Bit Error Distribution and Mutation Patterns of Corrupted Packets in Low-Power Wireless Networks

Florian Schmidt, Matteo Ceriotti, Klaus Wehrle
Communication and Distributed Systems Group
RWTH Aachen University, Germany
{schmidt,ceriotti,wehrle}@comsys.rwth-aachen.de

ABSTRACT

It is well known that wireless channels produce higher bit error rates than wired connections. However, little knowledge exists about how bit errors are distributed within messages. In this paper, we present results from our experiments in an 802.15.4 sensor node testbed investigating the distribution of errors within erroneous frames. We identify three effects that can only partially be explained by coding and channel conditions: (1) errors are not independently distributed, but to a certain extent bursty, (2) coding leads to some bits being more stable than others, and (3) some content is inherently more stable than other during transmission. We discuss hypotheses on the origins of these effects and give some preliminary ideas on how to leverage them.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless Communication*

Keywords

sensor networks; IEEE 802.15.4; testbeds; measurements; bit error distribution within frames; coding

1. INTRODUCTION

Low-power wireless devices are slowly populating everyday scenarios interconnecting the physical world with the virtual one, realizing visions such as the Internet of Things. One of their enabling features is the ability to communicate over a wireless channel consuming little energy. However, this commonly results in unreliable links with hardly predictable behavior depending on the scenario. Facing such limitations and acquiring knowledge on the causes behind such unreliability are stepping stones to fulfill the risen expectations.

Unreliability is typically observed in low-power wireless networks only as the ratio of not correctly received frames or, in rare cases, bits. Therefore, corrupted messages are treated as lost, usually requiring additional energy overhead for retransmissions. This approach is also motivated by the common focus on message corruption that is caused by external dynamic interference, such as IEEE 802.11 [8] networks corrupting low-power IEEE 802.15.4 [7] transmissions. As a result, the common belief is that corruption of and inside packets is an erratic, not structural, effect in low-power networks.

In this work, we take a different, more in-depth perspective. We observe properties of corrupted messages in IEEE 802.15.4 systems without interference from coexisting networks. Instead of limiting ourselves to observing corrupted packets or bit error rates, we extend the analysis to identify how transmitted symbols are transformed upon corruption. This provides a unique perspective that has, to our knowledge, never been investigated before in 802.15.4 networks (Section 2).

After some preliminary study in a small indoor network, we executed measurements in an outdoor experimental setup (Section 3) composed of 20 devices employing a common CC2420 radio chip [3], which implements the IEEE 802.15.4 PHY standard. In this scenario, we gathered extensive data on corrupted messages for a long period of time, confirming the results from our initial small-scale investigation. The results (Section 4) clearly show that these systems have links where specific symbols are more likely to break than others, with structural, repeating mutation patterns.

We then try to discuss the reasons for the effects that we measured (Section 5) and give simple guidelines for software developers to exploit the identified higher stability of specific symbols. While we currently cannot identify all underlying causes, the results that we present in this work are likely to be observed in most deployments of this technology as they are, from our own experience, independent from the scenario and already measurable in small-scale networks. Therefore, we believe that the subject requires further investigation (Section 6) to increase the overall reliability of these systems at the benefit of the user.

2. RELATED WORK

The behavior of low-power wireless networks in real settings has been studied extensively in the last decade. The main focus was typically on the characterization of link properties in terms of correct packet reception rates as affected by environmental conditions. To study and exploit the causes of message failures, in particular in indoor scenarios, the impact of external interference on the observed RSSI signal has been investigated and generic error correction schemes have

WiNTECH'13, September 30, 2013, Miami, Florida, USA.
Copyright 2013 ACM 978-1-4503-2364-2/13/09 ...$15.00.
http://dx.doi.org/10.1145/2505469.2505475 .

been introduced to increase the link reliability. In this section, we discuss these approaches, focusing mainly on IEEE 802.15.4 low-power wireless communication, and relate them to our own investigation.

Link Characterization. Several experimental studies have identified the properties of communication links employing IEEE 802.15.4 radios. In [13, 16], the unreliability of low-power wireless is demonstrated both in terms of intermediate reception probabilities as well as asymmetries. The dependence of such properties upon the specific scenario at hand has also been demonstrated [10]. Finally, the same scenario may change over time together with its link characteristics, for example, as a consequence of temperature excursions [1]. While these works provide valuable insight based on experimental results, they exclusively address correctly received packets, treating not received and corrupted transmissions equally. Some prior work exists that gives insight into bit error distributions within frames for 802.11 networks. In particular, [15] analyzes an industrial 802.11b setup. Among other results, they briefly discuss bit error distributions within frames. A more in-depth analysis of sub-frame bit error distributions is given in [4]. However, their work focuses on the OFDM-based WiFi sub-standards, which is not directly comparable to the DSSS-based 802.15.4 transmission.

