Channel-Dependent Forward Error Correction for IEEE 802.15.4 O-QPSK

Fabian Graf, Thomas Watteyne, Filip Maksimovic, Michael Villnow

▶ **To cite this version:**

Fabian Graf, Thomas Watteyne, Filip Maksimovic, Michael Villnow. Channel-Dependent Forward Error Correction for IEEE 802.15.4 O-QPSK. SmartIoT 2024 - 8th IEEE International Conference on Smart Internet of Things, Nov 2024, Shenzhen, China. hal-04724725

HAL Id: hal-04724725
https://hal.science/hal-04724725v1

Submitted on 7 Oct 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Channel-Dependent Forward Error Correction for IEEE 802.15.4 O-QPSK

Fabian Graf
Siemens AG
Siemens Technology
Erlangen, Germany
fabian.graf@siemens.com

Thomas Watteyne
Inria
AIO Team
Paris, France
thomas.watteyne@inria.fr

Filip Maksimovic
Inria
AIO Team
Paris, France
filip.maksimovic@inria.fr

Michael Villnow
Siemens AG
Siemens Technology
Erlangen, Germany
michael.villnow@siemens.com

***Abstract*—IEEE 802.15.4 has become the de facto standard for low-power wireless systems in Industrial Internet of Things (IIoT) applications. Time Slotted Channel Hopping (TSCH) is used in the O-QPSK PHY version to improve reliability and enhance resistance to external interference. In this work, we introduce a MAC layer-based Forward Error Correction (FEC) scheme that remains compliant with the standard while being lightweight, due to its use of multiple smaller component codes, which results in low decoding complexity. This scheme is particularly suited for sensitive applications that seek to improve packet delivery ratio (PDR) at the cost of a lower information rate. We explore Reed-Solomon codes as component codes, analyzing the scheme's efficiency from a power consumption standpoint and its effectiveness in correcting erroneous packets. Testbed experiments demonstrate a significant boost in reliability, with the scheme typically recovering from 50% of all erroneous packets, though at a reduced information rate of 60%. Additionally, we highlight the benefits of adaptively adjusting the code rate across channels during runtime, as error patterns can vary not only over time but also depending on the channel.**

I. Introduction

Many popular wireless standards for Smart Home applications, such as ZigBee, Thread, and Matter, as well as those used in Industrial Internet of Things (IIoT) – including WirelessHART and 6TiSCH – are built on the IEEE 802.15.4 standard at the PHY and MAC layers. The latest revision of the standard, IEEE 802.15.4-2020 [1], covers a set of 19 different PHYs operating at different frequency bands and using various modulation schemes in order to satisfy the requirements of their dedicated area of application. Among these PHYs, the Offset-Quadrature Phase Shift Keying (O-QPSK) PHY using 2.4 GHz frequency band is by far the most commonly used one and in literature IEEE 802.15.4 always refers to this PHY, if not otherwise stated [2]. This O-QPSK PHY has firstly been introduced in IEEE 802.15.4-2006 [3] and has not undergone major modifications since.

IEEE 802.15.4 outperforms other wireless standards for low-power systems thanks to its high reliability and energy efficiency of the wireless *motes* arranged in a mesh topology. An essential key to achieving these features is time slotted channel hopping (TSCH), which has been included at first as an amendment in IEEE 802.15.4e [4] and later in its entirety into IEEE 802.15.4-2015 [5]. The idea of TSCH is to cut time into time slots and the available frequency band into channels. The nodes in the network consequently transmit and receive based on a communication schedule at a certain time and frequency. By iteratively changing channels, frequency diversity can be exploited to combat the two most common reasons for a wireless transmission to fail: external interference and multipath fading and attenuation (MFA).

While several other PHYs in IEEE 802.15.4 employ Forward Error Correction (FEC) schemes to provide a higher robustness against transmission errors, the O-QPSK PHY does not feature the use of FEC. The goal of FEC is not only to detect erroneous bits, but also to correct them by decoding the redundancy bits that are sent on top of the actual message. This does result in a lower information data rate. FEC is prevalent in all sorts of communication standards and is often directly implemented in hardware (HW) for performance reasons. In order to stay standard-compliant, we rely on a FEC strategy for the IEEE 802.15.4 O-QPSK PHY, which performs FEC in software (SW) on the MAC Layer.

The choice of the FEC scheme is a trade-off between different code parameters such as code length, decoding complexity and code rate. We show, based on 500,000 transmitted frames in a real world experimental testbed, that this choice may change during runtime as it also changes with the employed channel determined by the TSCH schedule. The contribution of this work is the idea of using an adaptive channel-dependent FEC scheme to significantly increase the reliability of the wireless transmission and consequently lower the overall power consumption of the motes by reducing the number of retransmissions. While earlier studies on FEC in IEEE 802.15.4 exclusively focus on numerical simulations over theoretical channel models, the mentioned FEC strategy is implemented on a microcontroller unit (MCU) and the arising overhead in terms of power consumption and latency is investigated as part of this work. The major design premises of the FEC scheme are to be as lightweight as possible, i.e. minimize the number of decoding calls, and stay standard compliant. This allows the presented strategy to be directly applicable to all IEEE 802.15.4 devices. By selecting specific FEC parameters that yield an information rate of just 60%, it becomes possible to recover from errors in 50% of all corrupted packets. This makes the scheme particularly well-suited for applications where both latency and reliability are

critical.

The remainder of this paper is organized as follows. Section II presents related work on FEC for the IEEE 802.15.4 O-QPSK PHY. Section III introduces the FEC strategy used throughout this paper. Section IV gives insights to the experimental setup used for the evaluation of the FEC scheme. Section V highlights the decision-making factors for the parameters of the FEC code. Section VI deals with the evaluation in terms of power consumption and packet salvation ratio. Section VII contains practical implementation hints on the use of FEC in an adaptive channel-dependent fashion. Section VIII concludes the paper and presents outlook on future work.

