An energy-efficient error correction scheme for IEEE 802.15.4 wireless sensor networks

Liang Li, Robert G. Maunder, Bashir M. Al-Hashimi and Lajos Hanzo
School of Electronics and Computer Science, University of Southampton, SO17 1BJ, United Kingdom
Email: rm@ecs.soton.ac.uk

***Abstract*—In this paper, we validate a novel augmentation to the PHYsical layer (PHY) of the IEEE 802.15.4 standard for wireless sensor networks. This augmentation implements Forward Error Correction (FEC) encoding within the transmitting sensor nodes, facilitating a significant reduction in their transmission energy. We detail the design, parameterisation and implementation of this FEC encoder and show that is has only an insignificant energy consumption compared to the transmission energy reduction that it affords. Our analysis shows that net energy savings of 24.8 – 31.4% can be achieved by the augmented PHY.**

I. Introduction

Wireless Sensor Networks (WSNs) are currently enjoying diverse application owing to recent improvements in their cost, flexibility and sensor size. However, the sensor nodes of a WSN are typically required to maintain sporadic but reliable data transmissions for extended periods of time. Furthermore, in many applications, the sensor nodes are required to be small, preventing the use of bulky batteries [1]. Therefore, improvements in the energy efficiency of the sensor nodes are also required in order for WSNs to find further diverse applications.

A starred WSN topology is often employed in low energy consumption applications [2], [3], such as Body Area Networks (BANs) [4] and short range environmental monitoring [5]. Here, the sensor nodes transmit to a central node, which coordinates the reactions of a higher-level system to the sensed data. Owing to its integration into the higher-level system, the central node typically has less-limited energy resources. It is therefore beneficial to redistribute energy consumption from the sensor nodes to the central node, whenever possible.

This energy redistribution can be achieved at each of the communication layers [6]. In the application layer, previous studies [7]–[9] have considered the optimal trade-off between the amount of data that is processed in the sensor nodes and the amount of raw data that is transmitted to the central node for processing. Energy redistribution can also be achieved in the network layer [10], [11] by grouping the sensor nodes into clusters. In this case, the energy consumption is redistributed to clusterheads having higher energy resources, which relay the data transmitted by the sensor nodes on to the central node. Furthermore, Media Access Control (MAC) protocols [3], [12], [13] can be devised to reduce the overhead associated with maintaining links between the sensor nodes and the central node.

In [14] however, we proposed an augmentation to the PHYsical layer (PHY) of the IEEE 802.15.4 standard for WSNs [15], in order to redistribute energy consumption as described above. This was achieved by employing a sophisticated Forward Error Correction (FEC) decoder within the central node, which reduced the transmission energy[1] required to achieve reliable communications. Although additional energy is consumed when performing FEC encoding in the transmitting sensor nodes, the net energy consumption of a Chipcon CC2430 [16] sensor node was shown to be reduced by 17.4 – 23.3% in [14]. However, this result was based on the conservative assumption that the FEC operations are performed by software running on the on-board 8051 processor, which has a much higher complexity than required. More significant energy consumption reductions could therefore be expected, if FEC encoding was performed in hardware. This paper therefore discusses the design, parameterisation and implementation of a dedicated FEC hardware module.

The rest of this paper is organised as follows. Section II reviews the augmented IEEE 802.15.4 PHY and the analysis of its energy savings that was detailed in [14]. In Section III, we detail for the first time the novel deterministic interleaver design that was employed to obtain the results of [14]. This interleaver design is suitable for all possible PHY payload lengths without imposing an excessive memory requirement. A novel Genetic Algorithm (GA) is employed for parameterising the interleaver in order to maximise its performance, as detailed for the first time in Section IV. Section V discusses a novel hardware implementation of the FEC encoder, which employs parallel ‘just-in-time’ processing in order to achieve a low processing latency and energy consumption. This energy consumption is analysed in Section VI and our analysis shows that it is insignificant compared to the transmission energy saving that it affords. As a result, the augmented PHY is shown to offer net energy consumption savings of 24.8 – 31.4%, which are significantly greater that those reported in [14] of 17.4 – 23.3%. Finally, we offer our conclusions in Section VII.

[1] Note that it is necessary to consider energy consumption rather than power consumption in this paper. This is because the amount of time required to encode a data payload can be different from the amount of time required to transmit it. For this reason, the relationship between the power consumed during encoding and transmission will be different from the relationship between the amount of energy these processes consume.