Interference Classification. Different works have focused on message corruption as caused by wireless interference from devices operating in the same frequency band. The focus was mostly at the level of RSSI as observed by a receiver under interference [2]. Such information can be exploited to classify the interference pattern and its source [5] to, then, take countermeasures, for example, by rescheduling communication. These approaches and studies use information related to possibly corrupted packets, but they limit their investigation to the RSSI fingerprint, without exploring how the actual message content is transformed by the interference.

Packet Error Correction Schemes. To face the problems of corrupted messages, different schemes have been proposed. In [9], the interplay between 802.15.4 and 802.11 networks is analyzed; based on the experimental observations, header redundancy and an effective forward error correction scheme are introduced to allow coexistence of the different network types. This study exploits the error bit distribution as caused by 802.11 networks on low-power 802.15.4 transmissions. As corrupted patterns may change over time, [12] proposes correction schemes capable of adapting to changing link characteristics, typical in indoor scenarios. However, we are not aware of any study that identifies and exploits significant mutation pattern in corrupted messages.

3. EXPERIMENTAL SETUP

In this section, we provide information about the setup of our measurements and observations.

3.1 Technology

In our investigation, we decided to use TelosB [11] wireless sensor devices, a common hardware platform for low-power embedded networks research. Most importantly for our work, they mount an on-board CC2420 radio [3], one of the currently most widespread chip implementing the IEEE 802.15.4 standard [7]. At the physical layer, the standard defines a DSSS (Direct Sequence Spread Spectrum) O-QPSK modulation, in the 2.4 GHz ISM band, with a nominal data rate of 250 kbps. Moreover, the devices we employed in our experimentation had an omni-directional inverted-F micro-strip antenna or an external 5 dBi one. The nodes were purchased at different times from different producers. Some were reused from previous deployments, while others were bought for use in the current deployment.

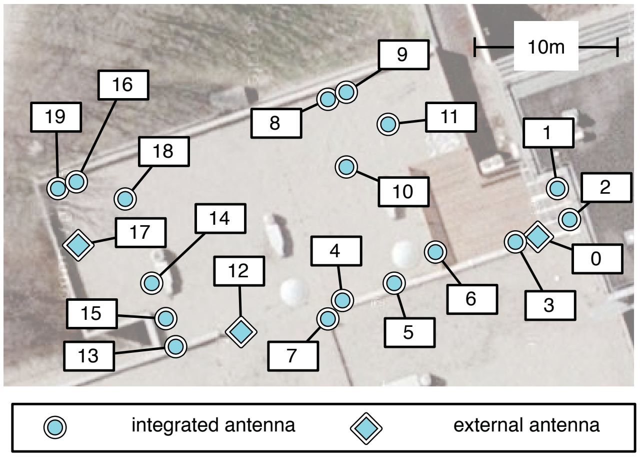

Figure 1: Outdoor testbed of 20 TelosB devices installed on the roof of our department and corresponding identifiers.

3.2 Deployment

We installed our devices in a 20-nodes testbed, as depicted in Figure 1. As mentioned before, some devices, nodes #0, #12 and #17, have external antennas, while all others have an integrated inverted-F micro-strip antenna. The nodes are connected via USB cables to routers so that a reliable back-channel is available for controlling each individual device, without affecting the wireless communication. We can, therefore, control experiments remotely and log extensive information on a remote server for later processing.

The nodes are installed on the roof of our department. The area is not accessible to people; the only dynamics in the environment are the ones related to weather conditions such as rain. All the devices are installed inside water-proof plastic boxes, which, in the general case, are placed on top of bricks. Nodes #9 and #16 are placed directly on the ground, and nodes #0, #3, #5 and #7 are installed at 2.60 m height on the facade of the adjacent building; finally, nodes #1 and #2 are at 2.60 m covered by eaves.

3.3 Software

In the measurements, our goal is to experiment with transmissions as directly observed by the radio, without any influence from higher layers in the network stack. For this reason, the devices do not execute any multi-hop routing or MAC protocol. Each node runs a simple TinyOS [6] application, forwarding messages received from the serial to the radio interface and vice versa. The message handling done by the radio chip takes care of adding a CRC to each outgoing message and verifying it upon reception. At reception time, the packet is validated, and typically discarded if the CRC does not match. Our application instructs the radio to pass on messages with failed CRCs, to collect erroneous messages and their content. The resulting message structure is depicted in Figure 2.