II. Related Work

The first effort towards FEC for the IEEE 802.15.4 O-QPSK PHY was published by Yu et al. [6]. The authors present FEC on the MAC layer based on relatively small linear block codes. Encoding is proposed in two ways. Either by encoding blocks of the linear block code length and appending redundancy directly after the information bits, or by adding redundancy at the end of the payload field, keeping the message readable for nodes without FEC. While this work does not reveal any statistics on bit error rates, it contains a comparison of different codes in terms of memory footprint and processing time.

The same authors later refine the FEC scheme by introducing separate encoding of the MAC header and payload [7]. The goal is to decrease processing at intermediate hops since just the Frame Check Sequence (FCS) needs to checked. Furthermore, theoretical results for the packet error rate per signal to noise ratio (SNR) are presented for the Additive White Gaussian Noise (AWGN) channel and Rayleigh fading channel.

Setting up on this FEC strategy, the same authors also present an adaptive algorithm, which switches between codes of different error correcting capability based on a Markov chain model [8]. The idea is to switch to a more powerful code when the packet delivery ratio (PDR) drops below a fixed level. Again, theoretical results for this adaptive algorithm are shown for the Rayleigh fading channel.

Barac et al. [9] provide an in-depth bit- and symbol-level analysis of IEEE 802.15.4 transmission errors in industrial environments. They illustrate typical bit-error patterns resulting from external interference and MFA, based on extensive packet transmissions in real-world industrial testbeds, highlighting the properties of error bursts. Their findings are utilized to develop FEC coding schemes combined with interleaving.

The same authors also leverage this error pattern information to introduce the Lightweight Packet Error Discriminator (LPED) method, which distinguishes between errors caused by MFA and external interference [10], [11].

The same authors also introduce PREED [12]: a Packet REcovery scheme by Exploiting the Determinism in IEEE 802.15.4 networks. The idea of PREED is to use the knowledge of the content of the header fields in advance to recover bytes in the corrupted packet. In particular, using the aforementioned a priori knowledge in combination with FEC can help recover the unknown bytes belonging to the same codeword as the known ones.

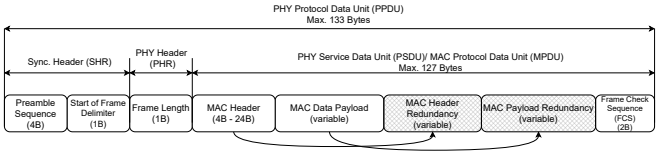


Figure 1: Forward Error Correction on the MAC Layer compatible with the IEEE 802.15.4 frame structure.

III. Forward Error Correction for the IEEE 802.15.4 O-QPSK PHY

The design of the FEC scheme used in this paper builds upon the approach presented by Yu et al. [6]. Instead of using one large FEC code to encode the entire frame, the strategy is to use multiple component codes, i.e. linear block codes with relatively small code lengths, to encode the MAC header and payload and add the resulting redundancy to the end of the payload field, as shown in Fig. 1.

Based on the MAC header, MAC payload, and the corresponding redundancy fields, a Cyclic Redundancy Code (CRC) is computed and appended as an FCS field right after the redundancy block. This resulting structure, the MAC protocol data unit (MPDU), is at most 127 bytes long and is passed as a PHY service data unit (PSDU) to the PHY layer. Subsequently, a frame length field (1 B) is added just before the PSDU. Finally, a preamble sequence (4 B) and an SFD (1 B) are prefixed as synchronization header fields. The resulting complete frame, the PHY protocol data unit (PPDU), reaches a maximum length of 133 bytes before being processed by the Direct Sequence Spread Spectrum (DSSS) mapper. DSSS maps each 4-bit hexadecimal symbol into one of 16 nearly-orthogonal pseudo-noise (PN) code sequence, each 32 chips long. The resulting 32 chips per symbol are then modulated onto a half sine pulse using O-QPSK. The carrier frequency f_c is determined by the selected IEEE 802.15.4 channel number (11 to 26) and is calculated as:

$$f_c = 2405 + 5(k - 11) \text{ MHz, for } k \in \{11, \ldots, 26\}$$

After demodulation, the received chip sequence is forwarded to the DSSS de-mapper. A maximum likelihood (ML) decoder determines the closest match by selecting the PN code sequence with the minimum Hamming distance to the received chip sequence from among the 16 possible sequences. As the number of Chip Errors per PN-Code (CEPP) increases, the likelihood that the ML decoder in DSSS selects an incorrect PN sequence grows [13]. To handle these errors, after removing the PHY header fields, a CRC check is performed. If the CRC check fails, the header is decoded using the header redundancy and another CRC check is executed. If the recalculated CRC still does not match the FCS, the payload is decoded using the payload redundancy. A final CRC check is conducted, and if the errors persist, the frame is discarded, and the packet is re-transmitted according to the automatic repeat

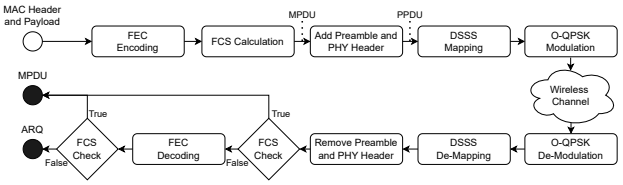


Figure 2: Communication chain of an IEEE 802.15.4 frame involving Forward Error Correction.