II. The Augmented PHY

As detailed in [14], the proposed augmentation to the IEEE 802.15.4 PHY can be employed to convey the payloads of IEEE 802.15.4 data frames using a reduced transmission energy. However, the augmented PHY imposes additional interleaving and rate-1 encoding operations upon the transmitting sensor nodes. As shown in the schematic of Figure 1, these operations are performed between the Pseudo Noise (PN) spreading and Offset Quadrature Phase Shift Keying (O-QPSK) operations of the standard IEEE 802.15.4 PHY [15].

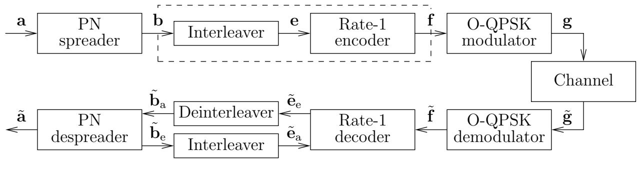

Fig. 1. Schematic of the augmented IEEE 802.15.4 PHY.

The augmented PHY of Figure 1 applies PN spreading to the M-byte PHY payload $\mathbf{a}$, where $M \in [10 \dots 127]$ [15, Section 6.3]. This is achieved by decomposing the payload $\mathbf{a}$ into sets of $k = 4$ consecutive bits and mapping these to $n = 32$-chip PN sequences [15, Table 24], like in the standard PHY. These PN sequences are concatenated to obtain the N-chip sequence $\mathbf{b}$, where $N = 8Mn/k$. The interleaver of the augmented PHY then employs a three-step so-called Dithered Relative Prime (DRP) process [17] to rearrange the order of the chips in $\mathbf{b}$. This process is detailed for the first time in Sections III and IV of this paper. As shown in Figure 1, the resultant N-chip sequence $\mathbf{e}$ is rate-1 encoded [18], as detailed in Section III. Finally, the encoded chip sequence $\mathbf{f}$ is O-QPSK modulated, like in the standard PHY. As detailed in Section III, the input of the augmented PHY's O-QPSK modulator $\mathbf{f}$ comprises the same number N of chips as the output of the PN spreader $\mathbf{b}$, like in the standard PHY. For this reason, the PN spreader and O-QPSK modulator remain completely unchanged, when the augmented PHY is employed in the transmitting sensor nodes.

In the receiving central node, additional rate-1 decoding and deinterleaving operations are employed by the augmented PHY. This employs iterative decoding [19], which repeatedly alternates the operation of the PN despreader and the rate-1 decoder, as shown in Figure 1. This is in contrast to the receiver of the standard PHY, which employs only the 'one-shot' operation of the PN despreader. Since the augmented PHY invests more decoding complexity in the central node than the standard PHY, it can achieve a desirable Payload Error Ratio (PER) using a reduced sensor node transmission energy. This is demonstrated by the simulation results of Figure 2, which considers transmission over a Line-Of-Sight (LOS) channel in the presence of Additive White Gaussian Noise (AWGN) having a constant power spectral density N_0, in common with [15, Figure E.2]. These results show that when transmitting $N = 640$-chip payloads, the augmented PHY can achieve a desirable PER of 10^{-3} at a transmission energy per chip E_c that is 3.22 dB lower than that required by the standard PHY. Furthermore, this gain increases to 6.75 dB, when $N = 8128$-chip payloads are employed, owing to the augmented PHY's interleaver gain [14], which is obtained when transmitting longer payloads.

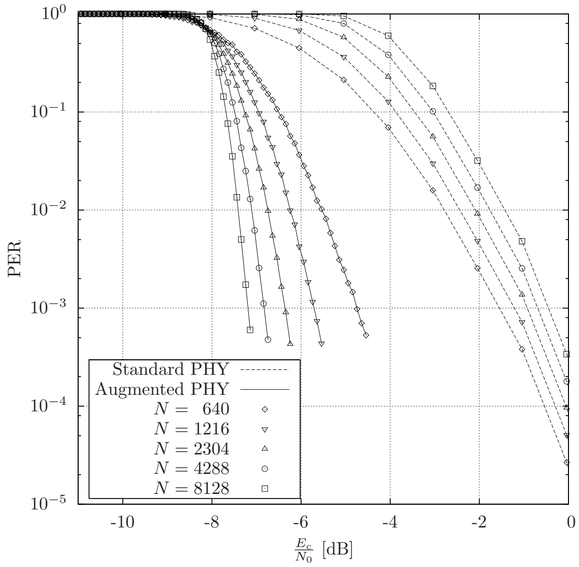

Fig. 2. PER performance of the standard and augmented PHYs for payloads comprising various numbers of chips N, when communicating over LOS AWGN channels, having a range of values for the Signal to Noise Ratio (SNR) per payload chip E_c/N_0.

