. An overall 6.1% average increase in PRR during nighttime is observed, likely caused by the difference in temperature.

Conclusion

We found that PRR is on average higher during nighttime. This can be compared to the ambiguous findings in [13], where they experience both better and worse PRR during nighttime in two different deployments. The circumstances presented in [13] make it difficult to verify the existence of any influence. However, given the correlation between RSSI and PRR on the representative link to shortwave radiation as well as that of temperature (which is typically higher during daytime), we conclude that there is a performance degradation during daytime compared to nighttime. This is also supported by the fact that the average PRR during nighttime was 72.7% whereas only 66.6% during daytime.

5.4 Further Analysis

Here we detail two different further analysis results that decouple the correlation between temperature and absolute humidity, and show what the correlation between temperature and PRR looks like for all links.

Temperature vs. Humidity

From the correlations of our different meteorological factors with RSSI and PRR shown previously, it is clear that temperature and absolute humidity shows the strongest correlations out of the four. Even though we measure absolute humidity, which is not directly dependent on temperature, we know that the air can hold more water at higher temperatures. Based on this we decouple the dependence between the two factors and see how they correlate with RSSI. This is done by only including the RSSI and absolute humidity values for the representative link during a specific temperature (16 °C). Similarly, also looking at the RSSI and temperature values for the representative link during during a specific absolute humidity (6 g/kg). The two fixed values were chosen since they are the most common. The result can be seen in Figure 13, where a stronger correlation to temperature during which humidity was fixed is observed with a Spearman correlation of -0.43.

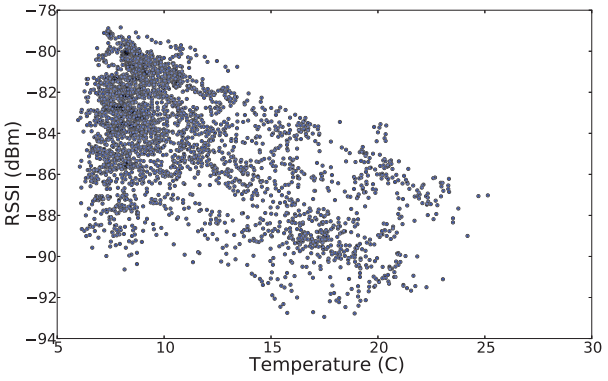
Correlations for all Links

The correlations presented so far have been computed for the representative link. To show how the correlations can differ over individual links, we plot the temperature correlation for each individual link during the entire experiment and the mean PRR for that link, shown in Figure 14. Here we see a variation in correlation over the different links with a tendency towards lower correlation when the link is either very strong or very weak. The figure also shows the difference between high links, low links and mixed links. Here high links are links between high mounted nodes (see Figure 1), low links between low mounted nodes and mixed links between one high and one low mounted node.

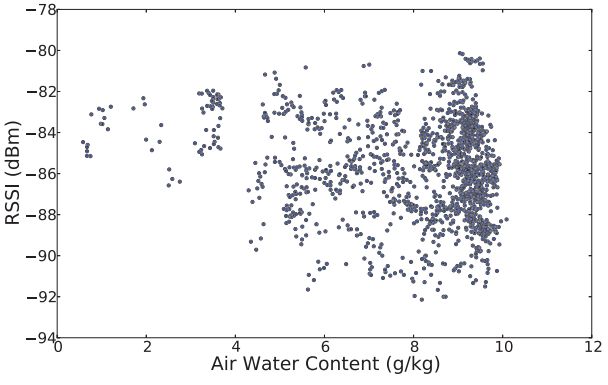
Conclusion

When the humidity is static there is a stronger correlation in temperature to RSSI than the other way around. We conclude that for the representative link, temperature is the more dominating factor. Based on this finding, we question if there exists a causal relationship between RSSI and absolute humidity.

The correlation between PRR and temperature varies between links, but it stays negative or just around 0 for all links. Links that are either very strong or very weak show a lower correlation to temperature. This is intuitive since they maintain a stable PRR while temperature varies.



(a) RSSI vs. temperature when absolute humidity is fixed to 6 grams of water per kg air. Spearman correlation is -0.43.



(b) RSSI vs. absolute humidity, in grams of water per kilogram of air, when temperature is fixed to 16°C. Spearman correlation is -0.15.

Figure 13: Scatterplotts of temperature and absolute humidity respectively where the other factor is kept fixed.

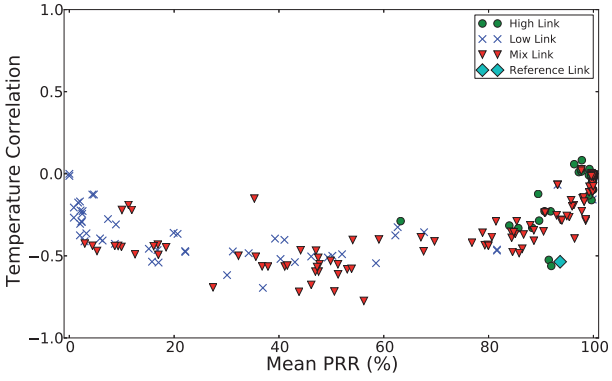


Figure 14: Correlation between temperature and PRR for all links, computed over the entire experiment. Links are divided into categories high, low and mix based on the height at which the sender and receiver were mounted.

6 Discussion

The fact that weather is such a chaotic system with many variables interacting with one another makes it challenging to study how certain factors influences link measurements. It requires a sound methodology that can obtain high-quality data. We believe that many of the contradicting results in previous work comes from this inherit difficulty of studying such a complex system. In this paper, we try to straighten out some of these contradictions by capturing many meteorological aspects and correlating them to link measurements.

We have found that temperature is the most highly correlated factor to RSSI and PRR. However, this does not imply that other factors are unimportant, since several of them can themselves influence temperature and vice versa. What is clear is the fact that the performance of an outdoor WSN most certainly is affected by, and correlated with, the meteorological conditions. This can be seen for example in the identified diurnal performance cycle and the slower moving seasonal variation.

Thus, in order to design, deploy and maintain well functioning WSNs,

we need to have a better understanding of how different conditions influence performance and how it can evolve over time.

6.1 Suggestions for outdoor WSN designers

Based on the findings presented in this paper we have come up with the following general suggestions for WSN designers.

Node Placement

Protect the sensor nodes from high temperatures as much as possible. Preferably, place them in a shaded environment or ventilate the housing.

Data Delivery

Expect variations in PRR both during different hours of the day as well as over different weeks and months. PRR will in general better during nighttime and colder months. Do not expect that all strong links (PRR $>$ 90%) stay strong, especially over long periods of time, but that the amount of intermediate links stays about the same.

Deployment Strategies

Once deployed, monitor the network for a minimum of 24 hours to ensure that the performance measure is high enough. Get an estimate of the conditions (temperature for example) by looking at historic data to understand what the network will be exposed to.

7 Summary

In this paper, we have deployed an outdoor WSN next to a meteorological research station collecting six months of link and meteorological measurements. We have shown general trends in the performance of the network, identifying a diurnal cycle where the network performed better during nighttime. We also observed a slower moving seasonal variation.

Furthermore we have analyzed the correlations of the four meteorological factors temperature, absolute humidity, precipitation and sunlight to the link metrics RSSI and PRR. By decoupling the temperature from humidity we concluded that temperature is the most dominant correlation factor. Based on the observed correlation between precipitation and the RSSI and PRR, we also conclude that there was no observable impact on either due to

precipitation. Finally, we have shown that the correlation between temperature and PRR varies over the links in our network, but remain a negative correlation.

Acknowledgments

We would like to acknowledge Hans Bergström and Sven Halldin at the Department of Earth Sciences at Uppsala University for their help in giving us access to the meteorological research station. This work was carried out within the Centre for Natural Disaster Science (CNDS) and Uppsala VINN Excellence Center for Wireless Sensor Networks WISENET.

References

[1] G. Anastasi, A. Falchi, A. Passarella, M. Conti, and E. Gregori. Performance measurements of motes sensor networks. In *Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, MSWiM '04, New York, NY, USA, 2004. ACM.

[2] K. Bannister, G. Giorgetti, and S. Gupta. Wireless sensor networking for hot applications: Effects of temperature on signal strength, data collection and localization. In *Proceedings of the fifth Workshop on Embedded Networked Sensors (HotEmNets' 08)*.

[3] C. Boano, J. Brown, Z. He, U. Roedig, and T. Voigt. Low-power radio communication in industrial outdoor deployments: The impact of weather conditions and atex-compliance. *Sensor Applications, Experimentation, and Logistics*, 2010.

[4] B. Capsuto and J. Frolik. Demo abstract: A system to monitor signal fade due to weather phenomena for outdoor sensor systems. In *Proceedings of the Fifth International Conference on Information Processing in Sensor Networks (IPSN'06)*.

[5] Crossbow Inc. TelosB datasheet. `http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/TelosB_Datasheet.pdf`.

[6] M. Holland, R. Aures, and W. Heinzelman. Experimental investigation of radio performance in wireless sensor networks. In *Wireless Mesh Networks, 2006. WiMesh 2006. 2nd IEEE Workshop on*, sept. 2006.

[7] W. Kang, J. Stankovic, and S. Son. On using weather information for efficient remote data collection in wsn. In *Workshop on Embedded Networked Sensors*, 2008.

[8] Marsta weather station, http://celsius.met.uu.se/?pageid=12.

[9] A. Panagopoulos, P.-D. Arapoglou, and P. Cottis. Satellite communications at ku, ka, and v bands: Propagation impairments and mitigation techniques. *Communications Surveys Tutorials, IEEE*, 6(3), 2004.

[10] O. Rensfelt, F. Hermans, L.-A. Larzon, and P. Gunningberg. Sensei-UU: a relocatable sensor network testbed. In *Procs. of WiNTECH '10*, September 2010.

[11] C. Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1), 1904.

[12] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis. Understanding the causes of packet delivery success and failure in dense wireless sensor networks. In *Conference On Embedded Networked Sensor Systems: Proceedings of the 4 th international conference on Embedded networked sensor systems*, volume 31, 2006.

[13] J. Sun and R. Cardell-Oliver. An experimental evaluation of temporal characteristics of communication links in outdoor sensor networks. In *RealWSN'06: Proc of the ACM Workshop on Real-World Wireless Sensor Networks in conjunction with ACM MobiSys*, Uppsala, Sweden, June 2006.

[14] Texas Instruments Inc. CC2420 - 2.4 GHz IEEE 802.15.4, ZigBee-ready RF Transceiver. `http://www.ti.com/lit/gpn/cc2420`.

[15] J. Thelen, D. Goense, and K. Langendoen. Radio wave propagation in potato fields. In *1st Workshop on Wireless Network Measurements*, 2005.

Paper II

Paper II
Hot Packets: A Systematic Evaluation of the Effect of Temperature on Low Power Wireless Transceivers

Carlo Alberto Boano, Hjalmar Wennerström, Marco Antonio Zúñiga, James Brown, Chamath Keppitiyagama, Felix Jonathan Oppermann, Utz Roedig, Lars-Åke Nordén, Thiemo Voigt, and Kay Römer. *Hot Packets: A Systematic Evaluation of the Effect of Temperature on Low Power Wireless Transceivers*
In Proceedings of the 5th Extreme Conference on Communication (ExtremeCom), August 2013, Thorsmork, Iceland.

Hot Packets: A Systematic Evaluation of the Effect of Temperature on Low Power Wireless Transceivers

Carlo Alberto Boano[1], Hjalmar Wennerström[2], Marco Antonio Zúñiga[3], James Brown[4], Chamath Keppitiyagama[5], Felix Jonathan Oppermann[1], Utz Roedig[4], Lars-Åke Nordén[2], Thiemo Voigt[5], and Kay Römer[1]

[1] Institute of Computer Engineering , University of Lübeck, Germany
[2] Department of Information Technology , Uppsala University, Sweden
[3] Embedded System Group , *TU Delft, The Netherlands*
[4] School of Computing and Communications , *Lancaster University, United Kingdom*
[5] Swedish Institute of Computer Science

Abstract

Temperature is known to have a significant effect on the performance of radio transceivers: the higher the temperature, the lower the quality of links. Analysing this effect is particularly important in sensor networks because several applications are exposed to harsh environmental conditions. Daily or hourly changes in temperature can dramatically reduce the throughput, increase the delay, or even lead to network partitions. A few studies have quantified the impact of temperature on low-power wireless links, but only for a limited temperature range and on a single radio transceiver. Building on top of these preliminary observations, we design a low-cost experimental infrastructure to vary the on-board temperature of sensor nodes in a repeatable fashion, and we study systematically the impact of temperature on various sensornet platforms. We show that temperature affects transmitting and receiving nodes differently, and that all platforms follow a similar trend that can be captured in a simple first-order model. This work represents an initial stepping stone aimed at predicting the performance of a network considering the particular temperature profile of a given environment.

1 Introduction

Wireless sensor networks (WSNs) have proven to be an excellent monitoring tool and nowadays many installations exist. They are, for example, used to monitor natural phenomena such as glaciers, infrastructures such as bridges, or production processes on oil platforms. Many of these deployments are heavily exposed to the environment and experience extreme temperature changes within a day and over seasons. Temperature has a significant impact on wireless communication and a system has to be designed to handle all possible temperature changes over the deployment lifetime. This is of particular importance if we rely on the system and expect a deterministic performance at any given point in time. For example, we expect that a WSN-based process automation on an oil rig operates reliably while the installation is cycling through the extreme temperature changes that are typically found in such deployments. A system failure caused by a wrong prediction of the impact of temperature changes on wireless communication is not acceptable.

Many studies describing experiences from WSN outdoor deployments have reported that diurnal (day/night) and seasonal (summer/winter) fluctuations of ambient temperature have a strong impact on communication quality. Lin et al. [1] have found a daily variation in the received signal strength (RSS) of up to 6 dBm, with the highest RSS values being recorded during night-time. Similarly, in their deployment in an Australian outdoor park, Sun and Cardell-Oliver [2] have measured on-board temperature daily variations between 10 and 50 °C, and noticed that links perform very differently between day and night. Also Thelen et al. [3] have noticed a drastic decrease of RSS at high temperatures in their potato-field deployment.

While the *macro-view* of the problem is clear (temperature has an effect on signal strength and link quality), this knowledge does not help us to fully understand the dependency between link quality and temperature. Furthermore, existing work does not allow us to predict the performance of a network with respect to communication-related temperature dependencies. The aim of this work is hence to develop a *micro-view* of the problem by analysing systematically the impact of temperature on different radio transceivers. We design a low-cost experimental infrastructure to vary the on-board temperature of nodes in a repeatable fashion and study the effects on transmitting and receiving nodes, isolating hardware-specific effects. Our results show that all platforms follow a similar trend that can be captured in a relatively simple first-order generic model for low-power wireless transceivers. Such a model can be used for planning and constructing wireless sensor networks providing dependable service despite temperature changes.

In the next section, we describe existing work in the outlined research area. In Sect. 3 we present results from a 1-year long outdoor deployment in Sweden that we used as a starting point for this work. We then describe and analyse the results of extensive lab experiments to systematically study the effects of temperature in a controlled setting. We develop a first-order model of temperature and link quality dependency in Sect. 4 and conclude our paper in Sect. 5.

2 Related Work

Results by Bannister et al. [4] from an outdoor deployment and from experiments in controlled scenarios have revealed that an increase in temperature causes a reduction in RSS. In their experiments in a climate chamber, the authors observe a linear decrease in RSS of about 8 dB over the temperature range 25-65 °C and show that this reduction may have severe consequences on the connectivity of a network. These results were confirmed by experiments by Boano et al. [5], [6], showing that one can safely decrease the transmission power of communications at low temperatures without deteriorating the performance of the network.