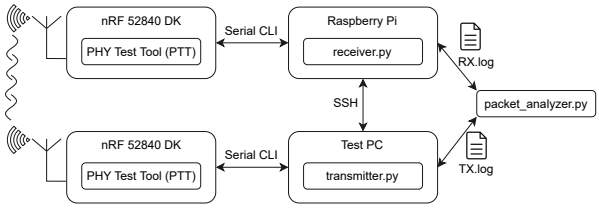


Figure 3: Test Setup consisting of nRF boards running a PHY Test Tool based on Zephyr.

request (ARQ) principle. The entire transmission process is illustrated in Fig. 2.

The rational for the choice of this FEC scheme are twofold: staying standard compliant and minimizing the number of decoding calls.

There are multiple reasons why maintaining compliance with the standard is crucial. *First*, the chosen FEC scheme does not violate the IEEE 802.15.4 frame structure, as the payload field may be filled individually and the other PHY layer fields remain untouched. *Second*, encoding does not include the preamble, start of frame delimiter (SFD) and PHY header, which are essential to ensure a correctly timed sampling process of the frame at the receiver side. *Third*, the way of calculating the FCS is unchanged. The computation of the FCS during the receive process is usually done in HW with insignificant cost in terms of latency and power consumption. Consequently, we do not want to discard the advantage of detecting corrupted frames immediately, i.e. without the need for decoding. *Fourth*, encoding is performed before DSSS. Thus, the beneficial orthogonal properties of the PN code sequences are not destroyed.

There are also several reasons why we choose an FEC scheme that minimizes the number of decoding calls. *First*, the systematic encoding structure. Systematic encoding means producing codewords with non-scrumbled information and redundancy bytes. This allows to place the redundancy of all encoded blocks to the end of the payload field. Besides the advantageous feature of improved readability of the payload, the decoder only needs to be called in case of failed CRC check. *Second*, separate header and payload encoding reduces the number of decoder calls. This is because a trivial CRC check can be done after header decoding and may already resolve all errors in the packet and would make decoding the remaining payload obsolete.

IV. Experimental Setup

To collect a large set of error patterns and evaluate the performance of the presented FEC scheme, we set up an experimental testbed based on the nRF52840-DK board from Nordic Semiconductor [14]. The MCU features a 64 MHz Arm Cortex-M4 and a IEEE 802.15.4 compliant 2.4 GHz radio. The testbed is installed inside the Siemens Sensor Application Lab. This is a challenging RF environment consisting of multiple office routers and sensor systems causing external interference and various objects, particularly metallic ones, that are responsible for the bouncing off echoes causing MFA. The radio transmit power can be switched between -20 dBm and +8 dBm, we drive it exclusively at 0 dBm being the default value. The receiver sensitivity is -100 dBm. The nRF52840's USB port is utilized to control the firmware running on the chips, which activates the radio during testing.

The firmware is based on a Nordic sample application known as IEEE 802.15.4 PHY Test Tool (PTT) [15]. The PTT operates as an application within the Zephyr Operating System, and is managed via a command line interface (CLI). It provides a comprehensive set of functions to experiment with the IEEE 802.15.4 radio, including setting transmit power and switching between receive and transmit modes. The PTT allows for switching between the 16 different IEEE 802.15.4 channels. To ensure both transmission and reception occur on the same channel, the Python test automation scripts coordinate a common channel via SSH. In our tests, we transmit one packet every 500 ms, which leaves enough time to do proper logging. We switch channels after every 20 packets to balance generating a substantial dataset and emulating a TSCH-like behavior as closely as possible. The device in transmit mode creates random payloads and encodes them into packets with a maximum size of 127 B. The receiver device continuously activates its radios to capture all incoming packets. In case of a failing CRC, the device starts the decoding routine trying to resolve the transmission errors. At the conclusion of the test, the log files from both the transmitter and receiver are analyzed using a Python-based packet analyzer script. This script compares transmitted and received payloads and identifies correct and erroneous packets along with meta-information, such as the receive signal strength indicator (RSSI) and the CRC check result from the receiver PTT. The complete test setup is illustrated in Fig. 3.

By default, packets failing the CRC check would be immediately discarded according to the standard. To counter this, we modify the radio driver and the PTT to retain the packet and perform the decoding routine as outlined in the flow diagram in Fig. 2. For the encoder and decoder implementations of the Reed-Solomon (RS) code, we refer to the appendix of the channel coding book by Morelos-Zaragoza [16].

V. Choice of FEC Parameters

The proposed FEC scheme is composed of several smaller linear block codes, known as component codes. From Section II, we know that RS codes are particularly well suited.

k	13	11	9	7	3
t	1	2	3	4	5

Table I: Error correcting capability t of a RS$(n = 15, k)$ code.

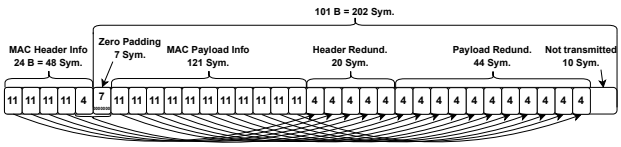


Figure 4: Encoding using a RS$(n = 15, k = 11)$ code.

Since IEEE 802.15.4 radios typically do not convey any soft information on the received symbols – which are used by more powerful soft-decision decoders – we rely on hard-decision decoders and the efficient decoding algorithms for RS codes. In this section, we determine what code length n, error correcting capability t and interleaver structure to use.