In order to assess the practical sensor node energy saving, the augmentation of the Chipcon CC2430 PHY [16] was considered in [14]. The energy consumed during transmission is given by $E_\mathrm{tx} = I_\mathrm{tx} \cdot V \cdot t_\mathrm{tx}$, where $t_\mathrm{tx} = N/f_\mathrm{tx}$ is the transmit duration and the IEEE 802.15.4 transmission rate is $f_\mathrm{tx} = 2 \cdot 10^6$ chips per second [15]. As may be expected, the current I_tx consumed during the transmission of a data payload depends on the particular transmit energy per chip E_c employed. In its maximum transmit power mode of 0.6 dBm, the Chipcon CC2430 consumes $I_\mathrm{tx}^\mathrm{std} = 32.4$ mA [16, Table 45]. At this transmit power, the amount of energy $E_\mathrm{tx}^\mathrm{std} = I_\mathrm{tx}^\mathrm{std} \cdot V \cdot t_\mathrm{tx}$ consumed by the standard PHY without augmentation is illustrated in Figure 3 for payloads comprising various numbers N of chips. As described above, the augmented PHY reduces the transmission energy required to achieve a desirable PER of 10^{-3} by 3.22 – 6.75 dB, depending on the length of the payload N. Corresponding reductions from the Chipcon CC2430's maximum transmit power of 0.6 dBm allow its current consumption $I_\mathrm{tx}^\mathrm{aug}$ to be lowered from 32.4 mA to 21.7 – 23.7 mA [16, Table 45]. Figure 3 shows the amount of transmission energy $E_\mathrm{tx}^\mathrm{aug} = I_\mathrm{tx}^\mathrm{aug} \cdot V \cdot t_\mathrm{tx}$ consumed by the augmented PHY for various values of N. These results show that the augmented PHY facilitates *gross* sensor node energy savings of $(E_\mathrm{tx}^\mathrm{std} - E_\mathrm{tx}^\mathrm{aug})$ that are 27.0 – 33.0% of $E_\mathrm{tx}^\mathrm{std}$, depending on the payload length N.

However, in order to determine the *net* sensor node energy saving $[E_\mathrm{tx}^\mathrm{std} - (E_\mathrm{tx}^\mathrm{aug} + E_\mathrm{pr1}^\mathrm{aug})]$ that is afforded by the augmented PHY, it is necessary to additionally consider the energy $E_\mathrm{pr1}^\mathrm{aug}$ consumed during the operation of the interleaver and rate-1 encoder that are boxed in Figure 1. In [14], it was assumed that these operations were performed by software running on the 8051 processor of a Chipcon CC2430 sensor

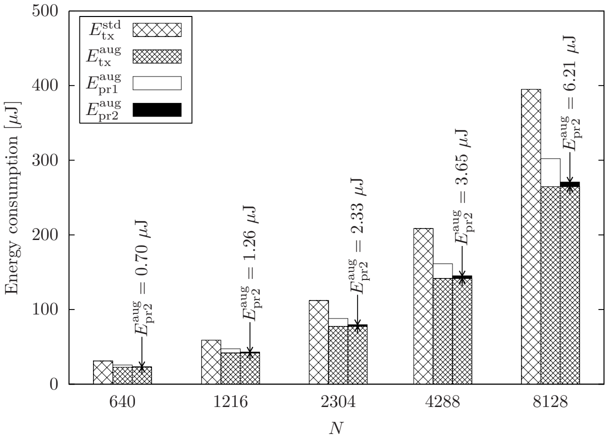

Fig. 3. Total energy consumed in the standard Chipcon CC2430 PHY $E_{\text{tx}}^{\text{std}}$, in the software implementation of the augmented PHY ($E_{\text{tx}}^{\text{aug}} + E_{\text{pr1}}^{\text{aug}}$) and in the proposed hardware implementation of the augmented PHY ($E_{\text{tx}}^{\text{aug}} + E_{\text{pr2}}^{\text{aug}}$).