A recent long-term outdoor deployment by Wennerström et al. [7] has further shown that the average packet reception rate (PRR) in a WSN of 16 Tmote Sky nodes dropped by more than 30% when changing temperature from -5 to 25 °C, and that a clear degradation in PRR and average link quality occurred during summer, confirming that daily and seasonal fluctuations of ambient temperature have a strong impact on the quality of sensornet communications.

These existing works simply report the degradation of signal strength and link quality as a consequence of an increase in ambient temperature and do not provide a deeper analysis of the problem. In addition, every reported analysis is unique in terms of experimental setup and hardware. The used radio chips range from Nordic NRF903 [2] and CC1000 [3] to the popular CC1020 [6] and CC2420 transceivers [1], [7], making it difficult to separate general from hardware-specific effects.

Bannister et al. [4] have attempted to quantify the loss of RSS due to temperature changes, but only for a limited temperature range and for a single radio chip. Furthermore, when simulating the reduction of communication range and connectivity degradation due to an increase in ambient temperature, the authors assume that communicating nodes have similar temperatures.

This work goes beyond existing work and studies the impact of sender and receiver temperature on link quality systematically using different hardware platforms. After isolating hardware-specific effects, we show that temperature affects all platforms in a similar way and derive a model that captures its impact on low-power wireless transceivers.

3 Experimental Results

In order to get a deeper understanding of the impact of temperature on WSNs, we study the evolution of link quality over one year in an outdoor deployment in Sweden. Our analysis shows that temperature has a strong impact on communication, with visible daily and seasonal differences.

Building on top of these results, we carry out a large set of experiments in controlled settings, where we can repeat and alter the conditions at different nodes separately. In all our experiments, we analyse the impact of temperature by measuring the hardware-based link quality metrics in IEEE 802.15.4 compliant radio transceivers [8], namely the received signal strength indicator upon packet reception (RSSI) and in absence of packet transmissions (noise floor), and the link quality indicator (LQI)[1].

3.1 Long-Term Outdoor Deployment

We now describe the impact of temperature on communication that we have observed in our outdoor deployment at a Swedish meteorological station spanning over a whole year.

Experimental Setup. We have deployed a sensor network comprising 16 TelosB sensor nodes outside Uppsala, Sweden, in an open field isolated from human activity and absence of electromagnetic interference. Sensor nodes are mounted on poles along a 80 meter straight line at intervals of 0, 20, 40, and 80 meters: on each pole, two nodes are mounted at 0.5 and 1.5 meters height, respectively. The nodes are powered via USB and attached to a Sensei-UU testbed [9], ensuring reliable and continuous data logging.

The software running on the sensor nodes periodically sends packets between every possible pair of nodes and works as follows. Each node is assigned the sender-role in a round-robin fashion every 30 seconds. During this phase, the designated sender transmits one packet per second addressed

[1] Please notice that the RSSI readings from all sensor nodes employed in our experiments are uncalibrated.

	1 year	1 month	1 day	1 hour
Lowest temp. (°C)	-22.2	-3.0	7.2	21.2
Highest temp. (°C)	61.3	63.7	63.8	55.9
Temp. difference	82.5	66.7	56.6	34.9

Table 1: Largest temperature variations on a single node as seen in our outdoor deployment.

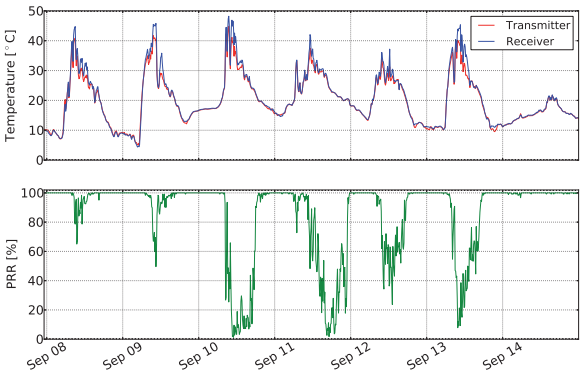
to each of the other nodes, again in a round-robin manner. When a packet is received by the intended recipient, a response packet addressed to the sender is sent. Each time a sensor node receives a packet – including when it is not the intended recipient – it logs several statistics about the received packet, namely RSSI, LQI, and noise floor. On-board ambient temperature is measured on each node every two seconds using the on-board SHT11 temperature sensor. More details on the experimental setup can be found in [7].

Impact of temperature on PRR.

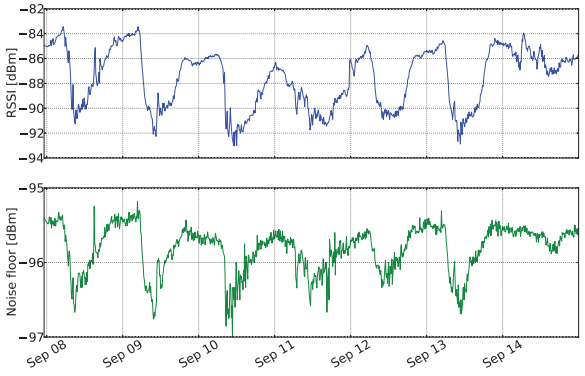
To highlight the impact that ambient temperature has on the links deployed in our outdoor WSN, we focus on a specific link, close to the edge of the communication range. Fig. 1(a) (top) shows the temperature of two nodes (transmitter and receiver) forming a unidirectional link during a week in September. Temperature varies as much as 40 °C between day and night since sensor nodes are enclosed into air-tight enclosures and exposed to direct sunlight. Therefore daily temperature fluctuations may cause a combined overall variation between the two nodes of up to 80 °C. Although the highest variations occur over the 24-hours, temperature can fluctuate by as much as 34.9 °C within one hour, as we show in Table 1, in which we summarize the largest temperature ranges observed in our 12-months deployment for different time intervals.

Fig. 1(a) (bottom) further shows that each substantial increase in temperature (typically occurring during daytime) results in a decrease in PRR, leading to an almost complete disruption of the connectivity between the two nodes.

Impact of temperature on RSSI and noise floor. The decrease in PRR is strongly correlated with a decrease in the RSSI computed over the received packets, as shown in Fig. 1(b) (top), hinting that the change in temperature – and not external interference – was the cause of the packet loss. In particular, the RSSI fluctuates between -84 and -92 dBm, the latter being

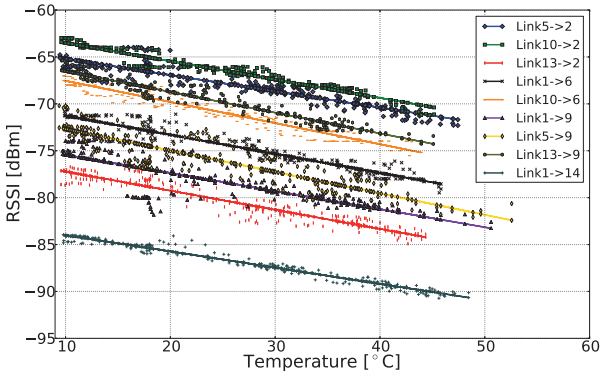


(a) Temperature and PRR

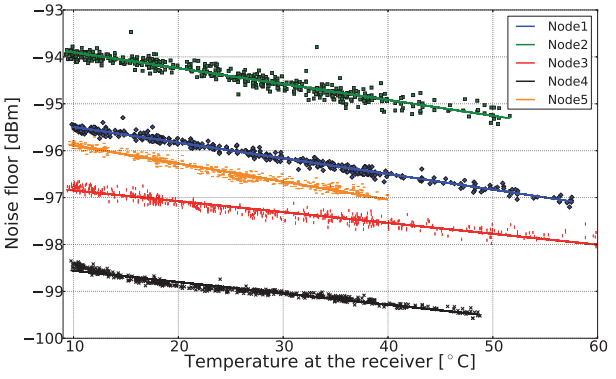


(b) RSSI and Noise floor

Figure 1: Temperature has a strong impact on the quality of links in our outdoor WSN. During daytime, when temperature is high, there is a significant reduction in PRR (a). Also the trend of RSSI and noise floor resembles the one of temperature, with a sharp decrease when temperature increases (b).



(a) RSSI



(b) Noise floor

Figure 2: The relationship between RSSI and temperature (a) and between noise floor and temperature (b) can be approximated as a linear function, and the trend is similar for different nodes.

the threshold below which no packets are received. Interestingly, also the noise floor follows a trend similar to the RSSI and decreases as temperature increases, but to a much lower extent, as shown in Fig. 1(b) (bottom).

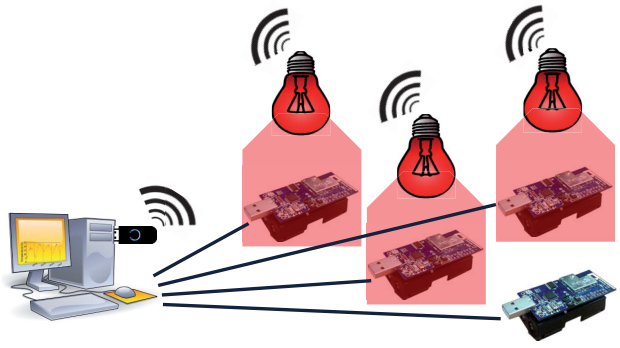
The strong correlation between temperature, RSSI, and noise floor is highlighted in Fig. 2(a) and 2(b), respectively. Fig. 2(a) shows the RSSI and the combined temperature of sender and receiver for nine links with different link quality over a timespan of three days. The relationship between temperature and RSSI can be approximated as a linear function and is clearly visible despite the intrinsic noise produced by long-term measurements. Using linear regression we have observed that different links have a similar trend, with an average slope of -0.205 and a standard deviation of 0.026.

Fig. 2(b) shows the noise floor of five nodes over the same 3 days. Also in this case, the relationship with temperature is approximately linear, with a similar slope among different nodes, but with a less pronounced decrease compared to RSSI (average slope of -0.034 $\pm$ 0.006).

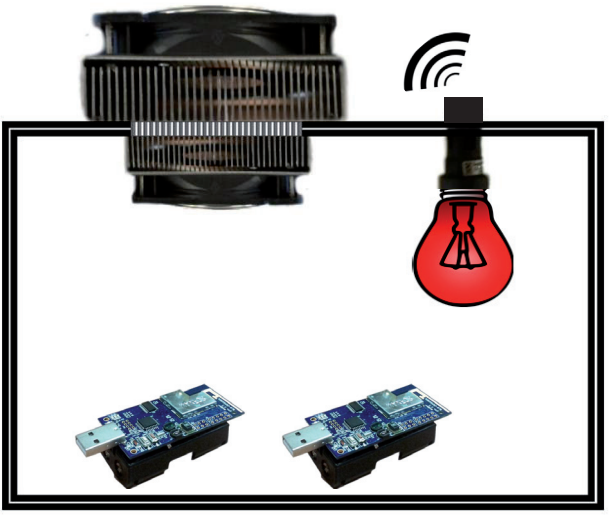
3.2 Controlled Testbed Experiments

To get a deeper understanding of the effects observed in Sect. 3.1, we have augmented an existing sensornet testbed with the ability of varying the on-board temperature of sensor motes and *reproduce the impact of temperature on link quality in a repeatable fashion.* We use this low-cost testbed infrastructure to systematically study the impact of temperature on different hardware platforms and to isolate the effects of temperature on transmitting and receiving nodes.

Experimental Setup. Fig. 3(a) shows an overview of our controlled experimental setup. We have extended an existing WSN testbed with the ability of varying the on-board temperature of sensor motes in the range -5 to +80 °C using infrared light bulbs placed on top of each sensor node. The light bulbs can be remotely dimmed using the 868 MHz frequency, and hence their operations do not interfere with the communications between the wireless sensor nodes, as the latter use the 2.4 GHz ISM band. In order to cool down the motes below room temperature, we have built custom Polystyrene enclosures as shown in Fig. 3(b), in which, in addition to the light bulb, a Peltier air-to-air assembly module by Custom Thermoelectric cools the temperature down to -5 °C when the enclosure is kept at room temperature and the light bulb is off. As we only have a limited number of Peltier enclosures, some of the nodes in the testbed are only warmed



(a) Setup overview



(b) Sketch of a Peltier enclosure

Figure 3: Experimental setup in controlled testbed experiments.

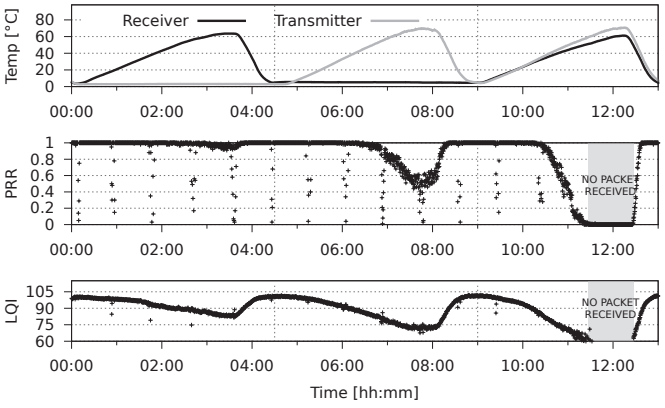
by the infrared light bulbs between room temperature and their maximum operating temperature range.

Our testbed is composed of Maxfor MTM-CM5000MSP and Zolertia Z1 nodes employing the CC2420 radio [10], as well as of Arago Systems WisMotes employing the CC2520 transceiver [11]. Sensor nodes are divided in pairs and form bidirectional links operating on different physical channels to avoid internal interference. All sensor nodes run the same Contiki software: each sensor node continuously measures the ambient temperature and relative humidity using the on-board SHT11 or SHT71 digital sensors, and periodically sends packets to its intended receiver at a speed of 128 packets per second using different transmission power levels. Statistics about the received packets are logged using the USB backchannel and are available remotely.

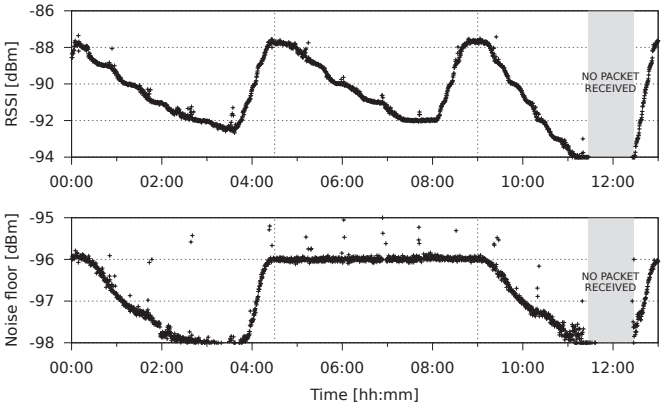
Validation of our controlled setup. Using our controlled testbed setup, we are able to reproduce the impact of temperature on link quality in a very fine-grained way. In a first experiment using Maxfor nodes, every link in the testbed is exposed to three heat cycles. First, each individual node, i.e., first the transmitter and then the receiver, is heated from 0 up to 65 °C. Afterwards, both nodes are heated in the same temperature range at the same time. Fig. 4(a) illustrates the impact of temperature on PRR and LQI on a particular link. The evolution of temperature at the transmitter and at the receiver over the 13-hours experiment is shown in the top figure. In correspondence to each increase of temperature, PRR and LQI decrease significantly, with the highest impact occurring when both nodes are heated. With both nodes heated, indeed, no packet was received and the connectivity between the two nodes was interrupted until the temperature started to decrease. Fig. 4(a) also shows that the packet loss rate is more pronounced when the transmitter is heated compared to the case in which only the receiver is heated, something that we have observed in the majority of links in our testbed.