A. Code Length and Code Rate

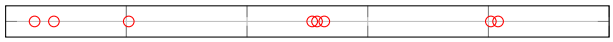
The code length n of the linear block code suited for the use in our scheme is bounded by the maximum size of the MAC header and MAC payload, respectively. One way to obtain linear block codes of exactly this length is shortening existing longer codes. According to the channel coding theorem, it is also recommended to use a large code length, since they provide better code rates than smaller codes assuming the same error correcting capability [17]. However, packet size might not always be fully utilized and decoding complexity grows with code length. Since DSSS takes 4 bit long symbols as input and RS codes with length $n = 15$ operate in the 2^4 Galois field, RS$(n = 15, k)$ codes are chosen throughout this paper. As a result, no further conversions are needed, and they provide an optimal balance between decoder complexity, fragmentation flexibility, and error correction performance. Depending on the code dimension k, i.e. the amount of information symbols, RS$(n = 15, k)$ codes guarantees to correct between $t = 1$ and $t = 5$ errors, as stated in Table I.

By fragmentation flexibility, we mean that the code does not "waste" too many bytes through zero padding in case the MAC header or MAC payload length are not multiples of k. For larger values of k this risk grows and the overall scheme's code rate becomes worse. A numerical example for the encoding procedure using a RS$(n = 15, k = 11)$ code of a frame with maximum header and payload size is shown in Fig. 4. Zero-padding is necessary and the available frame size can not be fully used. This results in an effective code rate of $R_{\text{eff.}} = \frac{48+121}{250-10} \approx 0.70$ in this case.

But in general, the effective code rate $R_{\text{eff.}}$ of this FEC scheme for an IEEE 802.15.4 frame size of $N_{\text{frame}} = 250$ symbols can be computed analytically, too. The number of not transmitted symbols is

$$N_{\text{not transmitted}} = N_{\text{frame}} \mod n.$$

(a) Multipath Fading and Attenuation (MFA) Pattern



(b) External Interference Pattern

Figure 5: Typical Error Patterns in IEEE 802.15.4.

The number of zero padding symbols required to encode the MAC header ($N_{\text{header, max}} = 48$ symbols) completely is computed as

$$N_{\text{zero padding}} = (k - (N_{\text{header}} \mod k)) \mod k$$

and the number of header redundancy symbols as

$$N_{\text{header, red.}} = \left\lceil \frac{N_{\text{header}}}{k} \right\rceil \cdot (n - k).$$

The amount of payload information symbols N_{payload} that can be placed inside the MAC payload field is then given by

$$N_{\text{payload}} = \left\lfloor \frac{N_{\text{frame}} - N_{\text{header}} - N_{\text{not transmitted}} - N_{\text{zero padding}} - N_{\text{header, red.}}}{n} \right\rfloor \cdot k.$$

Finally, the effective code rate can be determined as

$$R_{\text{eff.}} = \frac{N_{\text{header}} + N_{\text{payload}}}{N_{\text{frame}} - N_{\text{not transmitted}}}.$$

B. Error Correcting Capability

However, the number of potentially correctable errors does not sum up over all encoded blocks. Although the RS$(n = 15, k = 11)$ can correct $t = 2$ errors, the overall scheme shown in Fig. 4, which consists of in total 16 codewords, does not guarantee to correct $16 \cdot t = 32$ symbol errors. Moreover, the symbol error positions inside the frame are decisive. These so-called error patterns are strongly related to the environment. From real-world testbed experiments, it becomes clear that these error patterns can surprisingly unambiguously be classified in two groups [9]. Typical patterns from these groups are shown in Fig. 5. Packets corrupted by multipath fading and attenuation (MFA) are characterized by sparsely distributed errors and a generally low absolute number of symbol errors. However, those packets suffering from external interference show long error bursts.

Obviously, packets showing signs of external interference need to be encoded using codes with higher error correcting capability. Error patterns vary over time due to physical changes in the environment. However, they also vary with the used frequency associated with the channel of the TSCH schedule [18]. Consequently, we propose to dynamically change the code and its error correcting capability during runtime in an adaptive fashion based on the reported error patterns.

C. Interleaver

Increasing the error correcting capability is not enough to cope with long error bursts. The bursts usually pervade multiple consecutive encoded blocks and therefore easily exceed the error correcting capability of a single component code. When there are more than t errors in a single block, decoding fails and the whole frame cannot be recovered. We therefore recommended to shuffle the symbols inside the packet according to a fixed permutation algorithm. This process is called *interleaving* and also proposed in the IEEE 802.15.4-2020 standard for the Smart Utility Network (SUN) Frequency Shift Keying (FSK) PHY [1]. Although loosing the property of easy readability of the packet, this form of interleaving comes at negligible cost.

From the experiment using the setup described in Section IV, we analyze all erroneous packets, i.e. packets with failed CRC check, in terms of their symbol error positions. The packet size is 250 symbols (125 B). To identify error bursts, an analyzer examines the error positions, checking for consecutive occurrences. We permit a tolerance of up to 4 interruption positions inside a burst to count the subsequent errors to still the same error burst. The results are illustrated in Fig. 6 as boxplots for each channel. These boxplots display the median length of error bursts, the lower and upper quartiles, and whiskers extending to 1.5 times the interquartile range. The boxplot of channels that do not overlap with Wi-Fi channels are plotted as dashed lines. We clearly observe that the error burst length varies across the channels. IEEE 802.15.4 channels 20 and 25 show the smallest burst lengths indicating MFA as dominant source of errors. However, error bursts on IEEE 802.15.4 channels 15 to 19 appear to have the highest upper quartile and whisker, respectively. These channels suffer from error patterns characterized by external interference.

Obviously, a high number of symbol errors that is beyond the summed up error correcting capability of the component codes in the FEC scheme will never be able to be corrected. However, the results shown in Fig. 6 can be utilized to determine the necessary interleaver depth to effectively disrupt most error bursts by more evenly distributing errors across the component codes. When using the interleaver of the IEEE 802.15.4 SUN FSK PHY, the scheme shuffles symbols in a deterministic square constellation. Thus, only square numbers are accepted as interleaver depth. In order to cope with error bursts across all channels an interleaver depth of at least 64 is a reasonable choice.