node. Since the interleaver of Figure 1 employs a three-stage process and rate-1 encoding can be completed in a single step, it was assumed that each chip in the N-chip payload $\mathbf{b}$ can be processed using 4 clock cycles, requiring $C_{\text{pr1}}^{\text{aug}} = 4N$ cycles in total. The duration of this processing is therefore $t_{\text{pr1}}^{\text{aug}} = C_{\text{pr1}}^{\text{aug}} / f_{\text{pr1}}^{\text{aug}}$, where $f_{\text{pr1}}^{\text{aug}}$ is the system's clock frequency. Here, $f_{\text{pr1}}^{\text{aug}}$ is assumed to be 32 MHz, which is the clock frequency on the Chipcon CC2430 [16]. The energy consumed by interleaving and rate-1 encoding is given by $E_{\text{pr1}}^{\text{aug}} = I_{\text{pr1}}^{\text{aug}} \cdot V \cdot t_{\text{pr1}}^{\text{aug}}$. Here, a supply voltage $V = 3$ Volts equal to that of the Chipcon CC2430 and a current consumption of $I_{\text{pr1}}^{\text{aug}} = 12.3$ mA equal to the peak consumption of the on-board 8051 processor [16, Table 4] were conservatively assumed. The resultant processing energy consumptions $E_{\text{pr1}}^{\text{aug}}$ are shown in Figure 3 for a variety of payload lengths N. Using this software implementation of the augmented PHY, net energy savings $[E_{\text{tx}}^{\text{std}} - (E_{\text{tx}}^{\text{aug}} + E_{\text{pr1}}^{\text{aug}})]$ that correspond to 17.4 – 23.3% of $E_{\text{tx}}^{\text{std}}$ are afforded, as reported in [14].

As described in Section I however, a more efficient implementation of the augmented PHY in a transmitting sensor node would resemble the Chipcon CC2430, but with the addition of a hardware module dedicated to performing interleaving and rate-1 encoding. Since a dedicated module would be much simpler than an 8051 processor and because it could benefit from parallel processing, this approach would offer a reduced processing energy consumption $E_{\text{pr2}}^{\text{aug}}$ and therefore an increased net energy saving. In the following sections, we detail the design, parameterisation, implementation and characterisation of a hardware module for this purpose. We shall show that this module has an insignificant energy consumption $E_{\text{pr2}}^{\text{aug}}$ and facilitates net energy savings $[E_{\text{tx}}^{\text{std}} - (E_{\text{tx}}^{\text{aug}} + E_{\text{pr2}}^{\text{aug}})]$ that are 24.8 – 31.4% of $E_{\text{tx}}^{\text{std}}$, as shown in Figure 3.

III. Module design

As described in Section II, the proposed module implements the interleaver and rate-1 encoder that are boxed in Figure 1. Here, the standard IEEE 802.15.4 PN spreader [15, Section 6.5] provides the input chip sequence $\mathbf{b}$. As described in Section II, this comprises $8M/k$ number of $n = 32$-chip PN sequences [15, Table 24], where M is the number of bytes in the data payload $\mathbf{a}$. Since this has 118 possible values $M \in [10 \ldots 127]$, there are 118 possible lengths $N = 8Mn/k \in \{640, 704, 768, \ldots, 8128\}$ for the chip sequence $\mathbf{b}$ [14].

When repositioning the chips in the sequence $\mathbf{b} = \{b_i\}_{i=0}^{N-1}$, the interleaver of Figure 1 is required to desirably 'randomise' the order of the chips in the resultant sequence $\mathbf{e} = \{e_i\}_{i=0}^{N-1}$. In order to achieve this, the interleaver must fully exploit the grade of freedom for re-positioning the chips, which increases with the number of chips N. As a result, different interleaver designs are required for each of the 118 possible values of N and the associated parameters of the interleaver design must be stored in Read Only Memory (ROM).

A naive interleaver design would be parameterised by 118 arrays $\{\pi_{640}, \pi_{704}, \pi_{768}, \ldots, \pi_{8128}\}$, each of which would comprise N unique integers in the range of $[0, N-1]$. The operation of the naive interleaver can be formally specified as $e_i = b_{\pi_N[i]}$, where $\pi_N[i]$ is the i^{th} element in the array π_N and $i \in [0, N-1]$. However, this approach would require approximately 800 KB of ROM, which we consider to be excessive, since memory accesses are typically associated with relatively high energy consumptions in systems on chip [20].