Fig. 4(b) illustrates the impact of temperature on RSSI (top figure) and noise floor (bottom figure). The RSSI decreases in a similar way when transmitter and receiver are heated separately, whereas the decrease is more pronounced if both transmitter and receiver are heated at the same time. This proves that temperature decreases both the transmitted and received power [4], whereas the noise floor only decreases when the receiver node is heated, with an absolute variation smaller than the one of RSSI.

These results hence prove the validity of our setup and confirm the measurements obtained in our outdoor deployment, quantifying precisely the



(a) PRR and LQI



(b) RSSI and Noise floor

Figure 4: Impact of temperature on the quality of links in our controlled testbed. We heat transmitter and receiver nodes separately first, and then both of them at the same time. When temperature increases, PRR, LQI, and RSSI decrease significantly, with the highest impact occurring when both nodes are heated at the same time. The periodic noise is due to a Wi-Fi access point beaconing in proximity of the testbed.

impact on temperature on each individual node. We now derive a set of observations obtained running experiments using the same experimental setup, i.e., three heat cycles in which each node is heated individually first and then both nodes are heated at the same time, on different hardware platforms.

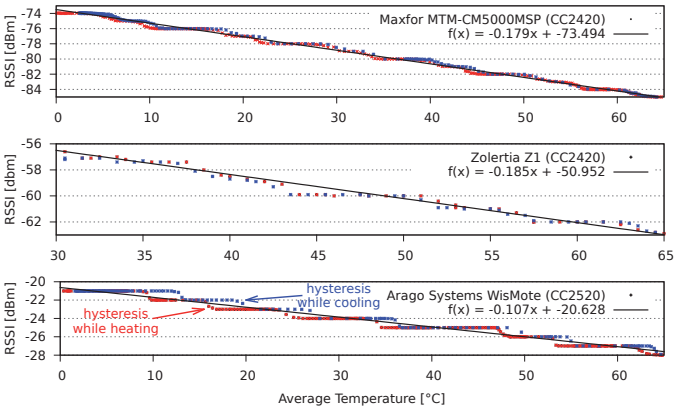
The decrease in RSSI is consistent among different platforms. The trend observed in our outdoor deployment showing that RSSI decreases in an approximately linear fashion with temperature holds for different platforms and different radio chips, but with a different slope. Fig. 5(a) shows the relationship between RSSI and temperature obtained on different platforms when heating both nodes at the same time. The hardware platforms employing the same CC2420 radio exhibit approximately the same slope.

The decrease in RSSI does not depend on how quickly temperature changes. In our setup, the heat cycles are characterized by a slow increase in temperature followed by a quicker cooling phase, as can be seen in Fig. 4(a). This allows us to observe that both RSSI and noise floor are not affected by how quickly temperature varies. Hence, the impact of temperature can be modelled using the absolute temperature value at the transmitter and receiver nodes.

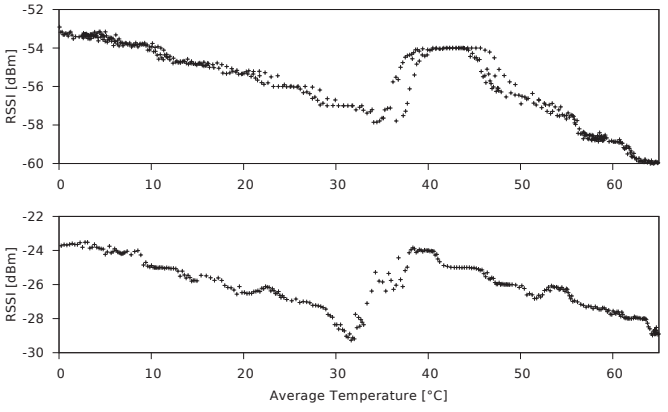
Discrete steps. On close inspection in Fig. 5(a), one can observe discrete steps in the relationship between RSSI and temperature. For the CC2420 platforms, the size of the prominent steps is 2 dBm, whereas for platforms employing the CC2520 radio the step is 1 dBm large. Bannister [12] has attributed the loss of RSSI to the loss of gain in the CC2420 Low Noise Amplifier (LNA). Our experiments bring further evidence to strengthen this claim, as there are references to 2 dBm steps in the CC2420 datasheet [10] with regard to the operation of the Automatic Gain Controller (AGC).

Hysteresis. Fig. 5(a) also shows an hysteresis in the relationship between RSSI and temperature that can be seen comparing the RSSI curve obtained when heating and when cooling down the motes. As for the discrete steps, the hysteresis also can be attributed to the operation of the AGC in the CC2420 radio. According to the CC2420 datasheet, hysteresis on the switching between different RF front-end gain modes is set to 2 dBm [10].

Non-linearity in the CC2420 curve. In our experiments, we have also noticed visible non-linearities when the RSSI is $\approx$ -28 and -58 dBm in the CC2420 platform, as shown in Fig. 5(b). These non-linearities were also



(a) Loss in RSSI when temperature changes



(b) Non-linearities in the CC2420 radio

Figure 5: Figure (a) shows that the relationship between RSSI and temperature is similar when using different hardware platform and can be approximated as a linear function, but with different parameters. Figure (b) shows the non-linearities in the response of the CC2420 radio measured using Maxfor nodes. Temperature on the x-axis is computed as the average temperature of the transmitter and receiver temperature.

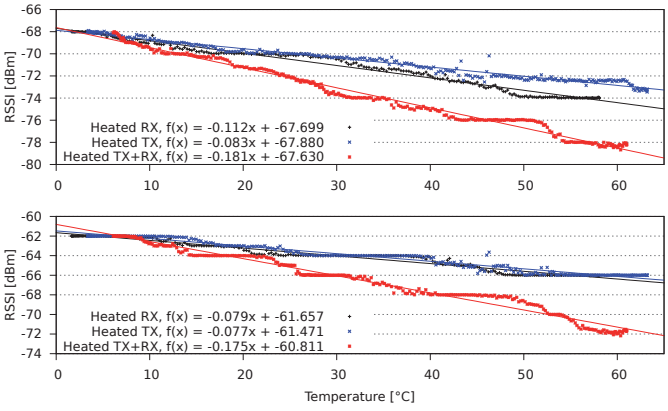
measured by Chen and Terzis [13], and may lead to a false approximation in case the RSSI of the considered link falls *exactly* in this region (as in the experiments of [4]). When deriving our linear approximation for the CC2420 transceiver, we hence do not consider links falling in this range.

RSSI loss on transmitter and receiver. Fig. 6(a) shows the relationship between RSSI and temperature obtained on Maxfor nodes when transmitter and receiver nodes are heated individually and when both nodes are heated at the same time. Top and bottom figures refer to the same link, but are obtained using a different transmission power. Despite the link is the same, the relationship between RSSI and temperature is slightly different, with a steeper decrease when the receiver is heated in the top figure. Although a comparison between curves is difficult due to the AGC operations (depending on whether we capture the transition between two discrete steps, we may obtain slightly different slopes), by averaging the data from all our experiments we have obtained a relationship between receiver and transmitter of 0.5348 ± 0.061. The RSSI seems hence to have a slightly steeper slope when the receiver node is heated.

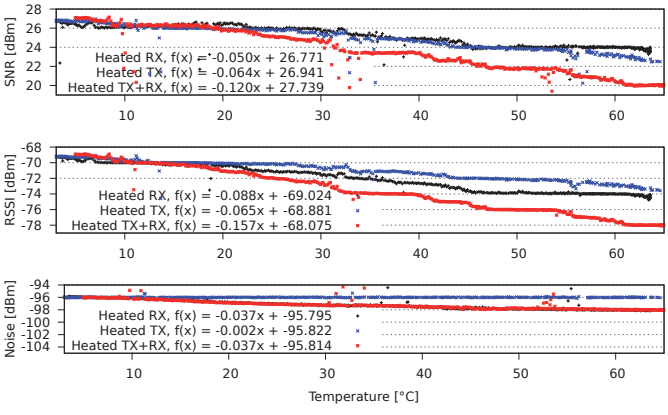
Impact on noise floor and SNR. Fig. 6(b) illustrates how noise floor, RSSI, and signal to noise ratio (SNR) vary on a given link when transmitter and receiver nodes are heated individually and at the same time. Since the noise floor decreases only when the receiver is heated, an increase in temperature on the transmitter has an higher impact on the SNR compared to an increase in temperature at the receiver. This also explains the different impact in PRR when heating the nodes individually that we observed in Fig. 4(a).

4 Platform Models

The effect of temperature on electric conductors and semiconductors is well known. Various models have been created for a large range of devices to capture the relation between ambient temperature and electric conductance (and current leakage). Our goal is to build on top of this knowledge to create a generic model for low-power radio transceivers. It is important to remark that the goal of our model is not to benchmark a specific radio chip against others, as this is already done by manufacturers. Our goal is to develop a simple model to predict the performance of a network under extreme environmental settings. We now describe the overarching effect of



(a) Loss in RSSI when using different TX powers



(b) Loss in noise floor, RSSI, and SNR for a given link

Figure 6: Relationship between RSSI, noise floor, SNR and temperature when transmitter (blue) and receiver (black) nodes are heated individually, and when both nodes (red) are heated at the same time.

temperature on radio transceivers and derive a generic model for low-power wireless transceivers.

4.1 The effect of temperature on RSS

In electric conductors, a higher temperature increases the resistance of the medium, whereas in semiconductors it leads to current leakages. In practice this means that, for a given voltage, a higher temperature reduces the current and hence the power of a device. In radio transceivers, these phenomena imply that a raise in temperature will reduce the SNR. A decrease in SNR leads to a lower link quality and a shorter radio link, which in turn may lead to lower throughput, higher delay or even network partitioning. Hence, our goal is to model the effect of temperature on SNR. Denoting PL as the path loss between a transmitter-receiver pair, P_t as the transmission power, P_r as the received power, and P_n as the noise floor at the receiver, the SNR is known to be:

$$\begin{aligned} SNR(dB) &= P_t - PL - P_n \\ &= (P_t - P_n) - (P_t - P_r) \end{aligned} \tag{1}$$

As we have shown in our empirical measurements, an increasing temperature has 3 main effects on the signal strength of radio transmissions; it (i) decreases the transmitted power, (ii) decreases the received power, and (iii) decreases the noise floor. We now model these three effects in Eq. 1.

4.2 A first-order model

Denoting α, β, γ as constants with units dB/K, and T_t, T_r as the temperature in Kelvin of transmitter and receiver, the effect of temperature on SNR can be defined as:

$$\begin{aligned} SNR &= (P_t - \alpha\Delta T_t) - (PL + \beta\Delta T_r) \\ &\quad -(P_n - \gamma\Delta T_r + 10\log_{10}(1 + \tfrac{\Delta T_r}{T_r})) \\ &= P_t - PL - P_n - \alpha\Delta T_t \\ &\quad -(\beta - \gamma)\Delta T_r - 10\log_{10}(1 + \tfrac{\Delta T_r}{T_r}) \end{aligned} \tag{2}$$

The proportional relation between ΔT and the constants α (effect on transmitted power), β (effect on received power) and γ (effect on noise floor) is based on the empirical observations made in the previous sections. The term $10\log_{10}(1 + \frac{\Delta T_r}{T_r})$ is derived analytically from the well-known thermal equation. There are two important trends to highlight in this model. First, changes in temperature have a higher impact on the transmitted and received

powers (linear relation of α and β), than on the thermal noise (logarithmic relation). Second, to some extent it is counter-intuitive that a higher temperature decreases the noise floor (negative sign of γ). This effect was also observed by Bannister, and he hypothesizes that it is due to the losses in the signal amplifier [12]. That is, a higher temperature not only reduces the gain of the signal but also the gain of the noise, and hence, the received signal strength (RSSI) is lower for both.

The accuracy of our model depends on identifying the right values for α, β and γ. In our case, these parameters are given by the slopes of the linear trends observed in our empirical results. These parameters are platform dependant, and hence require a systematic and fine-grained evaluation. Our testbed was designed to accomplish exactly that. For example, a network manager willing to deploy a network using the Maxfor platform, can use the slopes obtained in Fig. 6(b): $\alpha = 0.065$, $\beta = 0.088$ and $\gamma = 0.037$. Assuming that the network will be deployed in an environment where the maximum and minimum day temperature are 50 and 5°C respectively, the network manager can predict that the links can suffer an attenuation of $(\alpha + \beta - \gamma)\Delta T = 5.22$ dB (5 dB according to the SNR measurements in Figure 6(b) top). This level of attenuation can easily push a good link (with 100% PRR) to have a PRR of 0%.

5 Summary and Outlook

The central tenet of our study is that the important role played by ambient temperature in the performance of sensor networks can (and must) be analysed in a systematic way. Motivated by initial studies focusing on single platforms, we use a low-cost yet precise testbed to show that most platforms have similar intrinsic characteristics that can be easily modelled. Our results capture with good accuracy how temperature affects the signal strength in transmitters and receivers. A thorough understanding of the effect of temperature on low-power wireless links is a first necessary step of a much broader goal: the ability to predict the performance of sensor networks in various environmental settings.

References

[1] S. Lin, J. Zhang, G. Zhou, L. Gu, T. He, and J. A. Stankovic. ATPC: Adaptive transmission power control for wireless sensor networks. In *Proc. of the 4th SenSys*, 2006.

[2] J. Sun and R. Cardell-Oliver. An experimental evaluation of temporal characteristics of communication links in outdoor sensor networks. In *Proc. of the 2th RealWSN*, 2006.

[3] J. Thelen et al. Radio wave propagation in potato fields. In *Proc. of the 1st WiNMee*, 2005.

[4] K. Bannister, G. Giorgetti, and S. K. S. Gupta. Wireless sensor networking for hot applications: Effects of temperature on signal strength, data collection and localization. In *Proc. of the 5th HotEmNets*, 2008.

[5] C.A. Boano, J. Brown, N. Tsiftes, U. Roedig, and T. Voigt. The impact of temperature on outdoor industrial sensornet applications. *IEEE TII*, 6(3), 2010.

[6] C.A. Boano, J. Brown, Z. He, U. Roedig, and T. Voigt. Low-power radio communication in industrial outdoor deployments: The impact of weather conditions and ATEX-compliance. In *Proc. of the* 1^{st} *Sensappeal*, 2009.

[7] H. Wennerström et al. A long-term study of correlations between meteorological conditions and 802.15.4 link performance. In *Proc. of the 10th SECON*, 2013.

[8] N. Baccour et al. Radio link quality estimation in wireless sensor networks: a survey. *TOSN*, 8(4), 2012.

[9] O. Rensfelt et al. Sensei-UU: a relocatable sensor network testbed. In *Proc. of the 5th WiNTECH*, 2010.

[10] Texas Instr. *CC2420 datasheet, revision SWRS041c*, 2013.

[11] Texas Instr. *CC2520 datasheet, revision SWRS068*, 2007.

[12] K. Bannister. Impacts of thermal reduction in transceiver performance on outdoor sensing networks. Master's thesis, Arizona State University, Phoenix, AZ, USA, 2009.

[13] Y. Chen and A. Terzis. On the mechanisms and effects of calibrating RSSI measurements for 802.15.4 radios. In *Proc. of the 7th EWSN*, 2010.

Paper III

Paper III
Transmission Errors in a Sensor Network at The Edge of The World

Hjalmar Wennerström, Liam McNamara, Christian Rohner, and Lars-Åke Nordén *Transmission Errors in a Sensor Network at The Edge of The World* In Proceedings of the 5th Extreme Conference on Communication (ExtremeCom), August 2013, Thorsmork, Iceland.