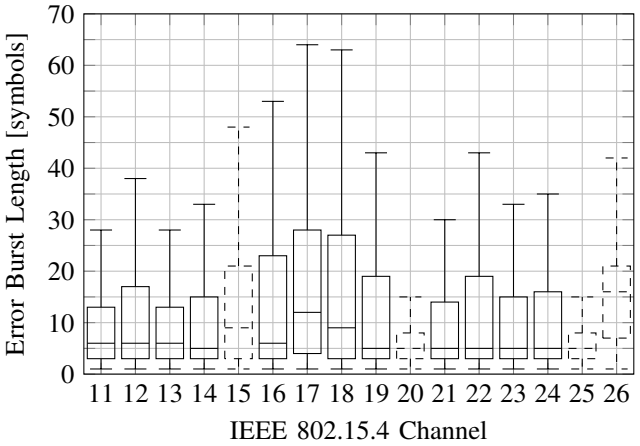


Figure 6: Measured symbol error burst lengths in 125 B (250 symbols) long packets on the 16 IEEE 802.15.4 channels.

VI. Practical Evaluation

In order to evaluate the effectiveness of the presented FEC scheme using a RS(15,9) component code, we analyze the experimental results in terms of power consumption, packet statistics and packet salvation ratio (PSR).

A. Power Consumption

The key question that needs to be clarified when investigating the power consumption of the FEC scheme is whether it exceeds the cost of a retransmission. Therefore, we attach Power Profiler Kits (PPKs) II [19] to the nRF boards running the PTT from the testbed. Besides the PPK's capability of measuring current consumption at a maximum sampling rate of 100 kS/s, it has an in-built logic analyzer. This allows capturing the state of particular GPIO pins, used to mark the start and end of certain operations in the firmware (FW).

The results of these measurements for the transmit process are visualized in Fig. 7 and for the receive process in Fig. 8. The first noticeable difference between the two graphs is the significantly higher current during the receive process, due to the radio being in constant receiving mode. In contrast, the transmitter's radio is only activated as needed. While a TSCH implementation with a fixed schedule could reduce the receiver's current draw, it was not included in this experiment. Indeed, the measured current levels (DC/DC Regulator active) of the two modes matches the values from the datasheet [14].

The first measured current samples in Fig. 7 are in the range of a few hundred μA. During the encoding process, a GPIO is pulled and a current of about 3.3 mA is detected for a duration of 580 μs. The current stays at this level afterwards for some PTT specific operations. The characteristic peaks of approximately 3 mA originate from the CLI of the PTT during logging activities. Later, the radio is activated for the transmit operation for about 5 ms at which the current rises up to an average value of 6.4 mA.

In Fig. 8 the radio is in receive mode. The datasheet indicates a current draw of 6.53 mA in this state, which aligns with the measured samples. During reception, additional CPU activity increases the current draw to around 9 mA for 2.4 ms. The subsequent decoding process, lasting about 1.2 ms, draws 8.6 mA. Depending on the number of bit errors, decoding may extend by about 1 ms, as also noted by Yu et al. [6]. This current level persists for a few more ms due to additional PTT operations. The narrow peaks following the receive and decoding processes are attributed to CLI logging activities.

Although these values are strongly application- and implementation-dependent, we provide an estimate of the over-

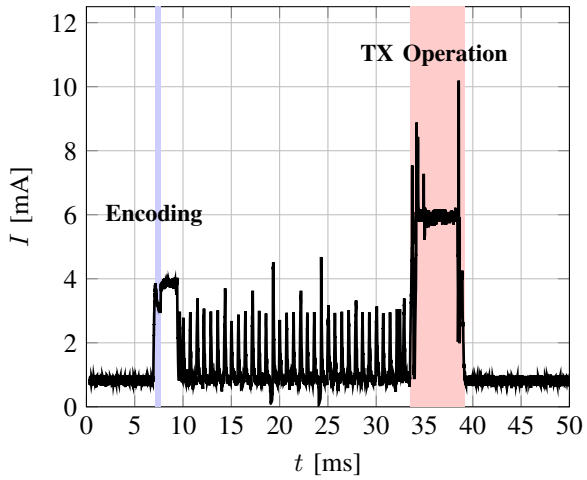


Figure 7: Current draw during transmit process.

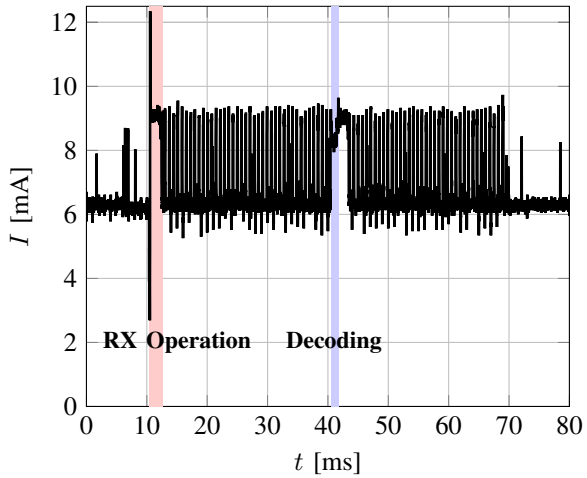


Figure 8: Current draw during receive process.