In order to control the messages sent over the radio and avoid possible message collision, which would affect the results, a single Java application running on a remote host is in charge of building MAC frames with a given payload and header, and of delivering them, via serial, to all of the TelosB nodes in the testbed. The internal message structure used for this delivery contains additional side information about which node should send the message, as well as the transmission power and the radio channel. The nodes can then adjust the channel and, in case they are the sender, send the message after a short delay to account for the time required by the potential receivers to switch channel. After delivering a message to a node, the Java application waits long enough for each node to receive and process the frame. It then requests each node to hand the received frame plus side information (RSSI, LQI, etc.) back to the central remote host for logging and off-line processing.

The Java application can be configured to execute specific experiment configurations and therefore send messages on a specific channel, cycling over different sizes, transmission powers, and senders. In order to control the actual content of the MAC layer frame payload, we allow for different patterns to be repeated, either filled with predefined constant values or randomly generated inside a defined interval. Finally, both the sent message and the received, potentially corrupted, packets, forwarded over the different testbed serials, are recorded in textual logs for later processing.

4. EXPERIMENTAL RESULTS

In the following, we will present the results from our experimental setup as described in Section 3. We will first give an overview over the setup and execution for the experiments we analyze, before going into the details of the evaluation.

We ran the experiments over the course of several weeks in December 2012 and January 2013. In each experiment, we instructed the nodes to take turns in sending a message, with the other nodes set to receive it. The inter-message interval was 500 ms. Even though the deployment was on the roof of a building and there was no physical influence on the deployment due to people moving between the nodes, there are WLAN access point installed in the building. Therefore, we focused our experiments on channel 26, which is outside of the spectrum allotted for 802.11 in Germany, to minimize the influence of interference. In addition to cycling through the nodes for sending duty, we also cycled through transmission powers. We picked three transmission powers chosen to provoke erroneous transmissions. The least powerful transmission power produced errors in nodes relatively close to the sender, while those messages were not sensed at all by far-away nodes. The highest power tended to produce erroneous messages in far-away nodes and correct receptions in close ones.

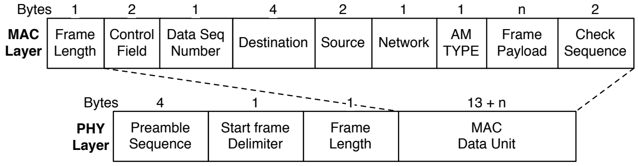

Figure 2: 802.15.4 message structure definition [7] with MAC layer fields as specified in the TinyOS implementation [14].

With regards to content, we sent different patterns for different experiments. In one experiment, we sent randomized payloads, that is, every time a message was sent, we randomly chose new content to fill the payload with. In other experiments, we set a fixed payload that was then sent over and over from each node to every other node. This was done to analyze the effects of content on error rates and distributions.

4.1 Error distribution within frames

In the following, we will look into the bit error distribution within received MAC frames. For these results, we aggregated all data sent between all sensor nodes. We then counted, for each bit position, how often it was broken. We normalized our bit error probabilities by only considering messages that had at least one broken bit in them; completely error-free messages were not considered for the calculation of bit error rates. This aggregation does not hide any connection-specific behavior between two nodes: whenever there were enough erroneous messages to produce statistically useful results, the patterns closely followed the general trend. The results can therefore be interpreted as representative for each connection.

In the following, we will present answers to the following questions: (1) Are errors more likely to occur towards the beginning or the end of a frame (influence of *position*)? (2) Are some bits more likely to break than others due to PHY modulation and coding (influence of *coding*)? (3) To what extent do bit errors occur in groups (*burstiness* of errors)? (4) Are some bit patterns more robust than others (influence of *content*)?

4.1.1 Influence of Position

To investigate the influence of position within a frame on error probability, we sent fixed-size frames with randomized payloads between all nodes in our testbed. Afterwards, we looked at the first bit in each frame and counted the number of times the bit arrived flipped. We did this for every bit in the frame. Therefore, Figure 3 shows the probability of each bit to break in the packet. Within each octet of bits, the least significant bit is plotted first and the most significant bit last.