Transmission Errors in a Sensor Network at The Edge of The World

Hjalmar Wennerström, Liam McNamara,
Christian Rohner, Lars-Åke Nordén

Department of Information Technology, Uppsala University, Sweden

Abstract

The performance of an outdoor wireless sensor network for remote monitoring is dictated by its ability to deliver sensed data to a sink. The often isolated location means that radio interference is typically low and with different patterns in transmission errors compared to an indoor deployment. To better understand these processes, we investigate packet errors in such a network. Specifically, we study characteristics of decoding errors within 802.15.4 transmissions.

We describe the experimental deployment of an outdoor sensor network, located above the Arctic circle, where we log both successfully received and broken packets. Results indicate that a substantial amount of received packets contain errors, where the number of errors in each packet are typically few. We distinguish between transmission errors and payload errors, and find that transmission errors are equally probable over all positions of a packet, whereas bit errors in the payload are not. This results in some bits having a 25% higher chance of being corrupted than others.

1 Introduction

Real deployment environments generally possess different characteristics than the theoretical one that the technology was designed for. Indeed, it is often most important only to design technology so that it works sufficiently well in a majority of situations. Outdoor wireless sensor networks (WSN), used to collect environmental data of the surroundings, can be deployed to monitor isolated sites that otherwise would not be possible to study as accurately. Such remote sites are likely to experience low radio interference

due to their distance from civilization (as compared to an office environment with many interfering wireless technologies).

Generally, researchers emulate low interference environments in size-constrained anechoic chambers, with nodes using low power transmissions as a substitute for large communication distances (or sometimes simply shifting to a quiet channel). Such techniques do not necessarily reflect a true spatially distributed wireless network in a low interference environment. Moreover, the IEEE 802.15.4 specification stipulates differing timings depending on the transmission power [4]. Real outdoor deployments may also experience a much wider range of environmental conditions than can be simulated in an indoor lab. Understanding how actual communication errors occur in such a low interference scenario can provide insights into choosing appropriate mitigation strategies, typically either Automatic Repeat Request (ARQ) or Forward Error Correction (FEC). Selecting and configuring the most appropriate strategy depends on trade-offs in application and energy requirements versus the exact link quality in terms of its error characteristics.

In this paper we study the characteristics of transmission errors in an outdoor 802.15.4 WSN located in the Arctic region. Specifically, we study communication where the Packet Reception Ratio (PRR) varies over time, examining unstable links. We capture correctly received packets as well as broken packets (containing errors in the payload) to gain a deeper appreciation of their occurrence. The goal is to understand a real deployment's prevalence of packet errors and how they are corrupted in the absence of external interferers.

The results show that, in this low interference scenario, there are very few errors per packet and that the errors occur equally over all positions in the transmitted chip sequences but not in the resulting payload bits. We see that at intermediate levels of PRR (>40%) a majority of packets may still be received despite containing errors. We also observe that there are times when no packets are received correctly but where up to 40% of sent packets are still received with errors. Looking at the specific bit errors we found that some bits have a 25% higher chance of being corrupted than others. Additionally, 90% of the corrupted bits co-occur with at least one more corrupted bit.

2 Deployment and Experimental Setup

Deploying an outdoor sensor network above the Arctic circle requires some extra thought and precautions to maintain operability. The purpose of the deployment is to investigate sensor network communication, specifically 802.15.4 and the unique characteristics of such a distinct environment on the radio communication. In this section we give some insights into the deployment's environment, how to make the hardware robust against the extreme climate and details on the experimental setup.

2.1 Environment and Deployment

We setup a sensor network at a polar research facility in Abisko, Sweden ($68°21'N, 18°49'E$), which annually has an mean temperature around 0°C and only 300 mm of precipitation. It experiences permanent daylight in the summer and an absence of the sun during the winter. The research station monitors Arctic ecology, meteorology and climate change. The surrounding area is a national park, with 40% of the ground above the tree line and patches of permafrost. Specifically, the network is deployed in an area characterized by a mixture of open space grassland and occasional mountain birch. Except for a few summer months the site is covered by thick snow and ice. Due to the isolated location, wireless interference is practically non-existent.

2.2 Experimental Setup

The network is comprised of 12 sensor nodes, mounted on four poles aligned along a 80 m straight line at distances of 0 m, 20 m, 40 m and 80 m. Nodes are mounted on each pole at heights of 1.5 m, 1.0 m and 0.3 m. These heights were chosen so that the lowest node would potentially be covered in snow, the middle node just above the snow and the high mounted node to ensure good radio communication. So that the setup would withstand the harsh environment, all the equipment was placed in enclosures with IP68 connectors to avoid any water damage.

The schematic setup, seen in Figure 1, shows the main components of the deployment. The twelve TelosB nodes [1] (Figure 1 A) are attached to our Sensei-UU testbed [7] via a USB backplane, ensuring both reliable data logging and power supply. The sensor nodes, which are among the most commonly used by the research community, are equipped with 802.15.4-

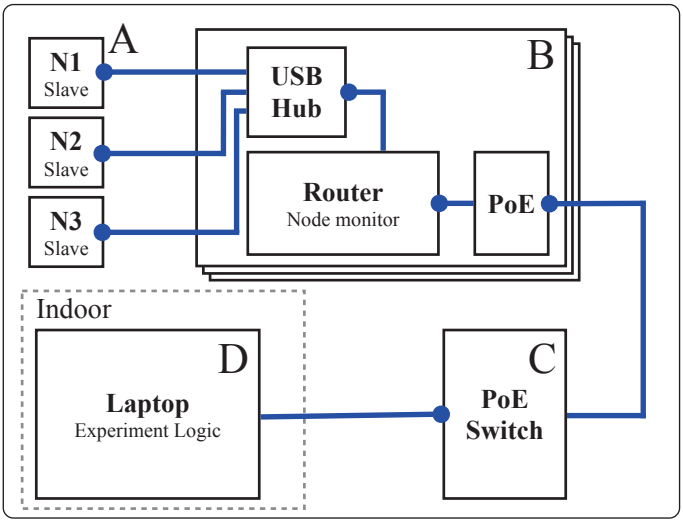


Figure 1: The four main components in the system design where nodes are marked as N1, N2 and N3

compatible CC2420 radio transceivers that operate in the 2.4 GHz ISM band [9]. At each pole we have a testbed box (Figure 1 B) containing a router, USB-hub and Power-over-Ethernet (PoE) splitter. The router is used to control and monitor the nodes, sending commands and handling node resets. The box and its three nodes is powered using a single Ethernet cable, reducing the cabling complexity. Figure 2 shows how it looks like when deployed.

Each of the four boxes are connected to a PoE switch (Figure 1 C) that is also located outside in the snow. The switch is connected to a laptop indoors (Figure 1 D) that runs the experiment logic. The setup is designed to be easy to deploy and to ensure continuous operations over long periods of time, that is also the reason that the nodes run on fixed power.

2.3 Software Logic

Our experiment consists of cyclically sending packets along each radio link, i.e., sending packets between every possible pair of nodes. The scheme works in a round-robin fashion where each node takes turns being the designated

Figure 2: Three of the mounted nodes and a testbed box on the ground.

sender, a role that changes every 30 seconds. The sender then transmits one packet per second addressed to each of the other nodes, again in a round-robin manner. When a packet is received by the intended recipient, that node replies with a response packet addressed to the sender. Throughout this process, all other nodes promiscuously receive any packets they overhear and log them accordingly, in effect generating up to 11 packet receptions per sent packet.

To capture 802.15.4 packets containing errors in the payload (i.e., packets whose CRC fails) we have modified the radio driver of the nodes. This allows us to log all packets that have successfully been detected. The payload of each packet is 34 bytes in size and contains, except a two byte start sequence, the fields: packet type, sequence number, source address and destination address, that are repeated four times in sequence until the packet buffer is full. The repeated pattern in the payload is used during analysis of broken packets to compute the ground truth. Through a majority vote of the the same piece of received data we can deduce what it should be, and identify errors with that information. Packets where we are not able to find consensus (due to many errors) in the voting are disregarded, we note that such packets are very rare ($<0.1\%$) among all broken packets.

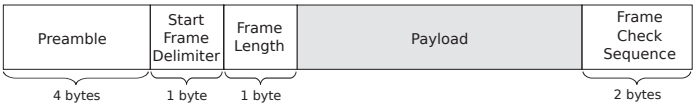


Figure 3: The anatomy of a 802.15.4 packet

3 Data Transmission in 802.15.4

IEEE 802.15.4 is a low-rate wireless protocol intended for embedded device communication and is commonly used in low power wireless sensor networks. In this section we give a brief introduction and highlight technical issues that are relevant when analysing the results. More details can be found in the specification [4].

3.1 Encoding

The radio uses direct-sequence spread spectrum modulation where each sent byte is divided into two four-bit *symbols*. Each of the symbols, with a value between 0 and 15, are then each mapped to a 32-bit pseudo-random *chip sequence* for transmission. A receiver maps the received 32-bit chip sequence to the closest matching valid sequence, which corresponds to a four-bit symbol. The 32-bit chip sequences are designed with a high Hamming distance between them, allowing for better symbol matching in the presence of chip errors due to interference.

The chip sequence to symbol conversion is done in hardware, meaning that without major modifications one cannot obtain the 32-bit chip sequence a sensor node received but only the symbol to which it was mapped. Thus, there is no direct measure of the number of chips that were flipped during transmission, only retrospective knowledge of which symbols were corrupted.

The 32-bit chip sequences are designed in such a way that the first eight symbols are represented by the same chip sequence only shifted four bits, these correspond to symbols 0 thorough 7. The second half of the chip sequences (8 through 15) are made by inverting every other bit of the sequences from the first half, e.g., the sequence for symbol 8 is the sequence for symbol 0 but with every other bit flipped. This scheme, as we will show, has consequences in the way errors manifest.

3.2 Packets

A 802.15.4 packet, seen in Figure 3 consists of a four byte preamble, used to detect and synchronize a transmission. The preamble is followed by a one byte Start Frame Delimiter (SFD) that is used to indicate the start of a frame. If a receiver is unable to successfully decode the preamble or misses the SFD it will ignore the rest of the packet.

Once the receiver has synchronized using the SFD it then reads the next byte indicating the length of the packet. This is followed by the payload of the specified length. The last two bytes contain the Frame Check Sequence (FCS), which is a checksum of the payload. The FCS is used for error detection by computing the cyclic redundancy check (CRC). Packets that fail the CRC contain errors and are usually disregarded and consequently lost to the receiver. However in this work we pay special attention to those packets where the CRC fails.

Packets are lost either when the receiver fails to synchronize with the SFD or the CRC fails, and in order to combat this packet loss there are two standard approaches. The first one is to use ARQ in which the receiver indicates a lost packet, by (not) sending an acknowledgement, and the sender simply retransmits the whole packet. The other option is to use a FEC scheme that is able to detect and correct a given number of corrupted bits through the proactive inclusion of redundant information. The choice of FEC scheme decides the number of errors one can detect and correct, at the cost of more overhead, therefore the choice is heavily influenced by the error characteristics of the transmission.

4 Results

We chose to study the error patterns from a subset of links in our deployment, specifically links that are on the edge of the communication range and exhibit a changing PRR over time. The data presented in this section was collected during three weeks between April 2nd and 21st, 2013. The links span 40 m between sender and receiver where nodes are mounted either at 1.0 m or 0.3 m.

4.1 Packet Errors

Firstly, we separate packet receptions into three categories. Packets that were *successful* in their reception and decoding, packets that were received but the decoding was *broken* due to error(s) and packets that were com-

	Nr of packets	Percent of total
Successful	218,638	35.9%
Broken	86,566	14.2%
Missed	304,175	49.9%

Table 1: The number of packets over the studied links during April 2nd to 21st for the three different categories.

pletely *missed*. The number of packets in each category for the chosen links during the entire experiment can be seen in Table 1.

Stemming from these three categories we compute one metric for each of them, where we use PRR to measure the amount of successfully received packets. To compare PRR to the amount of received broken packets we define the Error Packet Reception Ratio:

$$EPRR = \frac{\#\ recived\ packets\ with\ error}{\#\ sent\ packets}$$

as the ratio between packets with corrupted payload and sent packets. From that we use a modified version of the metric Packer Error Rate and define Packet Missed Rate:

$$PMR = 1 - (PRR + EPRR)$$

as the ratio between sent and never received packets. From the definitions it holds that the sum of PRR, EPRR and PMR computed over the same time window is always 1.

An example of these three metrics for a specific link is shown in Figure 4. Here we see that even when PRR drops packets are still received, only with an increasing percentage having a corrupted payload. Note that there are periods when the EPRR is as high as 60%, showing that sometimes more than half of the packets are received but broken. The relationship between the three metrics plotted against each other gives some additional insights. Figure 5 shows EPRR and PMR for different values of PRR. We see that when the link is strong (PRR above 80%) the sent but not successful packets are for the most part received but contain errors. At very low levels of PRR on the other hand there is a larger range of values of EPPR and PMR. Interestingly, there are times when the PRR is 0 yet the EPRR is as high as 40%. This suggests that there are periods when the only available information is from broken packets, and making use of these becomes crucial in order to maintain operations. In addition we see in Figure 5 that the

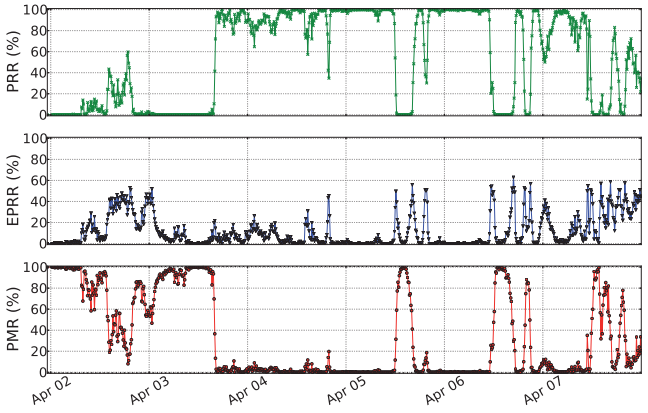


Figure 4: The three metrics for six days of the experiment, each computed over 10 minute intervals.

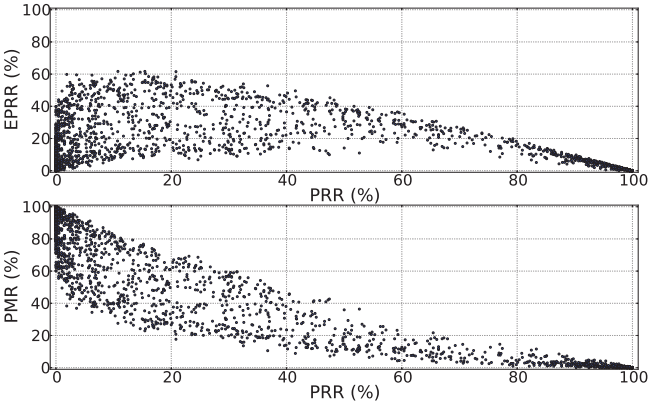


Figure 5: PRR vs. EPRR and PMR. EPRR can be as high as 60% when PRR is only at 10%. Again each point is a measurement over 10 minutes.

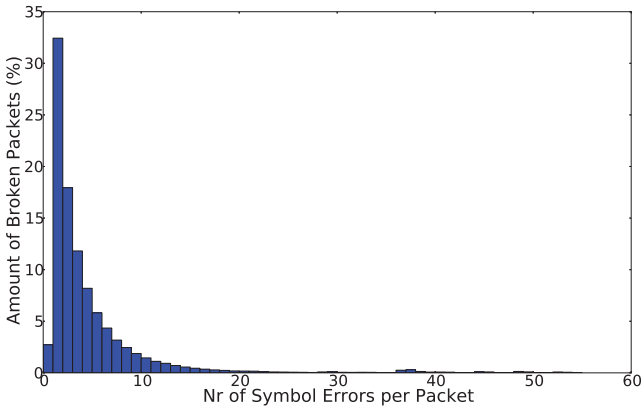


Figure 6: The amount of broken packets with different number of broken symbols in the payload.

spread of EPPR and PMR can be as much as 50% for a given value of PRR. This may suggest that there are times when packets get corrupted in different ways, and as a result either gets missed or received with errors. As PRR drops, the number of errors in the transmission increases making the probability of an error in the preamble (Figure 3) higher, resulting in more missed packets.