head introduced by the proposed FEC scheme to clarify the extent of energy savings compared to retransmission. Table II lists the overhead in terms of charge consumption caused by the FEC scheme using the RS(15,9) code. The scheme's effective code rate is $R_{\text{eff.}} \approx 0.58$, it is therefore fair to compare the scheme with an uncoded transmission of packets containing just $\lceil 0.58 \cdot 115 \text{ B} \rceil = 67$ B. Consequently, the overhead caused by the redundancy bits during transmission and reception is calculated by subtracting the charge consumption of a 67 B frame from that of a 115 B encoded frame. Fig. 9 shows the charge drawn during transmission (0 dBm) and reception on the PTT using the nRF52840-DK for different payload sizes. The resulting overhead for transmitting and receiving the redundancy block is $8.93\ \mu\text{C} + 2.92\ \mu\text{C} = 11.85\ \mu\text{C}$. Additionally, the cost of encoding is 1.91 μC. Just in the case of a CRC error, the decoding function is entered consuming a charge of 10.32 μC. This high value is due to the always-on radio during the receive mode of the PTT. We compare this value to the cost of a retransmission. In the retransmission

Operation	Time [ms]	Current [mA]	Charge [μC]	Always	On failure
Tx Encoder	0.58	3.30	1.91	✓	
Tx Send Redundancy	1.40	6.38	8.93	✓	
Rx Receive Redundancy	0.32	9.12	2.92	✓	
Rx Decoder	1.20	8.60	10.32		✓

Table II: Charge consumption overhead caused by FEC scheme using RS(15,9) as component code.

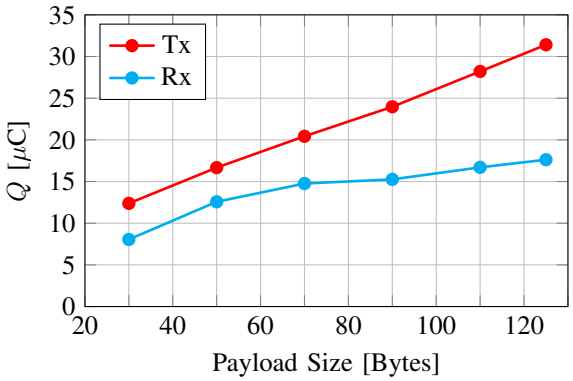


Figure 9: Drawn charge by Tx (0 dBm) and Rx operation in PHY Test Tool on nRF52840DK for various payload sizes.

scenario, we disable FEC and measure the charge drawn during a complete transmission and reception of a 68 B packet, including all PTT-related operations to ensure a fair comparison. The transmitter consumes 41 μC, the receiver 181 μC, for a total of 222 μC. Generally, the FEC overhead for a correct transmission without the need for decoding is 13.76 μC. Using FEC becomes more energy-efficient than retransmission if at least one out of $\left\lfloor \frac{222\ \mu\text{C}}{13.76\ \mu\text{C}} \right\rfloor = 16$ packets contains errors that the scheme can correct. If this condition isn't met, FEC is still beneficial, but switching to a less powerful component code with a better code rate is more efficient. For example, using RS(15,11) is advantageous if at least one out of 39 frames contains correctable errors. While this code offers a better rate, it has a lower error-correcting capability, which may not be sufficient in environments with higher error rates, explaining why starting with a more powerful code may be preferable.

B. Packet Statistics

The packet statistics based on the analysis of the log files of 500,000 transmitted packets are plotted in Fig. 10. The upper bar shows the results when FEC is disabled and the lower bar for the case that FEC tries to correct the packet if the CRC check fails. For this experiment we use a RS(15,9) code across all channels and an interleaver depth of 64. All the transmitted and received packets are categorized in one of 5 groups. The first group of correct packets are these frames which made it to the receiver without any error. The second group contains corrupted packets with failed CRC check and at least one error. The third group are recovered packets, which are those frames that have been erroneous initially, but the decoder is able to resolve all errors. The fourth group is denoted as lost packets and comprises transmitted frames that have never reached the receiver, i.e. the preamble sequence has not been detected.

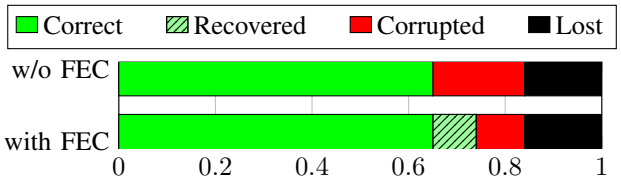


Figure 10: Packet statistics of experiment with and without FEC scheme based on (15,9) RS-Codes on all channels.

Although Fig. 10 does not show the fifth group of interfering packets, it is worth mentioning that the receiver sniffs all frames following the IEEE 802.15.4 structure and therefore also frames from nearby interfering networks.

By measuring an average RSSI of -83 dBm at the receiver, the experiment conditions may be considered as challenging. However, the intention of this setup is to collect a large set of erroneous packets to validate the effectiveness of the FEC scheme. Achieving a PDR of 63% is indeed not an unusual value under these conditions [20]. When enabling FEC, the PDR can be significantly improved up to 74% which comes at the cost of a lower information rate of just 58%.

C. Packet Salvation Ratio

To quantify the performance of the overall FEC scheme, we adopt the use of the metric packet salvation ratio (PSR), defined by Barac et al. [9] as

$$\text{PSR} = \frac{N_{\text{recovered}}}{N_{\text{corrupted}}},$$

which denotes the fraction of recovered packets out of all erroneous frames with CRC error.

In Fig. 11, the PSR for each channel is shown when using each of the different codes of the RS code ensemble of length $n = 15$. An interleaver depth of 64 has been used in all cases. For smaller depths, a major degradation in PSR can be observed as the error bursts more often stretch over a single component code and exceed its error correcting capability. For larger interleaver depths greater than 64, no significant changes are observed and the PSR values seem to saturate.