This problem was solved in the implementations detailed in [21]–[23] by employing deterministic interleaver designs. These require the storage of only a limited number of parameters, which are employed to compute the elements of the interleaver pattern in an on-line manner, as and when they are required. However, the designs detailed in [21]–[23] are optimised for turbo codes and are not suitable for the augmented PHY. This is because the interleaver is required to mitigate a higher level of correlation within the soft information exchanged in the augmented PHY, since the PN despreader of Figure 1 operates on relatively long blocks, comprising $n = 32$ chips.

For this reason, we employ a deterministic design resembling a DRP interleaver [17], which has been shown to effectively 'randomise' the order of the chips in the sequence $\mathbf{b}$ without requiring an excessive amount of ROM. Indeed, only 12 KB of ROM are required to store the parameters of our interleaver design, as listed in Table I. Like a DRP interleaver, our design is implemented in three stages, which are referred to as 'Interleaver 1', 'Interleaver 2' and 'Interleaver 3' in our discussions below. As exemplified by the criss-crossing arrows in Figure 4, these interleavers are employed to 'randomise' the order of the intermediate chip sequences $\mathbf{c}$, $\mathbf{d}$ and $\mathbf{e}$, respectively. Note that the intermediate chip sequences comprise the same number of chips as the input sequence $\mathbf{b}$, namely N.

Interleaver 1 Similarly to the first stage of a DRP interleaver, Interleaver 1 of Figure 4 employs a block-based rearrangement of the chips in the input sequence $\mathbf{b} = \{b_i\}_{i=0}^{N-1}$ in order to generate the sequence $\mathbf{c} = \{c_i\}_{i=0}^{N-1}$. More specifically, each block of $n = 32$ chips in $\mathbf{c}$ is provided by rearranging the order of the corresponding $n = 32$-chip PN sequence in $\mathbf{b}$. As shown in Figure 4, a different rearrangement is employed for each $n = 32$-chip PN sequence, as specified

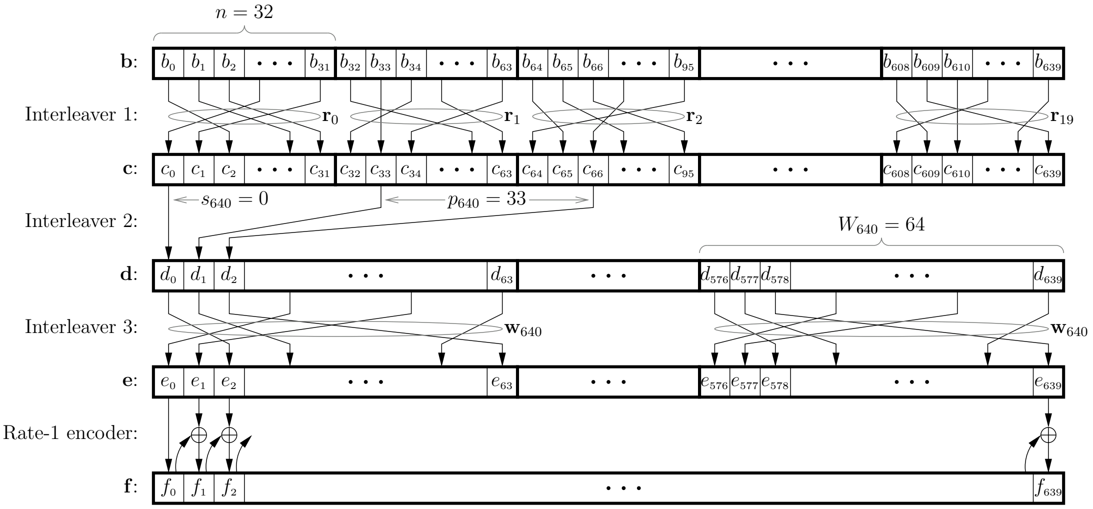
$n = 32$
b:
b_0 b_1 b_2 • • • b_{31} b_{32} b_{33} b_{34} • • • b_{63} b_{64} b_{65} b_{66} • • • b_{95} • • • b_{608} b_{609} b_{610} • • • b_{639}
Interleaver 1:
$\mathbf{r}_0$
$\mathbf{r}_1$
$\mathbf{r}_2$
$\mathbf{r}_{19}$
c:
c_0 c_1 c_2 • • • c_{31} c_{32} c_{33} c_{34} • • • c_{63} c_{64} c_{65} c_{66} • • • c_{95} • • • c_{608} c_{609} c_{610} • • • c_{639}
$s_{640} = 0$
$p_{640} = 33$
Interleaver 2:
$W_{640} = 64$
d:
d_0 d_1 d_2 • • • d_{63} • • • d_{576} d_{577} d_{578} • • • d_{639}
Interleaver 3:
$\mathbf{w}_{640}$
$\mathbf{w}_{640}$
e:
e_0 e_1 e_2 • • • e_{63} • • • e_{576} e_{577} e_{578} • • • e_{639}
Rate-1 encoder:
f:
f_0 f_1 f_2 • • • f_{639}