4.2 Symbol and Bit Errors

Errors in received packets are due to the 802.15.4 32-bit chip sequences being matched to the wrong symbol, resulting in a failed CRC. Since the chip sequences are matched to symbols in hardware, the only information we can obtain is the resulting symbols. We can use individual symbol errors as a more accurate metric when looking at packet errors, counting the number of errors in each packet and measuring their distribution.

The fact that most received broken packets only contain a small number of symbol errors in the payload can be seen in Figure 6. Here we show the distribution of the number of symbol errors per broken packet. Packets with two or fewer symbol errors in the payload make up more than 50% of all broken packets. Note that the packets with zero symbol errors are packets

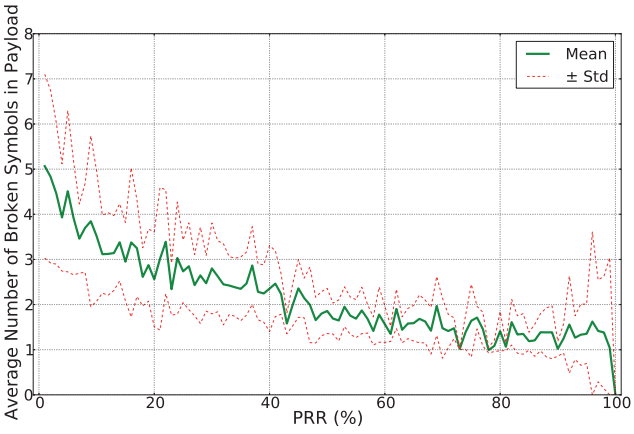
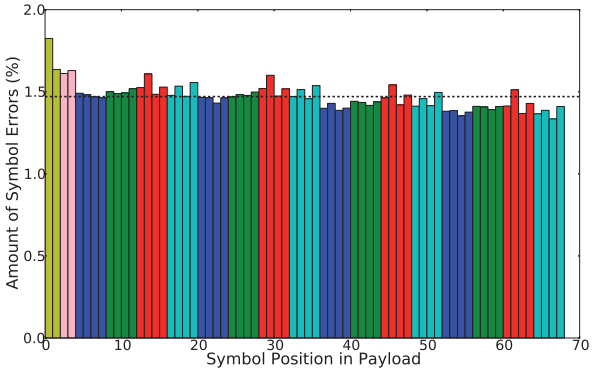


Figure 7: PRR vs. average number of symbol errors per broken packet. Only below 50% PRR does the average number of broken symbols go above two.

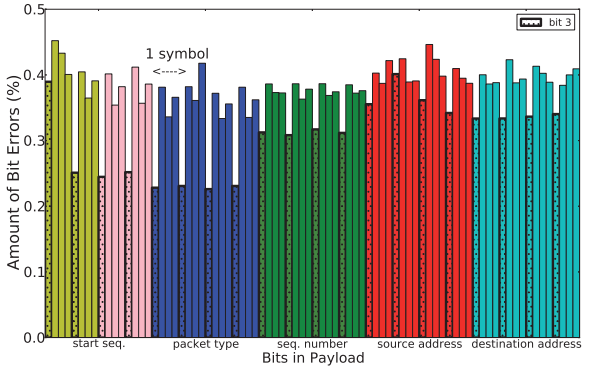
where the FCS (Figure 3) was corrupted, which is not part of the payload.

Looking at the average number of broken symbols for different values of PRR in Figure 7, we see that for high values of PRR it approaches the minimum of one broken symbol per broken packet. At 60% PRR the average number of broken symbols per broken packet is about 1.5. The occasional high number of broken packets at very high PRR values, resulting in a high standard deviation in Figure 7, is due to there only being very few broken packets, which by chance have many errors.

To further understand the occurrence of symbol errors in 802.15.4 packets, we compute the position of broken symbols within each packet. Figure 8(a) shows amount of errors for each symbol position, computed for all broken packets over the entire experiment. The horizontal dashed line indicates the level where packet errors are equally probable in all positions. As expected, symbol errors occur relatively evenly over all positions within transmissions, this is due to the fact that there is no radio interference. The slight decreasing tendency at later symbol positions comes from the fact that the length field (Figure 3) sometimes gets corrupted and fewer than 34 bytes will be received and consequently contain less errors (we chose to define it such that a missed symbol is not counted as an error symbol).



(a) The amount of errors for each symbol in the 34 byte packet. The black line indicates the theoretical distribution of errors without interference.



(b) The amount of bit errors for each part of the payload. Each bar represents one bit. Fields that are repeated in the payload have been aggregated for presentation purposes.

Figure 8: The amount of symbol and bit errors over all broken packets. Bars with the same color represent the same field.

Percentage of Broken Symbols with			
1 bit error	2 bit errors	3 bit errors	4 bit errors
22.3%	50.2%	24.4%	3.1%

Percentage of Payload Errors Caused by Symbols Creating			
1 bit error	2 bit errors	3 bit errors	4 bit errors
10.7%	48.2%	35.2%	5.9%

Table 2: The impact of symbol errors on the bit errors in the payload.

Symbol errors tell us something about the *transmission* errors of a packet since it is an aggregate representation of which chip sequences were received. However, we also want to consider the amount of *data* errors in the payload, measured as bit errors.

As described in Section 3.1, a symbol contains four bits of data, therefore a broken symbol may corrupt one, two, three or four bits of data. The number of bits that get corrupted depends on which erroneous symbol a chip sequence matches to. Ideally, one would think that one symbol error should only result in one bit error in the payload for a large majority of the symbol errors. This is however not the case for the weak links in our deployment as shown in the upper part of Table 2. We note that broken symbols causing 2 bit errors are the most common, and 1 and 3 bit errors are about equal, whereas 4 bit errors are rare. However, the impact on the payload errors, seen in the lower part of Table 2, show that only 10% of payload bit errors are caused by a mistaken symbol differing by one bit from the true symbol. This means that for 89.3% of the bit errors in the payload, they occur together, within a span of four bits, with at least one more bit error.

To further show how such symbol errors influence the payload errors we plot the position of payload errors over all packets in Figure 8(b). Here it is clear that the payload bit errors are not uniformly spread over the entire packet. We especially highlight the fact that every 4th bit in each symbol (bit 3) has a significantly lower likelihood of getting corrupted. Bits 0, 1 and 2 have about 25% higher chance of being corrupted than bit 3.

The non-uniformity of payload errors is an unintuitive artifact from the combination of symbol chip sequence construction and bias in each field's transmitted values. through analysis of our data we found that certain symbols have a bias to be more likely misinterpreted as other symbols, e.g., symbol 0 is very rarely misconstrued as symbol 8. This is an effect of the

design of the 16 chip sequences, where for this example the chip sequence of symbol 8 is the sequence for symbol 0 with every other bit inverted. The relation between the two sequences means that they will have a much lower likelihood of getting misinterpreted for one another. However, the non-uniformity of payload errors is also influenced by the fact that specific fields in the payload possess non-uniform value distributions.

5 Discussion

Understanding the extent and characteristics of transmission errors in low interference wireless sensor networks can give insights into possible performance enhancements. We have shown how the number of packets that can be received containing symbol errors, may comprise a significant amount of total packet transmissions over weak links. We can conclude that it would be beneficial to make use of this information, which would otherwise be lost. Given that nodes have to spend energy on receiving the entire payload before knowing if it contains errors or not, there should be more efficient strategies than just throwing it away immediately.

There is always the choice of performing ARQ or using a FEC scheme to improve data delivery. It seems that a FEC correcting at least two or three bit errors could substantially reduce the amount of broken packets. Making use of more intelligent ARQ, where repeatedly broken packets could be used to reconstruct the payload, could be another fruitful approach. The non-uniformity of payload errors also offers the potential for intelligent FEC schemes that rely on the fact that some bits are more likely to be successfully received than others.

The potential use of corrupted packets should not be dismissed. There are several possible uses from the information obtained. First of all, the fact that the receiver can receive a broken packet still tells you something about of the quality of the link. The more packets you can successfully receive with broken symbols the more probable it is that only a few symbols of the packet are being corrupted. This implies that there is likely no severe interference on the link and that a small change, such as marginally increasing the signal strength or changing the channel, may solve the problem. Another possibility would be to encode data in a different way, using the knowledge that some bits are less likely to get corrupted than others, and use those bits to represent the most important information.

We note that successfully received packets containing symbol errors in the payload is an additional source of information. Understanding the char-

acteristics and occurrence of transmission errors in this type of scenario will aid in determining both link and data quality. We remain cautiously optimistic about making use of corrupted data in environmental monitoring applications.

6 Related Work

Communicating over imperfect mediums using constrained devices is a major challenge in sensor networking. Jeong et al. design error mitigation schemes for different application scenarios [6]. They also note the fact that outdoor WSNs have a low number of errors per packet. In their outdoor experiment, without any FEC, they only have 0.12% 1-bit errors. Zhao et al. show how the set of packets where the preamble can be correctly detected and the set of packets where the CRC fails varies with distance [11]. Demonstrating that over weak links a large proportion of the sent packets are detected but fail the CRC.

Donapudi et al. conduct an experiment in a parking lot where they concluded that most packet errors are due to errors in the payload, motivating them to apply an FEC scheme [2]. They also show that at intermediate link qualities only one error per packet is by far the most common occurrence. Schmidt et al. investigate different FEC schemes, where the simplest one triples the number of sent bytes in order to do majority voting on the receiver [8]. They show that with bit error probabilities larger than 2% such a coding scheme performs poorly and suggest more complex approaches.

Collecting all available state about the nature of transmission errors to improve wireless communication approaches is frequently proposed. This may constitute using more advanced metrics than PRR when estimating link quality [3, 10]. Wu et al. [10] examine low level effects of other 802.15.4 interferers in an anechoic chamber with the use of software radios. On a per packet scale, large benefits can be seen from not simply throwing away packets with a bad CRC. Indeed, the use of symbols errors to perform partial packet recovery in 802.15.4 has been proposed by Jamieson et al. [5]. Their work uses software radios to gain access to the underlying received chip sequences when attempting to give estimates of the correctness of packets. Furthermore, they employ transmission power constraints to emulate long distance links. Their ideas on using fragmented CRCs could prove to be very useful for our scenario.

Our main contribution compared to this work is that we use standard sensor nodes deployed in a low interference environment to investigate sym-

bol error rate's impact on packet payloads.

7 Summary

In this paper we have detailed our WSN deployment at a polar research station in Abisko, Sweden. We described the experimental setup, how it is designed to be deployed in such an extreme environment and the way that communication links are setup to maximize the output of experimental data.

Furthermore, we have illustrated transmission error characteristics of links in such a low interference and isolated location. Results show that there are a significant number of packets sent over weak links that can be successfully received but contain errors in the payload. The errors in each 802.15.4 packet are few, typically only one or two symbols. We found that the amount of received broken packets can be as high as 40% when no packets are being successfully received. Finally, the collected data shows the probability of symbol errors is equal over all positions in the transmission, but the probability of bit errors in the payload can vary by up to 25%. We conclude that this is due to the specific coding scheme used in 802.15.4 and payload value bias.

Acknowledgments

This work was carried with support from the Centre for Natural Disaster Science (CNDS), EU/INTERACT project and the SICS Center for Networked Systems (CNS).

References

[1] Crossbow Inc. *TelosB datasheet*, Revision B edition, May 2004.

[2] S. Donapudi, C. Obel, and J. Madsen. Extending lifetime of wireless sensor networks using forward error correction. In *Norchip Conference, 2006. 24th*, pages 277–280, 2006.

[3] P. Heinzer, V. Lenders, and F. Legendre. Fast and accurate packet delivery estimation based on dsss chip errors. In *INFOCOM, 2012 Proc. IEEE*, pages 2916–2920, 2012.

[4] IEEE 802.15.4 Working Group. *IEEE Standard for Local and Metropolitan Area Networks - Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)*, IEEE Std 802.15.4-2011 edition, Sep 2011.

[5] K. Jamieson and H. Balakrishnan. Ppr: partial packet recovery for wireless networks. In *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '07, pages 409–420. ACM, 2007.

[6] J. Jeong and C. T. Ee. Forward Error Correction in Sensor Networks. 2003.

[7] O. Rensfelt, F. Hermans, L.-A. Larzon, and P. Gunningberg. SenseiUU: A Relocatable Sensor Network Testbed. In *Procs. of WiNTECH '10*, pages 63–70, September 2010.

[8] D. Schmidt, M. Berning, and N. Wehn. Error correction in single-hop wireless sensor networks - a case study. In *Design, Automation Test in Europe Conference Exhibition, 2009. DATE '09.*, pages 1296–1301, 2009.

[9] Texas Instruments Inc. *CC2420 - 2.4 GHz IEEE 802.15.4, ZigBee-ready RF Transceiver.*

[10] K. Wu, H. Tan, H.-L. Ngan, Y. Liu, and L. M. Ni. Chip Error Pattern Analysis in IEEE 802.15.4. *Mobile Computing, IEEE Transactions on*, 11(4):543–552, 2012.

[11] J. Zhao and R. Govindan. Understanding packet delivery performance in dense wireless sensor networks. In *Proceedings of the Embedded Networked Sensor Systems*, SenSys '03, pages 1–13. ACM, 2003.

Paper IV

Paper IV
All is not Lost: Understanding and Exploiting Packet Corruption in Outdoor Sensor Networks

Frederik Hermans, Hjalmar Wennerström, Liam McNamara, Christian Rohner, and Per Gunningberg *All is not Lost: Understanding and Exploiting Packet Corruption in Outdoor Sensor Networks*
To be published at the 11th European Conference on Wireless Sensor Networks (EWSN), February 2014, Oxford, England.

All is not Lost: Understanding and Exploiting Packet Corruption in Outdoor Sensor Networks

Frederik Hermans, Hjalmar Wennerström, Liam McNamara,
Christian Rohner, Per Gunningberg
Department of Information Technology, Uppsala University, Sweden

Abstract

During phases of transient connectivity, sensor nodes receive a substantial number of corrupt packets. These corrupt packets are generally discarded, losing the sent information and wasting the energy put into transmitting and receiving. Our analysis of one year's data from an outdoor sensor network deployment shows that packet corruption follows a distinct pattern that is observed on all links. We explain the pattern's core features by considering implementation aspects of low-cost 802.15.4 transceivers and independent transmission errors. Based on the insight into the corruption pattern, we propose a probabilistic approach to recover information about the original content of a corrupt packet. Our approach vastly reduces the uncertainty about the original content, as measured by a manifold reduction in entropy. We conclude that the practice of discarding all corrupt packets in an outdoor sensor network may be unnecessarily wasteful, given that a considerable amount of information can be extracted from them.

1 Introduction

Outdoor sensor networks experience significant variations in radio link performance over time [11, 17]. When links are in a transient state, they receive a large amount of corrupt packets, which are commonly discarded. Consequently, the information the sender intended to transmit is lost and has to be retransmitted. Therefore, corrupt packets incur a cost on the networks' limited energy budget for both transmitting and receiving the corrupt packet, and for retransmission.

We study corrupt packets from an 802.15.4-based outdoor deployment in a remote area. We find that corruption occurs to a non-negligible degree

on intermediate links. It emerges that corruption follows a distinct, stable pattern that holds over various time scales and across links. We explain this pattern by considering an implementation aspect of low-cost 802.15.4 transceivers, the tie resolution strategy in coding, and a channel model in which errors occur independently. While corruption in packets has been studied recently in outdoor networks [14] and earlier in the case of interference [10, 6], we are the first to explain the occurence of the observed pattern.