Two results are apparent here: First, the beginning of the packet shows a much less regular pattern than the rest. This is because the first 96 bits contain the MAC header. Even when the payload is randomized, most fields in the header will stay fixed for every packet that is sent out and are therefore not randomized. We will discuss the striking irregularities in this area in a more general sense in Section 4.1.4 when we analyze the influence of content on the error rate. The fact that in the very beginning of the message, some bits do not show any errors at all is an artifact of our processing and the way the radio chip saves the received data in memory. The first field of the message contains the frame length (cf. Figure 2). If the length field is broken, the message can end up truncated in the node's memory, and some parts of the data structure holding it may contain uninitialized values from previous messages. Therefore, if

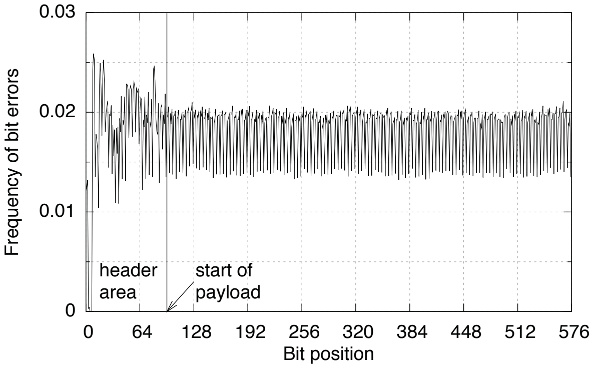

Figure 3: Influence of position on errors from 137570 erroneous frame receptions of randomized payloads. BER neither increases nor diminishes towards the end of the message.

the length field varied from the actual length by more than a couple of bytes and the frame was truncated, we discarded it during processing.

Second, within the payload, there is no sloping pattern that would indicate a tendency for later bits to break more or less often. While bit position has an influence on a small scale (which we will discuss in the next section), the bit error probability is roughly the same for bit 128 and bit 512.

4.1.2 *Influence of Coding*

Figure 3 already shows a noticeable variation in BER between adjacent bits. A magnified excerpt is shown in Figure 4. This sawtooth-like pattern has a period of 4 bits and repeats over the full length of the frame. We assume that this pattern is due to the coding of the message on the physical layer. The TelosB nodes used in our setup employ an 802.15.4 PHY [7] implementation. That means that each frame is encoded and spread via DSSS with specified chipping sequences. Each byte is split into two symbols of 4 bits each, by splitting the byte in half. Each 4-bit symbol is then mapped to a 32-bit chip sequence, which is then modulated with O-QPSK. Remember that, for our figures, we ordered the bits in each byte by increasing significance. That means that bit 384 is the least significant bit (LSB) and bit 391 the most significant bit (MSB) of byte 48. This conserves the PHY layer's order of putting information on the channel: later bytes are sent after earlier bytes, and within a byte, the four lower-significance bits are encoded and sent out before the higher-significance bits. The figure therefore shows that each 4-bit-group's MSB is significantly less likely to break than the others.

The 32-bit chip sequences used by 802.15.4 were designed to produce high Hamming distances to each other, so that even if some of the chip bits are flipped during transmission, the original symbol can be reconstructed. Figure 5 shows a graphical representation of the Hamming distances from each symbol's chip sequence to every other. (The minimum Hamming distance between two 32-bit chips is 12, and the maximum is 20. The diagonal line marks fields that are left empty because they have the same "from" and "to" fields.) This makes it easy to see that the chips follow a relatively regular pattern. High Hamming distances between symbols should mean that a mutation between them should be less likely than with lower distances.[1] Interestingly, our measurements do not match this reasoning: Since the MSB was more robust in our measurements, we would expect the top right and bottom left quadrants of Figure 5 to show higher Hamming distances. In fact, the opposite is true: these areas tend to show below-average distances.

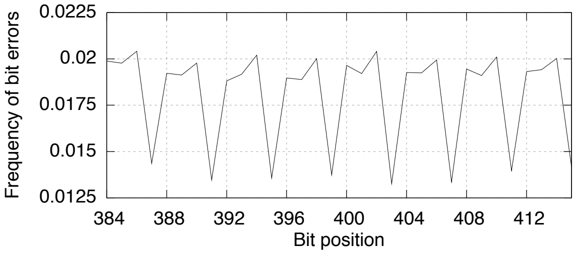

Figure 4: Magnified 32-bit excerpt from Figure 3. Within each 4-bit symbol, the MSB is significantly less likely to break than the other bits, leading to a noticeable "sawtooth" pattern.

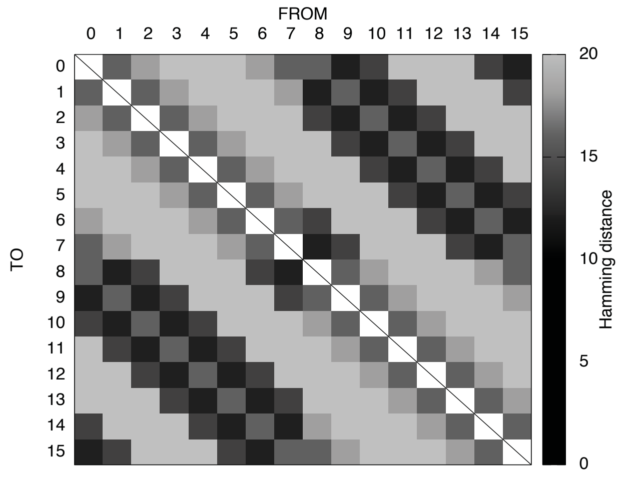

Figure 5: Hamming distances from every 4-bit symbol to every other symbol after coding them into 32-bit chips.