Fig. 4. Example operation of the interleaver and rate-1 encoder of Figure 1 for $N = 640$.

TABLE I
PARAMETERS OF THE INTERLEAVERS SHOWN IN FIGURE 4.

$\{\mathbf{r}_0, \mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_{253}\}$	Each of the 254 arrays $\mathbf{r}_u$ comprises $n = 32$ unique integers from the range $[0, n-1]$.
$\{s_{640}, s_{704}, s_{768}, \ldots, s_{8128}\}$	Each of the 118 integers s_N has a value from the range $[0, n-1]$.
$\{p_{640}, p_{704}, p_{768}, \ldots, p_{8128}\}$	Each of the 118 integers p_N has a value from the range $[1, N-1]$.
$\{W_{640}, W_{704}, W_{768}, \ldots, W_{8128}\}$	Each of the 118 integers W_N has a value from the range $[1, N]$.
$\{\mathbf{w}_{640}, \mathbf{w}_{704}, \mathbf{w}_{768}, \ldots, \mathbf{w}_{8128}\}$	Each of the 118 arrays $\mathbf{w}_N$ comprises W_N unique integers from the range $[0, W_N - 1]$.

by the parameters $\{\mathbf{r}_0, \mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_{253}\}$ of Table I. Note that 254 rearrangements are required, because the sequence $\mathbf{b}$ comprises N/n = 254 PN sequences, when it has a maximal length of $N = 8128$ chips. The operation of Interleaver 1 can be formally specified as $c_i = b_{j_i}$, where

$$j_i = n \cdot u + \mathbf{r}_u[v], \tag{1}$$

$\mathbf{r}_u[v]$ is the v^{th} element in the array $\mathbf{r}_u$, $v = i \bmod n$, $u = i \operatorname{div} n$ and $i \in [0, N-1]$. Here, the 'div' operator indicates integer division, while 'mod' is employed to represent the modulo operator.

Interleaver 2 Similarly, the operation of Interleaver 2 from Figure 4 can be specified as $d_i = c_{j_i}$, where

$$j_i = (s_N + p_N \cdot i) \bmod N \tag{2}$$

and $i \in [0, N-1]$. As in the second stage of a DRP interleaver, s_N identifies the index of the chip in $\mathbf{c}$ that provides the first chip in $\mathbf{d} = \{d_i\}_{i=0}^{N-1}$, as shown in Figure 4. The subsequent chips in $\mathbf{d}$ are provided by employing successive hops of p_N chips (modulo N) to select the corresponding chip in the sequence $\mathbf{c}$. Here, p_N is required to be a relative prime of N in order to ensure that each chip in $\mathbf{c}$ provides exactly one chip for $\mathbf{d}$. Note that the particular values that are employed for s_N and p_N depend upon the length N of the chip sequence, as shown in Table I.