Some earlier work has also addressed how to make use of corrupt packets. Apart from forward error correction, the approaches either selectively retransmit parts of a packet that are suspected to be corrupted [8, 5], or aim to reconstruct a correct packet from multiple corrupt packets [2].

We take a novel path to handling corrupt packets. We note that data in sensor networks is often inherently uncertain, e.g., due to limited accuracy of sensor readings. We therefore propose an approach that—rather than trying to exactly reconstruct a corrupt packet—probabilistically infers the packet's original content by exploiting the pattern in corruption. In combination with application knowledge, our approach enables recovery of information from corrupt packets. In contrast to earlier work, our approach does not need retransmissions of corrupt packets and hence does not incur an additional communication cost.

The evaluation of our approach shows that the uncertainty associated with a corrupt packet can be reduced significantly, as measured by an up to eight-fold reduction in entropy. We further validate our approach by applying it to data collected from a second deployment in another location, and find that it enables recovery of information from corrupt packets.

In summary, this paper makes the following core contributions:

- By analyzing data from a long-term outdoor deployment, we describe the distinct pattern of how 802.15.4 packets are corrupted. Crucially, we can explain the pattern by considering implementation aspects of low-cost 802.15.4 transceivers and a simple radio channel model.

- Based on our insights, we describe an approach that probabilistically infers the original content of a corrupt packet. We evaluate this approach on a data set from a separate deployment, and find that it enables recovery by correctly assigning high probabilities to the original content. We also achieve a manifold reduction in the uncertainty associated with a corrupt packet.

To ensure that this paper is focused and self-contained, we have decided

to leave out certain systems aspects, which we will address in future work.

The rest of the paper is organized as follows. We describe our deployment and data collection in Sec. 0.2, and briefly recap the IEEE 802.15.4 standard in Sec. 0.3. We analyze packet corruption in our deployment in Sec. 0.4, and describe our recovery approach in Sec. 0.5. Section 0.6 evaluates the approach, followed by a brief discussion of practical aspects in Sec. 0.7. We then survey related work in Sec. 0.8 and conclude the paper in Sec. 0.9.

2 Deployment and Data Collection

We deployed a sensor network at the outskirts of *City, Country*[1]. The network is located in an open field with no trees or bushes in the surroundings, in a remote location that very few people have access to. The deployment is therefore not affected by man-made radio interference, e.g., from WiFi or Bluetooth.

The network is comprised of 16 TelosB sensor nodes, which are equipped with 802.15.4-compatible CC2420 radio transceivers that operate in the 2.4 GHz ISM band [16]. The nodes are attached to four poles, with four nodes per pole (Fig. 1a). The poles are aligned along a straight line with a distance of 20 m between consecutive poles, as shown in Fig. 1b. On each pole, two nodes are mounted at 0.5 m above the ground and two nodes are mounted at 1.5 m.

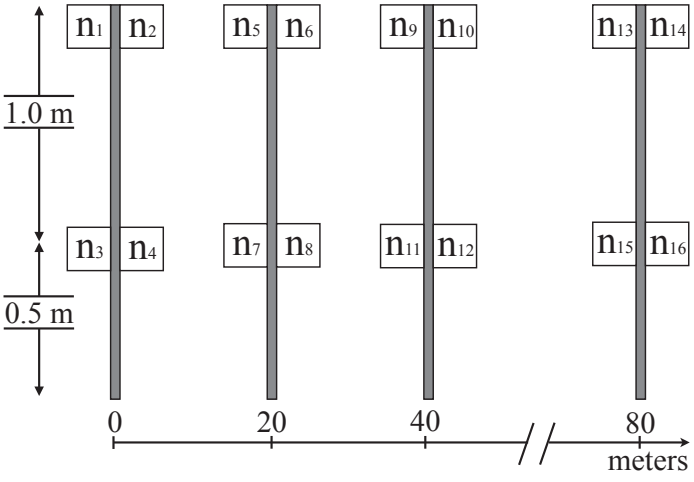
The purpose of the network is to study radio links in 802.15.4 outdoor networks. Therefore, nodes take turns in sending 34-byte long probing packets every 500 ms. Whenever a node receives a packet, it logs the received packet content and the signal-to-noise ratio and Link Quality Indication (LQI) associated with the packet. Rather than discarding corrupt packets, nodes are programmed to also log corrupt packets. For power supply and log data collection, all nodes are connected via USB to a central experiment monitor, which is a regular desktop PC. By analyzing the log files, which contain all sent and received packets (both correct and corrupt), we can determine which parts of a corrupt packet have suffered corruption. We use this information to analyze corruption in Sec. 0.4.

3 Recap of IEEE 802.15.4

We briefly recapitulate the aspects of the IEEE 802.15.4 standard that are relevant to this paper. IEEE 802.15.4 is a standard for low-rate, low-power

[1] Redacted for double-blind review.

(a) Pole with four sensor nodes



(b) Deployment layout

Figure 1: Outdoor deployment. Sensor nodes are labeled 1–16 in the right figure.

wireless communication [7], which has found wide-spread adoption in sensor networks. In the 2.4 GHz band, it offers a data rate of 250 KB/s.

A transmitted byte is represented by two four-bit *symbols*. 802.15.4 employs a direct sequence spread spectrum (DSSS) technique, in which each of the 16 possible symbols is represented by one *code word*. A code word, in turn, is represented by a 32-bit long pseudo-noise *chip sequence*.

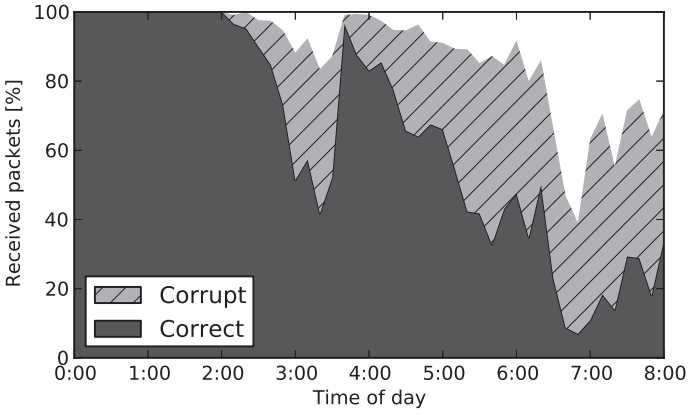
We clarify the operation of 802.15.4 with an example. A sender wants to transmit a packet with n bytes of payload to a receiver. The sender translates each byte to two symbols. For each symbol, it determines the corresponding code word. The code words' chip sequences are then modulated onto a carrier frequency. The receiver demodulates the incoming chip sequences and matches them to the known code words. In this way, the receiver decodes $2n$ symbols, from which it can construct the n payload bytes.

Synchronization is required to detect packet boundaries. A sender starts each packet with a predefined preamble, followed by a start frame delimiter and a length field. Upon decoding a preamble and start frame delimiter, a receiver knows that a packet is being transmitted. If the receiver fails to decode the preamble, the packet is lost.

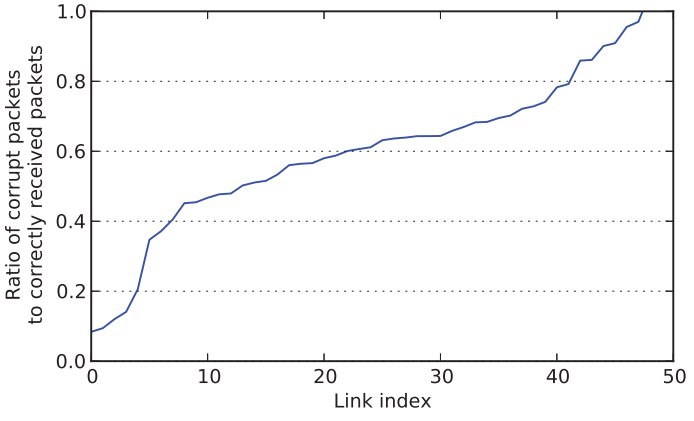
Due to noise on the radio channel, a receiver's demodulated chip sequence may differ from the chip sequence transmitted by the sender. In this case, the incoming chip sequence is matched to the closest code word. DSSS thereby achieves resilience against noise, since there is a many-to-one mapping between chip sequences and code words. If sufficiently many chips are demodulated incorrectly [19], the receiver matches the incoming chip sequence to an incorrect code word, and hence decodes the wrong symbol. In this case, packet corruption occurs. To detect corruption, 802.15.4 packets end with a two-byte cyclic redundancy check (CRC) field. A receiver computes the CRC for the incoming packet and compares it against the received trailing CRC field. If they mismatch, the receiver knows that corruption has occurred.

4 Packet Corruption in an Outdoor Sensor Network

In this section, we analyze corrupt packets that were received by nodes in our deployment over the course of one year, from June 2012 to June 2013. We begin by briefly quantifying the amount of corruption occurring in the deployment. Then, we describe how corruption affects individual transmit-



(a) Example of a deteriorating link



(b) Ratio of corrupt to correct packets

Figure 2: The left figure exemplifies the amount of corrupt packets received for a specific link. The right figure shows that almost all intermediate links receive a substantial amount of corrupt packets.

ted symbols, followed by a characterization of the effect of corruption on whole packets. Our analysis is focused on the regularities in corruption that enable the probabilistic recovery of information, as described in Sec. 0.5.

Because corrupt packets are usually discarded, the degree to which packet corruption occurs is unknown for most sensor networks. From our log data we observe that *intermediate links* (links that have a PRR between 10% and 90% [15, 1]) experience a substantial amount of packet corruption. Figure 2a shows a representative example of such an intermediate link. The depicted link initially has a high PRR, but deteriorates over time. As PRR falls, the amount of corrupt packets, indicated by the hatched gray area, grows.

Next we look at the amount of corruption over all intermediate links from the duration of the deployment. We observe that about 80% of intermediate links have a ratio of corrupt packets to correctly received packets of at least 0.5. That is, for every two correctly received packets, they receive one corrupt packet on average. Figure 2b illustrates this ratio of corrupt packets to correctly received packets for intermediate links for a time span of two weeks in March 2013.

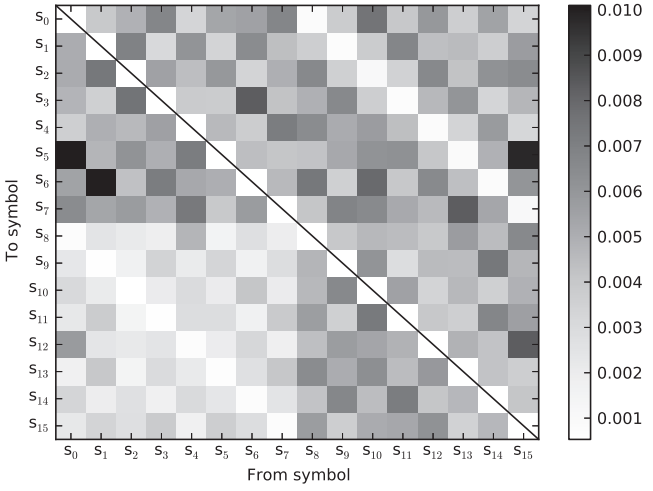
We conclude that packet corruption occurs at a non-negligible scale on intermediate links. Because intermediate links are the best candidates for improving network performance, this initial observation motivates to understand packet corruption in more detail.

4.1 Corrupt Symbols

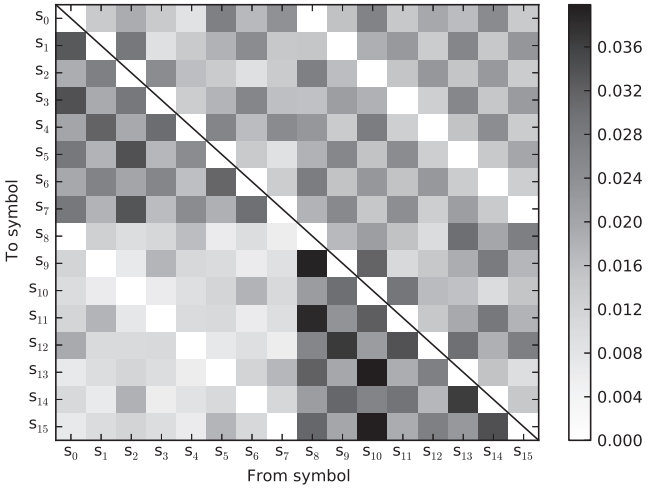
We now consider corruption at the finest level of granularity at which it can be observed in our deployment: the symbol level. If a node sends a packet containing a symbol s_i and due to corruption a receiver decodes the symbol incorrectly, which symbol s_j will the receiver likely decode?

Figure 3a shows how often each possible mutation $s_i \rightarrow s_j$ is observed over all links from the span of twelve months. The figure is a visual representation of the *mutation matrix*. An entry (j, i) of the mutation matrix denotes the frequency with which we observed a sent symbol s_i to be received as s_j. The matrix diagonal describes how often a symbol was decoded correctly. We omit the diagonal in the figure to focus on corruption. The darker the color in the figure, the higher the frequency. A distinct visual structure emerges in the figure, which leads us the following three observations:

Observation 1: Mutations are not uniformly distributed. If corruption occurs to a sent symbol, the received symbol depends on which symbol was sent. For example, if a node sends symbol s_0 which suffers corruption, the receiver most commonly decodes it as a s_5 (see column 0). Conversely, if a



(a) Symbol mutations in data set



(b) Symbol mutations in simulation

Figure 3: Symbol mutations as observed in the data set and in simulation. Our simulation model produces the same core pattern as the empirical measurements.

node receives a corrupt symbol as s_5, it is least common that s_{13} was sent (see row 5).

Observation 2: The most significant bit of a symbol is more stable than other bits. Note that the subdiagonal $(8, 0)$ to $(15, 7)$ and the superdiagonal $(0, 8)$ to $(7, 15)$ are lighter than the rest of the plot. For each column, the corresponding entry on these diagonals represents the symbol which differs from the sent symbol in only the most significant bit. For example, the sent symbol $s_0 = 0000_2$ is least commonly decoded as $s_8 = 1000_2$. This leads to the most significant bit of a symbol being more stable on average in our deployment.

Observation 3: Symbols s_0 to s_7 are more stable than other symbols. Consider the bottom left quadrant in Fig. 3a. It is significantly lighter than the other quadrants. This reflects the fact that in our data, the symbols s_0 to s_7 are unlikely to be decoded as s_8 to s_{15}. The converse is not true. Consequently, symbols s_0 to s_7 are less commonly corrupted than symbols s_8 to s_{15}.

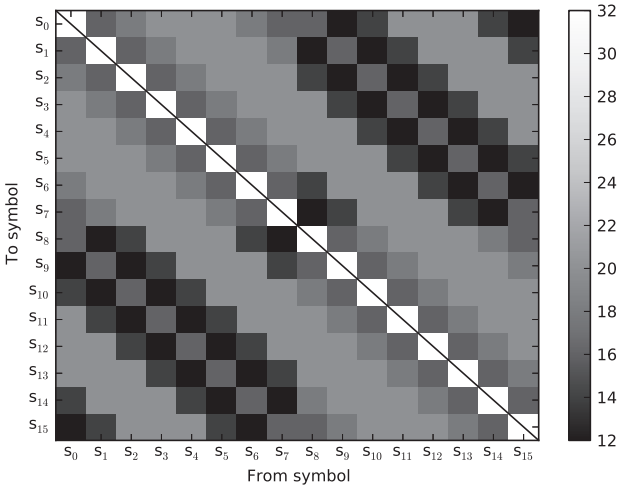
The pattern shown in Fig. 3a represents mutation frequencies aggregated over all links over the whole time span of the deployment. We confirmed that the pattern also holds for individual links, and at various time scales. An independent research group has recently observed a similar pattern in an outdoor environment [14], which suggests the observations to be general.