When assessing the PSR across various channels, several insights emerge. The PSR naturally declines as the effective code rate $R_{\text{eff.}}$ rises. External interference from IEEE 802.11 channel 6 has a significant negative effect on the PSR. Channels 20 and 25 on IEEE 802.15.4, dominated by MFA error patterns, show the highest PSR values, as these patterns are easier for the decoder to correct. The benefits of stronger error-correcting codes do not increase linearly. The gain from using the RS(15,3) code over the RS(15,7) code is minimal across all channels and does not justify the considerable loss in data rate. Nonetheless, achieving a specific target PSR necessitates different codes depending on the channel.

VII. An Approach for Adaptive Channel-Dependent FEC

The findings from Fig. 11 motivate the use of a rate-adaptive FEC scheme switching the component code depending on the currently observed error patterns. In this section, we highlight certain implementation details necessary to approach the design of such a scheme. Initially, there is no knowledge on error patterns available and therefore the same component code shall be used on all channels. Furthermore, a feedback loop needs to be established, which reports in certain intervals the result of decoding operations. Essential information in this context is the PDR, the number of packets with CRC error and the error positions inside packets that have successfully been resolved. These reports reach the central network manager, which implements a logic based on user requirements deciding for a more or less powerful scheme or switching off FEC completely for a certain channel. To roll out a change of the FEC component code to be used by the transmitter and receiver simultaneously, specific command messages may be defined or Application Performance Management (APM) interfaces, such as CORECONF [21] in the 6TiSCH context, can be used. To ensure the receiver uses the same code for decoding as used by the transmitter for encoding, a specific information field in the frame structure shall be reserved for that purpose. It is crucial to protect this information field in order to prevent from becoming a bottleneck, as bit flips in this field would directly result in desperate decoding attempts causing unnecessary overheads in power consumption and latency. Thus, a distinct, short and light-weight FEC code must be used to protect this field. Certainly, this FEC code for the scheme information field is not allowed to change during runtime and must be the same across all nodes in the network. This concept has also been already used in the Digital Video Broadcasting – Cable 2 (DVB-C2) standard [22] employing a 32 bit long rate 0.5 Reed-Muller (RM) code to protect the FEC frame header in a special way.

VIII. Conclusion

In this study we evaluate the practical use of a FEC scheme for the IEEE 802.15.4 O-QPSK PHY. Therefore, the scheme has been implemented as part of a PHY Test Tool inside a real world lab testbed. Based on gathered erroneous packets from this testbed a minimal interleaver depth to mitigate the harmful impact of long error bursts has been derived. However, the PSR varies over time and per channel. Power consumption measurements are evaluated and show that the overhead of sending these redundancy blocks is the crucial factor from a energy perspective. This means that the component code parameters need to be chosen so that the overhead for sending this redundancy does not exceed the cost of a retransmission. In the experiment, the RS(15,9) component code proved effective, correcting nearly every second erroneous packet. However, for the scheme to be worthwhile, there needs to be at least one erroneous packet in every 16 packets. Otherwise, a higher-rate component code is preferable. The results suggest that an adaptive, channel-dependent FEC scheme, which adjusts the code rate based on error patterns across the 16 IEEE 802.15.4 channels, is beneficial. Finally, implementation details for such a dynamic scheme are discussed, with future work focusing on the network manager's switching logic based on user requirements and observed error patterns.

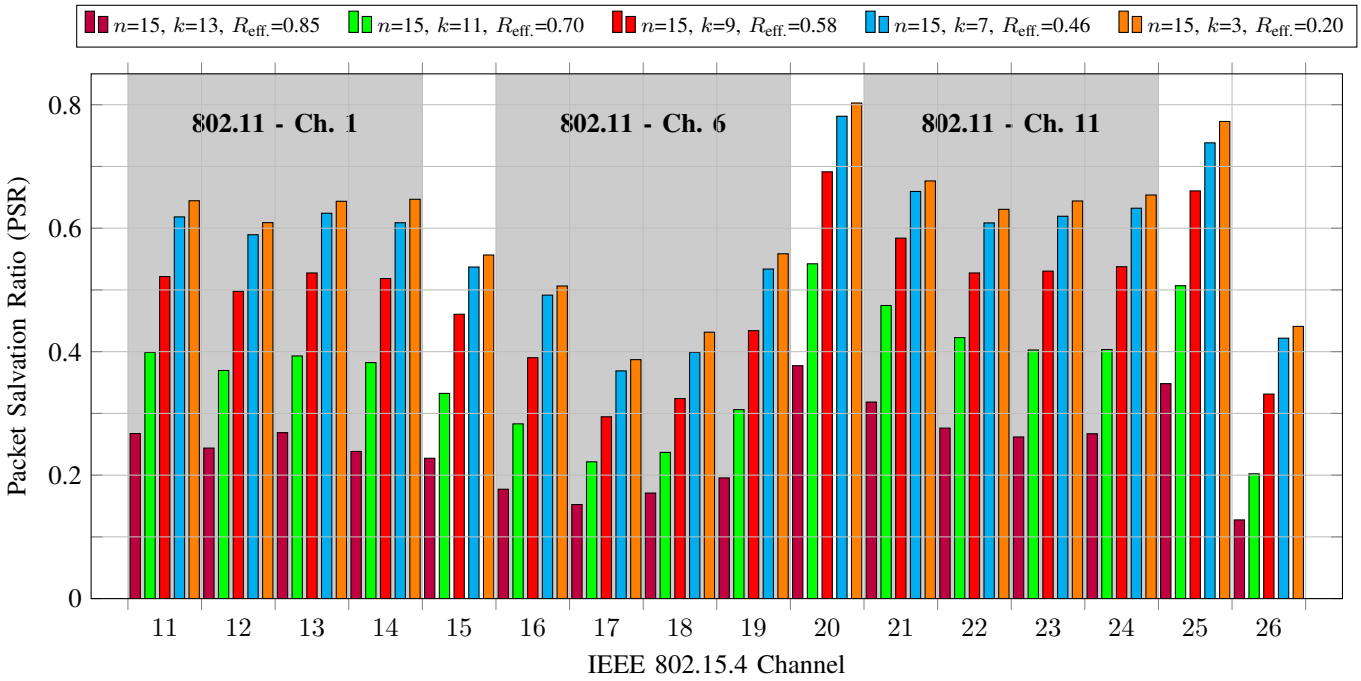


Figure 11: Packet Salvation Ratio across the 16 IEEE 802.15.4 channels for the presented FEC scheme (interleaver depth = 64) using different effective code rates R of the length n = 15 Reed-Solomon codes.