Interleaver 3 Similarly, the parameters employed for Interleaver 3 of Figure 4 depend upon the length N of the chip sequence. This interleaver employs a block-based rearrangement of the chips in $\mathbf{d}$ in order to obtain the sequence $\mathbf{e}$, similarly to Interleaver 1. However in contrast to Interleaver 1, Interleaver 3 employs the same rearrangement for each block of W_N chips in the sequence $\mathbf{d}$, as shown in Figure 4. As seen in Table I, this rearrangement and the block length are described by the parameters $\mathbf{w}_N$ and W_N, respectively. Clearly, W_N is required to be a factor of the chip sequence length N. For example, $W_{640} = 16$, $W_{1216} = 32$, $W_{2304} = 64$, $W_{4288} = 64$ and $W_{8128} = 64$. The operation of Interleaver 3 can be formally specified as $e_i = d_{j_i}$, where

$$j_i = W_N \cdot u + \mathbf{w}_N[v], \tag{3}$$

$\mathbf{w}_N[v]$ is the v^{th} element in the array $\mathbf{w}_N$, $v = i \bmod W_N$, $u = i \operatorname{div} W_N$ and $i \in [0, N-1]$.

Rate-1 encoder Finally, for each chip in the sequence $\mathbf{e} = \{e_i\}_{i=0}^{N-1}$, the rate-1 encoder of Figure 1 generates one chip for its output sequence $\mathbf{f} = \{f_i\}_{i=0}^{N-1}$. More specifically, $f_0 = e_0$ and $f_i = e_i \oplus f_{i-1}$ for $i \in [1, N-1]$, as shown in Figure 4, in which $\oplus$ indicates the modulo-2 addition of two binary chips. As a result, the output chip sequence $\mathbf{f}$ input to the standard IEEE 802.15.4 O-QPSK modulator [15, Section 6.5.2.4] also comprises N chips.

IV. MODULE PARAMETERISATION

Let us now describe the off-line algorithm employed to design values for the interleaver parameters of Table I, in order to ensure that the order of the chips in the sequence $\mathbf{b}$ is effectively 'randomised'. Note that the N-chip input sequence $\mathbf{b}$ has $2^{N/8}$ legitimate permutations, since the PN spreader of Figure 1 has a coding rate of $k/n = 1/8$ [15, Table 24]. As described above, the module detailed in this section maps each of these permutations to a different permutation of the output chip sequence $\mathbf{f}$. The particular mapping that is employed depends upon the parameters of the interleavers. Our off-line algorithm used for designing these parameters attempts to maximise the minimum Hamming distance $d_{\text{H}}^{\min}$ between the legitimate permutations of $\mathbf{f}$, as we shall detail below. In this way, the number of chip errors that is required to transform the transmitted permutation of $\mathbf{f}$ into any other legitimate permutation is maximised. This maximises the probability that transmission errors can be detected and corrected by the iterative decoder of Figure 1, optimising its performance.

Though it is beyond the scope of this paper, it can be shown that $d_{\mathrm{H}}^{\mathrm{min}}$ can be as low as six if the interleaver does not effectively 'randomise' the order of its chips. However, it can be shown that $d_{\mathrm{H}}^{\mathrm{min}}$ will increase to at least 24, provided that the interleaver parameterisation satisfies two conditions. The first condition requires the interleaver to separate every pair of chips from each $n = 32$-chip PN sequence in $\mathbf{b}$ with at least two chips from other PN sequences, when they are re-positioned in $\mathbf{e}$. For example, this condition will *not* be satisfied, if the chips b_{66} and b_{98} (which are constituent of the same $n = 32$-chip PN sequence in $\mathbf{b}$, as shown in Figure 4) are interleaved to positions e_{343} and e_{341}, respectively (which are separated by only one other chip position, namely e_{342}). The second condition of achieving $d_{\mathrm{H}}^{\mathrm{min}} \geq 24$ requires each $n = 32$-chip PN sequence in $\mathbf{b}$ to have no more than one chip in $\mathbf{e}$ that is adjacent to a chip within each of the other PN sequences. For example, if the chips b_{32} and b_{34} are interleaved to positions e_{512} and e_{125}, respectively, and the chips b_{610} and b_{639} are interleaved to e_{124} and e_{513}, respectively, then the second condition will not be satisfied.

The described conditions of achieving $d_{\mathrm{H}}^{\mathrm{min}} \geq 24$ motivated the design of a GA (GA) [24], which was employed for selecting beneficial values for the parameters of Table I. The first goal of the proposed GA was to maximise the minimum positional separation between any two chips in $\mathbf{e}$ that originate from the same $n = 32$-chip PN sequence in $\mathbf{b}$. Our GA's second goal was to minimise the maximum number of occurrences that any two $n = 32$-chip PN sequences in $\mathbf{b}$ have chips that are positioned next to each other in $\mathbf{e}$. Clearly, in order to achieve these goals, the grade of freedom for the interleavers to re-position the N chips in the sequence $\mathbf{b}$ must be fully exploited, as described above.