To the best of our knowledge, no explanation has been offered so far as to why the pattern emerges. As Schmidt et al. point out [14], the pattern is surprising because it shows a negative correlation to the pairwise hamming distances of the code words defined in the 802.15.4 standard (see Fig. 4a). For example, the hamming distance between the code words for s_0 and s_8 is low, yet this mutation is among the least common in our deployment.

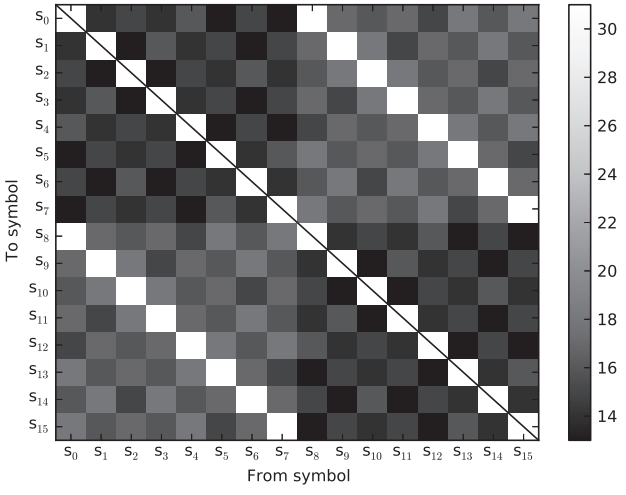
We attribute the first two observations to an implementation aspect of low-cost 802.15.4 transceivers. Rather than implementing an O-QPSK demodulator, as suggested in the 802.15.4 standard, many low-cost transceivers use an MSK demodulator instead [13]. While an MSK demodulator can correctly receive a chip sequence sent by an O-QPSK modulator, the received chip sequence will be transformed. Therefore, MSK-based 802.15.4 transceivers use a transformed set of code words to ensure compatibility with other 802.15.4 transceivers. The hamming distances between the transformed code words are shown in Fig. 4b.

Observation 1 can be explained by considering that each code word varies in its hamming distances to other code words. Therefore, the symbol that a corrupt chip sequence is decoded as depends on which symbol was sent.

Next, observation 2 follows directly from the observation that the MSK-



(a) Hamming distances between code words of the 802.15.4 standard



(b) Hamming distances between MSK-transformed code words

Figure 4: Hamming distances for 802.15.4 code words (left) and MSK-transformed code words (right). The MSK transformation explains observations 1 and 2.

transformed code word for symbol s_i has the highest hamming distance to the MSK-transformed code word for the symbol which differs from s_i only in the most significant bit. This is visualized by the light sub- and superdiagonals in Fig. 4b. Therefore, the most significant bit is more stable on average.

It remains to explain observation 3, which states that symbols s_0 to s_7 are more stable than the other symbols. This observation does not follow from the use of transformed code words, because code word distances are of course symmetric. We can explain the observation by considering how ties are resolved. A tie occurs if a received chip sequence matches two or more code words equally well. In this case, the transceiver must resolve the tie by choosing one of the matching code words. In a simple simulation, we found that if ties are resolved in a specific order[2], a pattern very similar to the empirically observed mutation matrix emerges (Fig. 3b). With the found order, a tie between two code words $s_{0\leq i\leq 7}$ and $s_{8\leq j\leq 15}$ will always be resolved in favor of the first symbol. Consequently, symbols s_0 to s_7 are more stable, as stated by observation 3.

Our simulation assumed one sender and one receiver, a fixed signal-to-noise ratio (SNR) at the receiver, and a channel model in which chip errors are independent, as would be expected in an additive white Gaussian noise channel, for example. Fixing the SNR implies a fixed chip error probability at the receiver.

We draw another useful conclusion from the similarity of the empirical and the simulated mutation matrix. The similarity suggests that the radio channel in the deployment can be modeled by a channel in which errors are independent, as assumed in our simulation. Note that differences in absolute values in Fig. 3a and Fig. 3b can be explained by considering that the simulated mutation matrix is based on a fixed SNR, whereas the empirical mutation matrix is based on packets received at various SNR levels. Nonetheless, the similarity holds.

In summary, we conclude that corruption follows a distinct pattern, which we attribute to MSK-transformed code words, the tie resolution strategy and a radio channel with independent chip errors.

4.2 Errors in Packets

In the previous section, we considered the effect of corruption on individual symbols. We now shift our focus one layer up in the network stack to the

[2]The order is $s_7, s_6, \ldots, s_0, s_{15}, s_{14}, ..., s_8$.

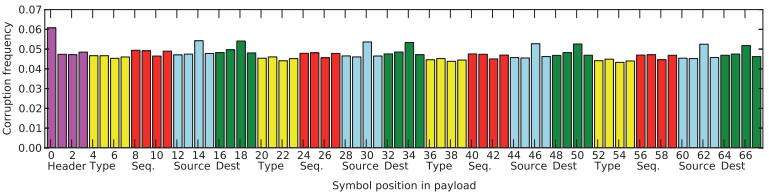


Figure 5: Symbol error frequency over positions of the payload. Error frequencies are roughly uniform. The variations can be attributed to the payload structure and content.

link layer and consider how corruption affects whole packets, i.e., sequences of symbols.

4.2.1 Distribution of Errors

How are symbol errors distributed within a packet? Figure 5 shows the distribution of errors within the corrupt packets. For each position of the payload, the plot shows the frequency with which a symbol at this position was corrupt. The x-axis is annotated with the content of the payload. For example, the first four positions contain the packet header.

The error frequency is similar across all positions, ranging from 4.5% to 6%. Although the distribution is roughly uniform, there are notable deviations. First, the symbol at position 0 is most often corrupt. Second, there is a periodicity: the error frequencies for positions 4 to 19 are similar to the frequencies for positions 20 to 35, and so on. These deviations can be explained by the content of sent packets. For all packets sent in our deployment, position 0 always contains symbol s_8, which we know to be least stable. Furthermore, the payload of the probing packets sent in our deployment repeats itself after position 20, giving rise to the observed periodicity. Finally, because the sent packets contain structured rather than random content, some positions have slightly higher corruption frequencies than others. The observed deviations from uniformity are within range of the deviations we would expect due to the effects described in Sec. 0.4.1.

4.2.2 Correlation of Errors

Are errors correlated? I.e., does an error at position x tell us something about whether an error occurred at position y? We computed pairwise

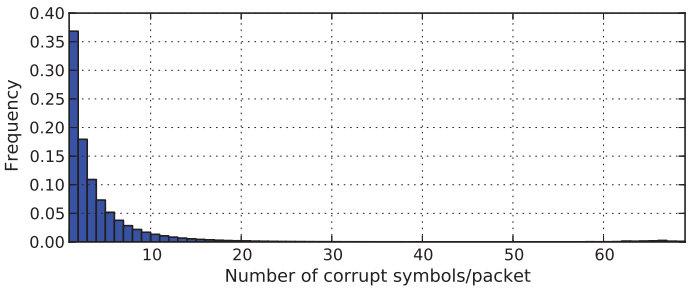


Figure 6: Number of corrupt symbols per packet. Most packets suffer only little corruption.

correlations between all positions over all corrupt packets. The maximum absolute correlation between any two symbol positions is less than 0.09. Considering that a value of 0 indicates no correlation at all, we conclude that there are no notable correlations between errors at different positions. Therefore, symbol errors are independent from each other. This observation agrees with our assumption that the deployment's radio channel is well described by assuming independent chip errors.

4.2.3 Amount of Corruption in a Packet

Finally, we quantify how many symbols in a corrupt packet are incorrect. Figure 6 shows a normalized histogram of the number of symbol errors per corrupt packet. The figure shows that most packets have very few errors, and that the frequency of occurrence decreases with an increase in the amount of corruption.

5 Recovering Data from Corrupt Packets

We now describe how we use the observations from the previous section in an approach which for a given corrupt packet defines a probability distribution over the possible sent packets.

5.0.4 Computing a Probability Distribution over Possible Sent Packets

We consider how to infer the likely sent data from a received corrupt packet. Our goal is to assign probabilities to the possible sent data, given the data in a corrupt packet. Recall from Sec. 0.3 that corruption occurs if sufficiently many chips in an incoming chip sequences are decoded incorrectly and hence the chip sequence is matched to the wrong code word. For the remainder of this analysis, we denote the probability of an individual chip in a received chip sequence being flipped as p_{chip}. For now, assume that we know p_{chip} for each received packet. We will revisit this assumption in the next section.

For a given value of p_{chip}, we can compute through simulation a corresponding mutation matrix $M^{p_{\text{chip}}}$. For example, the mutation matrix shown in Fig. 3b describes the mutation probabilities for $p_{\text{chip}} = 0.3$. Most importantly to our approach, the matrix rows describe the mutation probabilities for a received symbol. Note that the matrix main diagonal describes the probabilities of a symbol being received correctly.

For a given received packet, we now want to infer the first symbol of the sent packet. Let the first symbol of the received packet be s_j, and consider the case in which we know the packet to be corrupt because the CRC failed. By considering row j of $M^{p_{\text{chip}}}$, we can attach a probability to each possible sent symbol that could have led to the receiver decoding s_j. More specifically, the probability that the sent symbol s_i is decoded as the received symbol s_j is given by the entry $M^{p_{\text{chip}}}_{j,i}$ of the mutation matrix. We write $p(s_i|s_j) = M^{p_{\text{chip}}}_{j,i}$. For example, the probability $p(s_5|s_{13})$ that symbol s_5 was sent when s_{13} was received is given by $M^{p_{\text{chip}}}_{13,5}$. This reasoning holds for all position of the packet.

Because symbol errors are independent, we can readily assign probabilities to sequences of sent symbols. Assume, for example, that a receiver decoded the sequence of symbols $r = (s_{13}, s_3, s_0, s_{11})$. What is the probability that the actual sent symbols were $t = (s_5, s_3, s_1, s_{11})$? Due to independence, this probability is given by the product of the individual mutation probabilities:

$$\begin{aligned} p(t|r) &= p(s_5, s_3, s_1, s_{11}|s_{13}, s_3, s_0, s_{11}) \\ &= p(s_5|s_{13}) \cdot p(s_3|s_3) \cdot p(s_1|s_0) \cdot p(s_{11}|s_{11}) \\ &= M^{p_{\text{chip}}}_{13,5} \cdot M^{p_{\text{chip}}}_{3,3} \cdot M^{p_{\text{chip}}}_{0,1} \cdot M^{p_{\text{chip}}}_{11,11} \end{aligned}$$

In the manner we just outlined, a probability can be assigned to every

possible sent symbol sequence for a given received symbol sequence and a given value of p_{chip}. This conceptually simple idea comprises our recovery approach. For each received, corrupt packet, we can compute a probability distribution over the possible sent packets. To compute the distribution, all we need to know is the chip error probability p_{chip} during packet reception.

To summarize, our approach determines a probability distribution over the possible sent data for given received data in a corrupt packet and a given p_{chip}. This concludes our description of recovery. We deliberately do not specify how the probability distribution is to be used by an application, because we believe that application knowledge should drive this process.

5.0.5 Estimating p_{chip}

To assign probabilities to possible sent data, we need an estimate of the chip error probability p_{chip} for each corrupt packet. Unfortunately, low-cost transceivers do not provide such an estimate directly. Although there is a well-defined relationship between SNR and the chip error probability [20], we found the resolution of SNR reported by low-cost transceivers too low for a meaningful p_{chip} estimate.

To overcome this obstacle, we estimate p_{chip} for each packet by considering the LQI value associated with the packet. In the case of CC2420 transceivers, LQI is reflective of the correlation of an incoming chip sequence to the matched code word over the first eight symbols of a packet [16]. Therefore, we expect it to reflect the chip error rate. We construct a mapping from LQI values to chip error estimates as follows: for each LQI value l, we determine the empirically observed symbol error probability for symbol s_0. We then calculate the chip error probability p^l_{chip} that yields the same symbol error probability for s_0. We then construct a mapping from LQI to chip error probability by interpolating a 3rd degree spline through the resulting (l, p^l_{chip}) tuples. Our mapping is defined on LQI values in the range from 32 to 90, which covers 98.5% of all corrupt packets in our data set. We constrain the mapping to this range because we observe only very few corrupt packets with LQI less than 32 or higher than 90, and we therefore have little support to construct a mapping for these values.

While we do not expect our LQI to p_{chip} mapping to be perfect, we note that given the information that low-cost transceiver usually provide about the channel, it is difficult construct with a more well-defined estimate. We are confident that if transceivers were to provide high-resolution SNR measurements, a more exact estimator of chip error probabilities can be designed.

6 Evaluation

Our proposed approach defines a probability distribution over the possible sent data for given received data in a corrupt packet. We now address two questions pertaining to the resulting distributions. First, to what extent does the approach reduce uncertainty about the original content of a corrupt packet? Second, is the resulting distribution for a given corrupt packet meaningful? I.e., does it assign probabilities in a way such that the actual sent data has a high probability?

6.1 Reduction in Uncertainty

Let us address the first question of how much our approach reduces the uncertainty associated with a received, corrupt packet.

We consider the case in which a node sends a packet containing a 16-bit word t that we are interested in. This word could, for example, encode a sensor measurement, but for the sake of this analysis, we assume that we do not have any application-specific knowledge about the likely content t. We assume that a corrupt packet is received that contains the 16-bit word r. Due to the corruption, we do not know whether $r = t$ or not.

As a base case, assume that the corrupt packet is simply discarded. In this case, we know nothing about t. It could have taken any of the $2^{16} = 65{,}536$ possible values with equal probability. We measure the uncertainty associated with the discarded, corrupt packet by considering the entropy of the probability distribution over all possible words t' that could have been transmitted. There are 16 bits of entropy:

$$\begin{aligned} H_{\text{discard}} &= -\sum_{t'=0}^{65535} p(t') \log p(t') \\ &= -\sum_{t'} 2^{-16} \log 2^{-16} = 16. \end{aligned}$$

Next, we consider the case in which we do not discard the corrupt packet. The corrupt packet contains a word r, but we do not know if $r = t$. Using the approach described in Sec. 0.5, we can compute the probability $p(t'|r)$ for every possible 16-bit word t'. As in the case of the discarded packet, we can compute the entropy, which depends on the estimate of p_{chip} and on the received word r:

$$H_r = -\sum_{t'=0}^{65535} p(t'|r) \log p(t'|r).$$

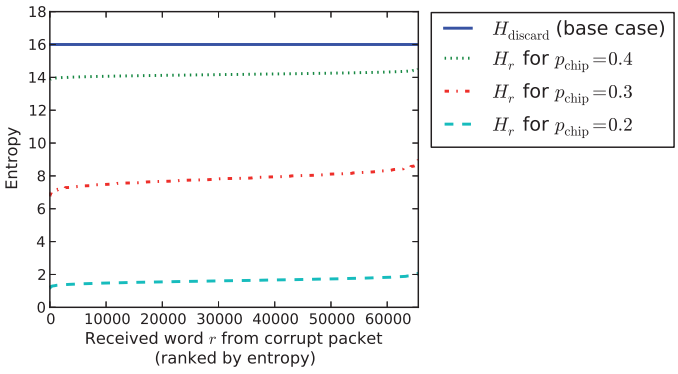


Figure 7: Uncertainty associated with a corrupt packet in the case the packet is discarded, and when our proposed approach is used. Our approach significantly reduces the uncertainty, as measured by entropy.