4.1.3 *Burstiness of Errors*

While it is generally accepted knowledge that bit errors are not independently distributed within frames, there is little insight into any actual interdependence. From our experiments, we analyzed the burstiness of errors, that is, how likely a burst of length n (exactly n errors in sequence) is, compared to single-bit errors.

For this, we chose from our experiments those links between nodes that had seen at least 1000 erroneous messages. This left us with 213 links (of 380, because each of our 20 nodes could receive data from every other node). We then counted the number of occurrences of n-bursts for each con-

[1] This assumes an independent distribution of errors. While this is not generally true for transmissions (and also not in our case, cf. Section 4.1.3), we can assume this in this special case, since we are looking at average bit flip probabilities over many frames and long periods.

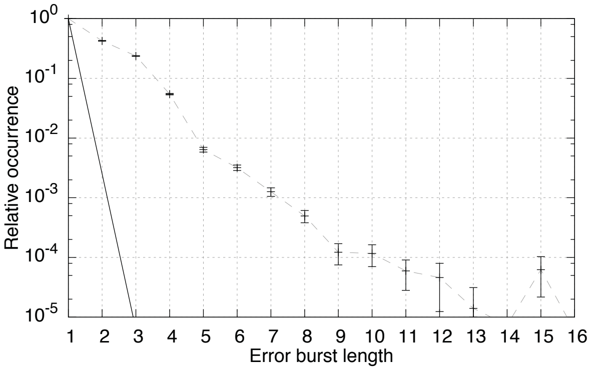

Figure 6: Burstiness of errors. From our experiments, we took 213 links that had at least 1000 erroneous frames each. For each, we counted how often bursts of specific lengths occurred and then normalized the number to the amount of single bit errors. The error bars denote 99% confidence intervals. Bursts longer than 10 bits occurred so rarely that the numbers are not reliable. The line on the left denotes the expected relative occurrence of bursts if errors were independently distributed.

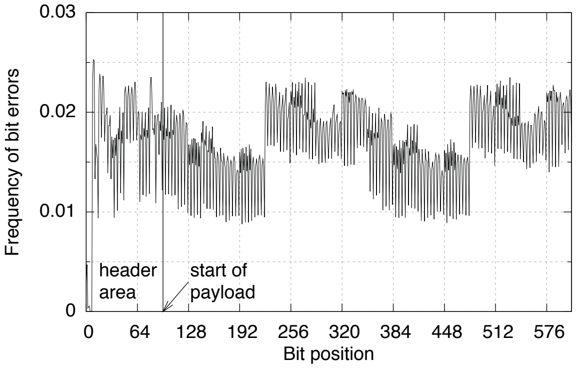

Figure 7: Influence of content on errors from 200390 erroneous frame receptions of static payloads. Some symbols are more likely to break than others.

nection, and then normalized the results by dividing each number by the number of 1-bursts (single bit errors) for that connection. That way, for each connection, a value of 0.5 for a burst of length n would denote that this n-burst occurred half as often as single bit errors, regardless of the actual number of errors in frames for that link.

Figure 6 shows the results from those 213 links. Error bars denote 99% confidence intervals. For lucidity, the data points are connected by a dashed line. The solid black line on the left side of the graph denotes the expected results if errors were independently distributed. For this, we counted the number of bit errors over all messages, calculated the average BER in those messages, and then calculated probability of a burst of length n occurring by computing $BER^n(1-BER)$, that is, the probability that n bits are erroneous and the following bit correct. There are several conclusions to draw from these results. (Note that bursts longer than 10 bits occurred so rarely that it is not possible to draw substantiated conclusions from them.)

First, the relative occurrences of n-bursts are remarkably stable. Even though the average BER varied by a factor of 25 between the highest- and lowest-BER link (0.07% vs. 1.8%), the results were so similar that even 99% confidence intervals are hardly visible for smaller n. Second, bit errors indeed are not independently distributed within frames. Bursts occur more often than they should if errors were independently distributed.