Acknowledgment

This document is issued within the frame and for the purpose of the OpenSwarm project. This project has received funding from the European Union's Horizon Europe Framework Programme under Grant Agreement No. 101093046.

References

[1] IEEE Computer Society, "IEEE Standard for Information technology–Local and metropolitan area networks– Specific requirements– Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks," *IEEE Std 802.15.4-2020 (Revision of IEEE 802.15.4-2015)*, 2020.

[2] Belli, Dimitri and Barsocchi, Paolo and Palumbo, Filippo, "Connectivity Standards Alliance Matter: State of the art and opportunities," *Internet of Things*, pp. 101 005 – 101 232, 2023.

[3] IEEE Computer Society, "IEEE Standard for Information technology–Local and metropolitan area networks– Specific requirements– Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks," *IEEE Std 802.15.4-2006 (Revision of IEEE 802.15.4-2003)*, 2006.

[4] ——, "IEEE Standard 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY)- Amendment 1: MAC Sublayer," *IEEE Std 802.15.4e-2012 (Amendment to IEEE 802.15.4-2011)*, pp. 1–225, 2012.

[5] ——, "IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks," *IEEE Std 802.15.4-2015 (Revision of IEEE 802.15.4-2011)*, 2016.

[6] K. Yu, M. Gidlund, J. Åkerberg, and M. Bjorkman, "Reliable and low latency transmission in industrial wireless sensor networks," *Procedia Computer Science*, vol. 5, pp. 866–873, 2011.

[7] K. Yu, F. Baracć, M. Gidlund, J. Åkerberg, and M. Björkman, "A flexible error correction scheme for IEEE 802.15. 4-based industrial wireless sensor networks," in *IEEE International Symposium on Industrial Electronics*, 2012.

[8] K. Yu, F. Barac, M. Gidlund, and J. Åkerberg, "Adaptive forward error correction for best effort wireless sensor networks," in *2012 IEEE International Conference on Communications (ICC)*, 2012.

[9] F. Barac, M. Gidlund, and T. Zhang, "Scrutinizing bit-and symbol-errors of IEEE 802.15. 4 communication in industrial environments," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 7, pp. 1783–1794, 2014.

[10] ——, "LPED: channel diagnostics in WSN through channel coding and symbol error statistics," in *IEEE Ninth Int. Conference on Intelligent Sensors, Sensor Networks and Inf. Processing (ISSNIP)*, 2014.

[11] F. Barac, S. Caiola, M. Gidlund, E. Sisinni, and T. Zhang, "Channel diagnostics for wireless sensor networks in harsh industrial environments," *IEEE Sensors Journal*, vol. 14, no. 11, pp. 3983–3995, 2014.

[12] F. Barac, M. Gidlund, and T. Zhang, "PREED: Packet recovery by exploiting the determinism in industrial WSN communication," in *International Conf. on Distributed Computing in Sensor Systems*, 2015.

[13] U. Pešović and P. Planinšič, "Error Probability Model for IEEE 802.15.4 Wireless Communications in the Presence of Co-channel Interference," *Physical Communication*, vol. 25, pp. 43–53, 2017.

[14] Nordic Semiconductors. (2019) Nordic nRF 52840 DK. [Online]. Available: https://infocenter.nordicsemi.com/pdf/nRF52840_PS_v1.1.pdf

[15] ——. (2024) IEEE 802.15.4 PHY Test Tool. [Online]. Available: https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/samples/peripheral/802154_phy_test/README.html

[16] R. H. Morelos-Zaragoza, *The Art of Error Correcting Coding*. John Wiley & Sons, 2006.

[17] W. Ryan and S. Lin, *Channel Codes: Classical and Modern*. Cambridge University Press, 2009.

[18] F. Graf, T. Watteyne, F. Maksimovic, and M. Villnow, "Bit- and Symbol-Error Patterns in IEEE 802.15.4 TSCH Mode," in *ISCC 2024 - IEEE Symposium on Computers and Communications*, 2024.

[19] Nordic Semiconductors. (2020) Nordic Power Profiler Kit II. [Online]. Available: https://infocenter.nordicsemi.com/pdf/PPK_2_User_Guide_20201201.pdf

[20] K. Brun-Laguna, P. Gomes, P. Minet, and T. Watteyne, "Moving Beyond Testbeds? Lessons (We) Learned About Connectivity," *IEEE Pervasive Computing*, vol. 17, no. 4, pp. 15–27, 12 2018.

[21] M. Veillette, P. Van der Stok, A. Pelov, A. Bierman, and C. Bormann, "CoAP Management Interface (CORECONF)," IETF, Tech. Rep., 2023.

[22] European Standard (Telecommunications series), "Digital Video Broadcasting (DVB) - Frame structure channel coding and modulation for a second generation digital transmission system for cable systems (DVB-C2)," *ETSI EN 302 769 V1.2.1*, pp. 1–111, 2010.