Figure 7 depicts the entropy for both the base line case and our approach for different values of p_{chip}. The y-axis denotes the entropy. The x-axis relates to the received word r as described below.

In the base case, in which the corrupt packet is discarded, the entropy is 16 regardless of which word was received in the corrupt packet, and regardless of the chip error probability. With our approach, which assigns probabilities the possible sent words by considering which word r was received in the corrupt packet, the entropy depends on r. This is an effect of our observation that symbol mutations are not uniform. For a given chip error probability p_{chip}, the entropies H_r are plotted in increasing order along the x-axis.

The figure makes it clear that our approach significantly reduces the entropy associated with a corrupt packet. In the case of a chip error probability of 0.2, the entropy is reduced to less than 2 bits—an eight-fold reduction of the entropy of the base case. For a higher chip error probability of 0.3, the entropy is still halved in comparison to the base case. In the case of an extreme chip error probability of 0.4, the entropy is reduced by two bits. However, for such a high chip error probability, most packets will be lost rather than corrupt because the preamble is likely to be corrupt as well. We therefore conclude that for realistic chip error probabilities of 0.2 to 0.3, our approach vastly reduces the uncertainty associated with a corrupt packet.

6.2 Evaluation of Probability Assignment

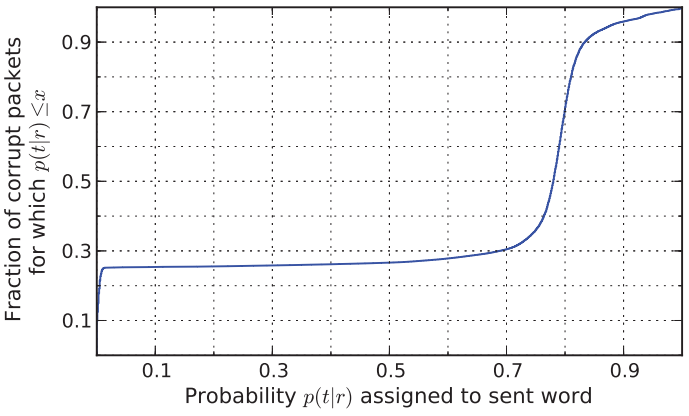
We have shown how our probability assignment reduces the uncertainty associated with a corrupt packet. It remains to show that the probability assignment is sensible, i.e., that there is a meaningful relationship between the probability assigned to possible sent words t' and the word t that was actually sent.

To address this question, we consider corrupt packets from a deployment different from the one that provided the data for the analysis in Sec. 0.4. Evaluating our approach on data from a different deployment increases our confidence in the generality of our findings, and helps understand whether our approach is strongly tied to the observations from deployment A[3]. Deployment B is located above the polar circle (latitude of $66^\circ\,33'\,44''$ N) in a climate that differs significantly from the climate at deployment A. It consists of 12 TelosB sensor nodes, and has a spatial layout similar to deployment A. We consider corrupt packets from deployment B that were received during the first week of April 2013. We focus on this week because the deployment did not have any operational problems, such as failing nodes.

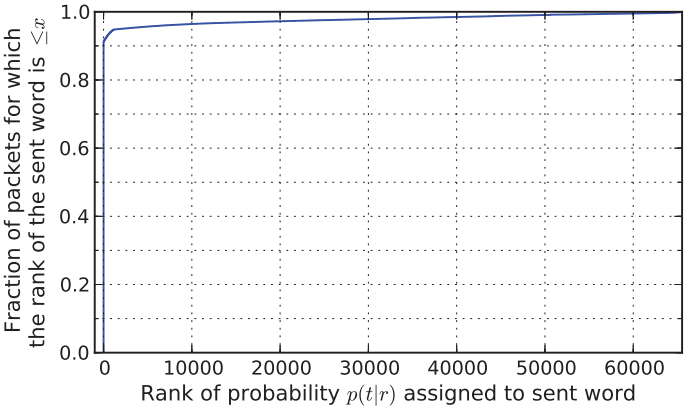
The data set of deployment B contains ca. 440,000 corrupt packets. We know the correct payload for each corrupt packet from our log data. We consider a 16-bit word t in these packets that describes the source address of the sender. We do not use any knowledge about the possible content of this word. For each corrupt packet, we estimate the chip error probability p_{chip} based on the packet's LQI measurement, as described earlier. We use the estimate to compute $p(t|r)$, which is the probability that is assigned to the sent word t when r was received. Clearly, it is desirable that a high probability be assigned to the sent word.

Figure 8a shows the empirical cumulative distribution function of the probability assigned to the sent word t for each corrupt packet. The x-axis shows the probability assigned to the sent word. The y-axis shows for how many of the packets this probability was below the corresponding x value. Note that for only 30 % of the corrupt packets, a probability of less than 0.7 is assigned to the sent word. For only 28 % of the corrupt packets, the probability assigned to the sent word is less than 0.5. It follows that for most corrupt packets, a high probability is assigned to the word that was actually sent. For these packets the probability assignment is sensible. However, a very low probability is assigned to the correct word for about 25% of the corrupt packets. Note that this does not imply that the

[3]We refer to the deployment described in Sec. 0.2 as *deployment A*, and to the deployment described here as *deployment B*. We would reveal location names if accepted.



(a) Probabilities assigned to sent word



(b) Rank of the sent word

Figure 8: Our approach correctly assigns a high probability to the actual sent word, both in absolute (left) and relative (right) terms.

assignment is wrong—in cases of a high chip error probability, there will be high uncertainty. High uncertainty means that the probability distribution will be more uniform across all possible sent words. The question thus is whether these low-probability assignments come from corrupt packets with high chip error probabilities. Before turning to this question, we conclude from Fig. 8a that for more than 70% of the corrupt packets, the probability assignment is sensible, because it assigns a high probability to the sent word.

We now order all possible sent words by decreasing order of assigned probability and determine the rank. E.g., the word t' that has been assigned the highest probability has rank one, the word with the second highest probability has rank two, etc. If two or more words have the same probability, they have the same rank. We are interested in the rank assigned to the sent word t.

The distribution of rank of the sent word is shown in Fig. 8b by an empirical cumulative distribution function. The figure shows that in 95% of the cases, the sent word is assigned a very high rank. I.e., the sent word is assigned a higher probability than most other possible candidates. We take this as an indication that even in cases where the highest probability is not assigned to the sent word, the sent word still takes a very high probability in comparison to other possible candidates. For the remaining 5% of corrupt packets, the rank is almost uniformly distributed up to the maximal rank of 65,536. We attribute this observation to misestimations of the chip error probability p_{chip}. Since we estimate p_{chip} from LQI, and LQI is only measured over the first eight symbols of a packet, it may by pure chance sometime misrepresent the chip error probability.

To summarize, we have shown in this section that for most corrupt packets, the sent word is assigned a high probability in comparison to other candidates. We conclude that the probability assignment we described in Sec. 0.5 indeed assigns probabilities in a meaningful manner. This observation suggests that our estimator of p_{chip} based on LQI is sufficiently accurate to enable recovery. Our approach thus enables sensor networks to infer the possible sent word corresponding to a corrupt packet.

7 Practical Considerations

We briefly discuss three aspects of practicality.

First, our approach determines a probability distribution over possible sent data. It does, by design, not produce a single value. When application knowledge about the likely content of a packet is available, this knowledge

can be combined with our probability distribution to constrain the likely sent data even further. Application knowledge could be, for example, knowing the domain of a measured value from previous measurements. Such knowledge is often used to detect outliers, assess data quality, or handle missing data (e.g., see [19, 9, 4]). Such approaches are largely orthogonal to our proposal. Because the distributions computed by our approach are not centered around a single value, we believe that in combination with application knowledge, an even more exact inference of the content of corrupt packets is possible.

Second, a related question pertains to the complexity of our proposal. Note that the maths involved in determining the probability distribution is computationally very simple. Yet, for a received n symbol sequence, there are 16^n different possible sent values in the case of corruption. Enumerating all of them is infeasible for larger values of n. However, even for situations with moderately high chip error probabilities, many probabilities will be very close to zero. We envision that an application performing recovery will be interested in the top $k \ll 16^n$ possible sent sequences with the highest probabilities. These can be determined efficiently without enumerating all possible values. Therefore, we are confident that recovery can be performed in-network by nodes that are slightly more powerful than the TelosB-type nodes.

Third, the packets we analyzed in this paper were all sent and received by Texas Instruments CC2420 transceivers. Although this particular chip has a very high prevalence in academic research, the question arises of how well our findings translate to other 802.15.4 radio chipsets. In part, the observed pattern is an effect of the use of MSK demodulators, which are cheap to implement [12]. Therefore, they are common in low-cost transceivers. Consequently, we expect the pattern to hold for other transceiver, too. Note that because the pattern emerges even on short time scales, its presence in a particular radio chipset can be readily verified in an anechoic chamber.

8 Related Work

Wireless channels are inherently unstable [21, 15], causing errors in transmissions, and making mitigation strategies for these errors a wide field of research.

Schmidt et al. study corruption in an 802.15.4-based outdoor network and make observations similar to ours [14]. They point out that bit errors are not equally probable over all positions in the payload in 802.15.4 packets.

They compare their empirical results to the expected values using code words as used with O-QPSK modulation. Han et al. identify patterns in the bit error probabilities of the payload in 802.11, which are not due to the channel conditions nor hardware-specific [3].

By using a software-defined radio, Wu et al. characterize the error patterns of individual 802.15.4 chip sequences in order to determine the channel conditions [18]. Similarly, Jamieson et al. implement a scheme in which they count the differences between the received and the known chip sequences to estimate the likelihood of a symbol being corrupt [8]. They then use this information, as part of a MAC protocol, to only re-transmit symbols that were likely corrupted. Dubois-Ferrière et al. combine successive alternating packets in order to infer the correct payload [2]. They show that this is feasible even when consecutive packets are broken, making the approach more robust than regular forward error correction. Hauer et al. propose to selectively retransmit parts of a packet during which there was a strong variation in received signal strength [5].

9 Conclusion

In this paper, we have described how corruption systematically affects symbols and packets in an outdoor 802.15.4 sensor network. We described a pattern in corruption that we attributed to the use of MSK demodulators, a specific tie resolution strategy when decoding, and a channel model with independent errors. These insights allowed us to formulate a novel probabilistic approach to recover information from corrupt packets. We showed that the approach reduces the uncertainty associated with a corrupt packet, and that it correctly assigns a high probability to the data that was actually sent. We will address systems aspects of our approach in future work and develop a concrete implementation of the proposed ideas.

We conclude that patterns in packet corruption in outdoor sensor networks can be understood, and that information may be recovered from some corrupt packets. All is not lost when it comes to corrupt packets, and therefore discarding all of them is unnecessarily wasteful.

References

[1] N. Baccour, A. Koubâa, L. Mottola, M. A. Zúñiga, H. Youssef, C. A. Boano, and M. Alves. Radio link quality estimation in wireless sensor networks: A survey. *ACM Trans. Sen. Netw.*, 8(4), Sept. 2012.

[2] H. Dubois-Ferrière, D. Estrin, and M. Vetterli. Packet combining in sensor networks. ACM SenSys '05.

[3] B. Han, L. Ji, S. Lee, B. Bhattacharjee, and R. R. Miller. Are All Bits Equal? Experimental Study of IEEE 802.11 Communication Bit Errors. *IEEE/ACM Trans. Netw.*, 20(6), 2012.

[4] D. Hasenfratz, O. Saukh, and L. Thiele. Model-driven accuracy bounds for noisy sensor readings. IEEE DCOSS '13.

[5] J.-H. Hauer, A. Willig, and A. Wolisz. Mitigating the Effects of RF Interference through RSSI-Based Error Recovery. EWSN '10.

[6] F. Hermans, O. Rensfelt, T. Voigt, E. Ngai, L.-A. Nordén, and P. Gunningberg. SoNIC: Classifying Interference in 802.15.4 Sensor Networks. IPSN '13.

[7] IEEE Computer Society. *802.15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs).*

[8] K. Jamieson and H. Balakrishnan. PPR: partial packet recovery for wireless networks. ACM SIGCOMM '07.

[9] L. Kong, M. Xia, X.-Y. Liu, M.-Y. Wu, and X. Liu. Data loss and reconstruction in sensor networks. IEEE INFOCOM '13.

[10] C.-J. M. Liang, N. B. Priyantha, J. Liu, and A. Terzis. Surviving Wi-Fi Interference in Low Power ZigBee Networks. ACM SenSys '10.

[11] S. Lin, J. Zhang, G. Zhou, L. Gu, J. A. Stankovic, and T. He. ATPC: adaptive transmission power control for wireless sensor networks. ACM SenSys '06.

[12] J. Notor, A. Caviglia, and G. Levy. CMOS RFIC Architectures for IEEE 802.16.4 Networks. Technical report, Cadence Design Systems, Inc., '03.

[13] T. Schmid. GNU Radio 802.15.4 En- and Decoding. Technical report, Department of Electrical Engineering, University of California, Los Angeles '06.

[14] F. Schmidt, M. Ceriotti, and K. Wehrle. Bit Error Distribution and Mutation Patterns of Corrupted Packets in Low-Power Wireless Networks. WiNTECH '13.

[15] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis. An empirical study of low-power wireless. *ACM Trans. Sen. Netw.*, 6(2), Mar. 2010.

[16] Texas Instruments Inc. CC2420 - 2.4 GHz IEEE 802.15.4, ZigBee-ready RF Transceiver. `http://www.ti.com/lit/gpn/cc2420`.

[17] H. Wennerström, F. Hermans, O. Rensfelt, C. Rohner, and L.-A. Nordén. A Long-Term Study of Correlations between Meteorological Conditions and 802.15.4 Link Performance. IEEE SECON '13.

[18] K. Wu, H. Tan, H.-L. Ngan, Y. Liu, and L. Ni. Chip Error Pattern Analysis in IEEE 802.15.4. *IEEE Trans. Mob. Comp.*, 11(4), 2012.

[19] X. Wu and M. Liu. In-situ soil moisture sensing: measurement scheduling and estimation using compressive sensing. ACM/IEEE IPSN '12.

[20] F. Xiong. *Digital Modulation Techniques*. Artech House, 2 edition, Apr. 2006.

[21] M. Zúñiga Zamalloa and B. Krishnamachari. An analysis of unreliability and asymmetry in low-power wireless links. *ACM Trans. Sen. Netw.*, 3(2), June 2007.

Recent licentiate theses from the Department of Information Technology

2013-006 Kristoffer Virta: *Difference Methods with Boundary and Interface Treatment for Wave Equations*

2013-005 Emil Kieri: *Numerical Quantum Dynamics*

2013-004 Johannes Åman Pohjola: *Bells and Whistles: Advanced Language Features in Psi-Calculi*

2013-003 Daniel Elfverson: *On Discontinuous Galerkin Multiscale Methods*

2013-002 Marcus Holm: *Scientific Computing on Hybrid Architectures*

2013-001 Olov Rosén: *Parallelization of Stochastic Estimation Algorithms on Multicore Computational Platforms*

2012-009 Andreas Sembrant: *Efficient Techniques for Detecting and Exploiting Runtime Phases*

2012-008 Palle Raabjerg: *Extending Psi-calculi and their Formal Proofs*

2012-007 Margarida Martins da Silva: *System Identification and Control for General Anesthesia based on Parsimonious Wiener Models*

2012-006 Martin Tillenius: *Leveraging Multicore Processors for Scientific Computing*

2012-005 Egi Hidayat: *On Identification of Endocrine Systems*

2012-004 Soma Tayamon: *Nonlinear System Identification with Applications to Selective Catalytic Reduction Systems*

UPPSALA
UNIVERSITET

Department of Information Technology, Uppsala University, Sweden