Third, there is a noticeable drop between 4-bursts and 5-bursts. This is the border between bursts that can be confined to one symbol, and bursts that spread more than one symbol. (Note that bursts of lengths 2 to 4 can also spread two symbols, but do not have to.) This makes sense considering the coding, because to corrupt bits in two symbols, a much higher number of chip bits has to be corrupted (a minimum of 7 in each symbol, to overcome the minimum Hamming distance of 12 between two symbols). If that is true, there should be a similar drop between 8-bursts and 9-bursts. From our data, it is hard to draw conclusions in that area, however, because those already occur very rarely.

Fourth, 4-bursts occur much more rarely than 3-bursts. Even relatively speaking, the difference between those two is much higher than between 2-bursts and 3-bursts. This matches our results from Section 4.1.2. Since every 4-bit symbol contains one bit that breaks significantly less often, this bit can act as a “burst breaker” that ends a burst at length 3.

4.1.4 *Influence of Content*

During our experiments, we not only transmitted random payload data, but also investigated bit patterns, to analyze whether the content of a message has influence on the error distribution. In one example that we present here, we sent messages of 64 bytes payload, with a fixed pattern consisting of the hexadecimal values 0x0000, 0x1111, 0x2222, . . . , 0xFFFF; this means that each symbol was repeated 4 times before switching to the next one. This produces 32 bytes, so the overall pattern was repeated once more. This fixed pattern was broadcast over and over from each node, and every reception by another node was recorded. The results and effects shown in the following could also be seen in experiments with other patterns; we experimented with a number of patterns, and only present one setup here for brevity, and because it forms a representative example.

The difference between the various 4-bit symbols is very noticeable. Symbols with a most significant bit (MSB) of 1 tend to break more often than those with an MSB of 0. These results came as a surprise to us. We cannot explain such an imbalance from the coding. Since the coding tries to spread the Hamming distances as evenly as possible between symbols, the differences should be much less pronounced, and not skewed towards high-value (MSB=1) symbols.

This effect is even larger than the difference between bits within symbols: The coding effect, while still very noticeable, does not conceal the differences. Especially the jump between bit pattern 0x7777 and 0x8888 is so striking that the bit most likely to break in the first pattern is still more robust than the bit least likely to break in the second pattern. Figure 8 shows a magnified excerpt of Figure 7. There are clear differences between the error probabilities, depending on which symbol is transmitted. Each symbol forms a

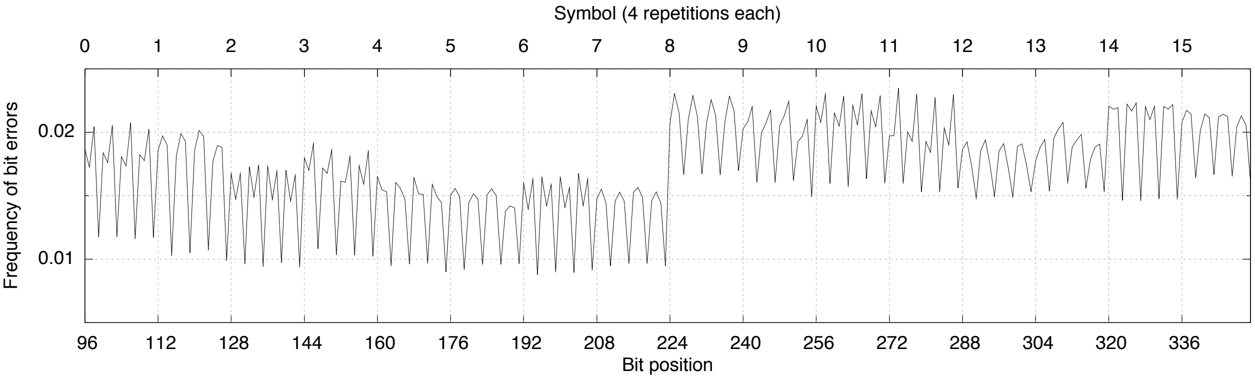

Figure 8: Magnified 32-byte excerpt from Figure 7. Some symbols are more likely to break than others. In general, low-value symbols (MSB=0) are more stable than high-value symbols.

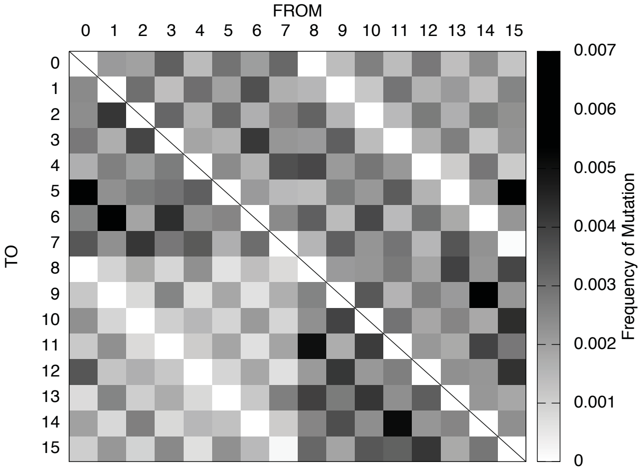

Figure 9: Frequency with which a symbol mutates (decodes into another symbol) due to errors. Low-value symbols (0-7) have a much smaller probability to mutate into high-value symbols (8-15), contributing to their overall higher stability.

characteristic error pattern that is clearly different from the other symbols.

The fact that coding still shines through, but is in part masked by content when it comes to effect on error probability, can be seen in Figure 9. For this figure, we took all results from all nodes in all experiments, and looked at how often each symbol broke and mutated into another symbol. The color pattern was chosen to match Figure 5: low Hamming distances (darker in Figure 5) should, in theory, result in higher error rates (darker in Figure 9). Therefore, if coding were the major influence, the pattern should look similar to Figure 5.

However, there is no apparent correlation. In fact, the only noticeable correlation is a negative one: a single bit flip of the MSB is quite unlikely (which matches our results from Section 4.1.2), as can be seen by the light diagonal lines from $(8, 0)$ to $(15, 7)$ and from $(0, 8)$ to $(7, 15)$. These, however, are mutations between symbols that have a comparatively low Hamming distance to each other.

The most noticeable effect in Figure 9 is probably the lower values in the bottom left quadrant. This means that it is relatively unlikely for low-value symbols (0–7) to mutate into high-value symbols (8–15). This is the main contributor to the result shown in Figure 8: low-value symbols are more stable than high-value symbols.

5. DISCUSSION

In this section, we first want to compare our results to those of [4], the most in-depth analysis of bit error distributions within frames available in the literature.

With respect to position within a frame, the authors noticed what they called a "sloping pattern" that led to a steady increase in BERs towards the end of the packet. While we did not see this effect in our setup, this might simply be due to the length of the frames. WLAN frames tend to be much larger than low-power wireless frames, which are allowed by the standard to have a maximum size of 127 bytes. In fact, the frames Han et al. analyzed had sizes of more than 1000 bytes. In several setups, their sloping pattern did not start until after our frames would have already ended.

They also noted the effect that coding and modulation have on repeating patterns in the error distribution. While their work focuses mostly on OFDM, they present also some results for DSSS. The sawtooth pattern was witnessed by them, too, in both modulation systems. We agree with them that this strong correlation to coding parameters suggests an interaction between coding and bit error distribution.

With regard to content, the authors do not provide any results, since they used a simple all-zeroes payload for the bulk of their experiments. Our results are indeed puzzling to us. When we first noticed this influence of content on error probability in a small setup, we suspected a hardware error, or a mere coincidence. However, we saw these effects in all our test runs. The effects were the same in our outdoor testbed, which contains TelosB nodes from several different production runs. As such, we can rule out coincidence, setup-specific effects, and hardware issues that are specific to manufacturing.

At this point in time, we cannot provide any satisfying or substantiated explanation. We tried to rule out all effects that would be due to special circumstances of our node deployment. One possibility that we cannot exclude is that the radio chip used on the TelosB nodes simply produces less robust wireless output for some input symbols, for example during modulation. Even if this should be the case, we still consider our results important, due to the widespread use of the CC2420 radio chip in real systems. These deployments should then all experience this behavior.

Even though we cannot give certain answers to why these effects occur, the identification alone can give some suggestions for further work exploiting those characteristics. For example, in cases where only few different values have to be saved in data fields of larger size (e.g., a deployment with just 2 or 3 message types that are encoded in the 8-bit AM type header field), it makes sense to choose values that will be encoded into more stable symbols. In the case of error-tolerant payloads (i.e., messages are not discarded because it is hoped the content can still be of use, even if partially erroneous), content should be saved into the message in ways such that the most likely mutations lead to the least distortion in the received results. Moreover, the identified mutation patters could serve as basis for refined link models, significantly improving the adherence of low-power wireless simulation to real-world behavior.

Finally, knowing these characteristics and the different robustness of symbols, it might be possible to create an additional "coding" that translates high-BER symbols into low-BER symbols. Of course, this coding would introduce additional messaging overhead (which leads to higher energy consumption for larger frames, and an increased frame error rate), so the tradeoff between these two will require scrutiny and case-by-case consideration of advantages and disadvantages.

6. CONCLUSION

In this paper, we presented results from our study of bit error distributions within erroneous frames for 802.15.4-based low-power networks. Our analysis makes three effects manifest: errors are not evenly distributed, but show bursty tendencies; within each 4-bit symbol that is coded together for transmission, the most significant bit (MSB) is remarkably less likely to break; some symbols are more stable than others, with low-value (MSB=0) symbols in general being more stable than high-value (MSB=1) symbols.

While we cannot provide satisfying explanations for some of these effects, we discussed possible contributors. Furthermore, we consider these results important and relevant, since they give insight into the actual distribution of errors, and could be used to improve the robustness of data exchange in low-power wireless networks. In addition, these results can be used to create more accurate error models for network simulations. In the future, we plan to investigate additional hardware platforms and deployment scenarios to gather more insight into the generality of our results. We will then explore several possibilities to exploit the acquired knowledge in novel coding schemes.

Acknowledgments

We thank Tobias Suhrborg for support in the early phases of this work. This research is funded by the DFG Cluster of Excellence on Ultra High-Speed Mobile Information and Communication (UMIC) and the Alexander von Humboldt Foundation.

7. REFERENCES

[1] C. A. Boano, J. Brown, N. Tsiftes, U. Roedig, and T. Voigt, "The impact of temperature on outdoor industrial sensornet applications," *IEEE Transactions on Industrial Informatics*, vol. 6, no. 3, pp. 451–459, aug 2010.

[2] C. A. Boano, T. Voigt, C. Noda, K. Römer, and M. A. Zúñiga, "JamLab: Augmenting sensornet testbeds with realistic and controlled interference generation," in *Proceedings of the* 10^{th} *IEEE International Conference on Information Processing in Sensor Networks*, ser. IPSN '11, apr 2011, pp. 175–186.

[3] Chipcon Tech., "CC2420 Datasheet," focus.ti.com/docs/prod/folders/print/cc2420.html.

[4] B. Han, L. Ji, S. Lee, B. Bhattacharjee, and R. Miller, "All Bits Are Not Equal – A Study of IEEE 802.11 Communication Bit Errors," in *Proceedings of the Twenty-Eigth Annual Joint Conference of the IEEE Computer and Communications Societies*, ser. INFOCOM '09. IEEE, Apr. 2009, pp. 1602–1610.

[5] F. Hermans, O. Rensfelt, T. Voigt, E. Ngai, L.-A. Norden, and P. Gunningberg, "SoNIC: classifying interference in 802.15.4 sensor networks," in *Proceedings of the 12th international conference on Information processing in sensor networks*, ser. IPSN '13. New York, NY, USA: ACM, 2013, pp. 55–66.

[6] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System architecture directions for networked sensors," in *Proceedings of the 9th international conference on Architectural support for programming languages and operating systems*, ser. ASPLOS IX. New York, NY, USA: ACM, 2000, pp. 93–104.

[7] *Low-Rate Wireless Personal Area Networks (LR-WPANs)*, IEEE Std. 802.15.4, 2011.

[8] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, IEEE Std. 802.11, 2012.

[9] C.-J. M. Liang, N. B. Priyantha, J. Liu, and A. Terzis, "Surviving wi-fi interference in low power zigbee networks," in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '10. New York, NY, USA: ACM, 2010, pp. 309–322.

[10] L. Mottola, G. P. Picco, M. Ceriotti, Ş. Gună, and A. L. Murphy, "Not all wireless sensor networks are created equal: A comparative study on tunnels," *ACM Trans. Sen. Netw.*, vol. 7, no. 2, pp. 15:1–15:33, Sep. 2010.

[11] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," in *Proceedings of the 4th international symposium on Information processing in sensor networks*, ser. IPSN '05. Piscataway, NJ, USA: IEEE Press, 2005.

[12] J. Singh and D. Pesch, "Towards energy efficient adaptive error control in indoor wsn: A fuzzy logic based approach," in *Proceedings of the 2011 IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems*, ser. MASS '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 63–68.

[13] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis, "An empirical study of low-power wireless," *ACM Transactions on Sensor Networks*, vol. 6, no. 2, pp. 16:1–16:49, Mar. 2010.

[14] TinyOS, "TEP 111: message_t," http://www.tinyos.net/tinyos-2.1.0/doc/html/tep111.html.

[15] A. Willig, M. Kubisch, C. Hoene, and A. Wolisz, "Measurements of a wireless link in an industrial environment using an IEEE 802.11-compliant physical layer," *IEEE Transactions on Industrial Electronics*, vol. 49, no. 6, pp. 1265–1282, Dec. 2002.

[16] M. Zúñiga and B. Krishnamachari, "An analysis of unreliability and asymmetry in low-power wireless links," *ACM Transactions on Sensor Networks*, vol. 3, no. 2, Jun. 2007.