V. Module implementation

In this section, we describe a hardware module that implements the interleaver and rate-1 encoder that are boxed in Figure 1. The schematic of Figure 5 is employed for the proposed module, which could be integrated between the PN spreader and the O-QPSK modulator of a standard IEEE 802.15.4 implementation, such as the Chipcon CC2430. In the following discussions, we shall detail the proposed module's Input and Output (I/O) interface, datapath, ROM and controller.

The proposed module is specifically designed to avoid imposing any changes upon the I/O interfaces of the standard PN spreader and O-QPSK modulator. These exchange $n = 32$-chip PN sequences [15, Section 6.5.2.4], at a rate of $f_{\mathrm{tx}}/n = 62.5 \cdot 10^3$ per second, where $f_{\mathrm{tx}} = 2 \cdot 10^6$ chips per second, as described in Section II. These features motivate our module's employment of a $f_{\mathrm{pr2}}^{\mathrm{aug}} = 62.5$ kHz clock, which is supplied using the 'Clk' port shown in Figure 5, as well as the 32-chip I/O ports 'Data_in' and 'Data_out'.

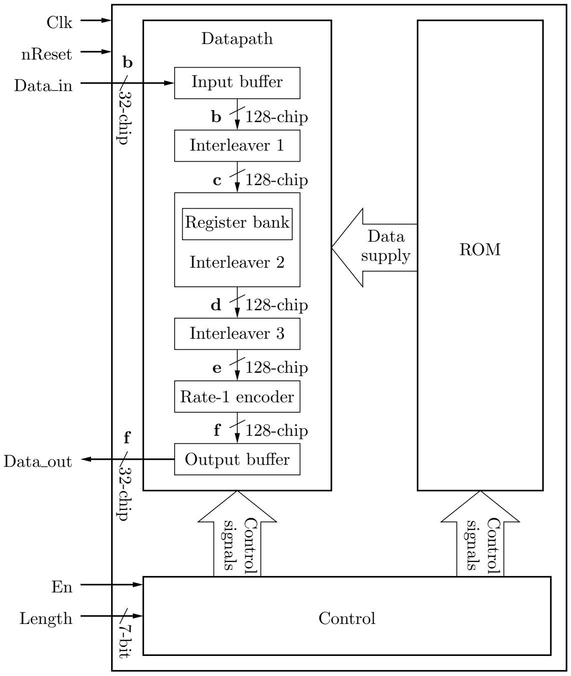

Fig. 5. Schematic of the proposed hardware implementation.

Note that the proposed module has three other ports, as shown in Figure 5. The module begins operating when a logic one is placed upon the 'En' port, allowing its input port to be synchronised with the PN spreader's output port. As described in Section III, the module operates in one of 118 different modes, depending on the length N of the chip sequence $\mathbf{b}$. The value of N can be extracted from the 7-bit 'frame-length field' employed in the IEEE 8092.15.4 PHY header [15, Section 6.5.3], which conveys the number of bytes in the PHY payload $\mathbf{a}$, namely M. This 'frame-length field' is provided to the module using its 7-bit 'Length' port. Finally, the 'nReset' port may be used to reset the registers employed within the proposed module.

A serial structure is employed within the datapath block of the proposed module, as shown in Figure 5. Here, the interleaver of Figure 1 is implemented in three stages, namely Interleaver 1, Interleaver 2 and Interleaver 3, as described in Section III. These stages interleave multiple chips in parallel [25], in order to process the chips at the same rate that they are supplied by the PN spreader, as shown in the timing diagram of Figure 6 and detailed below. This approach facilitates a low processing latency and 'just-in-time' processing, which reduces the number of registers that are required to store intermediate results.

A uniform 128-chip dataflow width is chosen within the datapath block of Figure 5. This uniform width allows Interleaver 2, Interleaver 3 and the rate-1 encoder to be operated within a single clock cycle, without the need for intermediate registers. However, owing to its 8128-chip register bank, Interleaver 2 cannot be tightly connected to Interleaver 1. More specifically, this register bank must collect all of the chips from the sequence $\mathbf{c}$, before Interleaver 2 can commence generating the sequence $\mathbf{d}$, owing to the relative prime number based

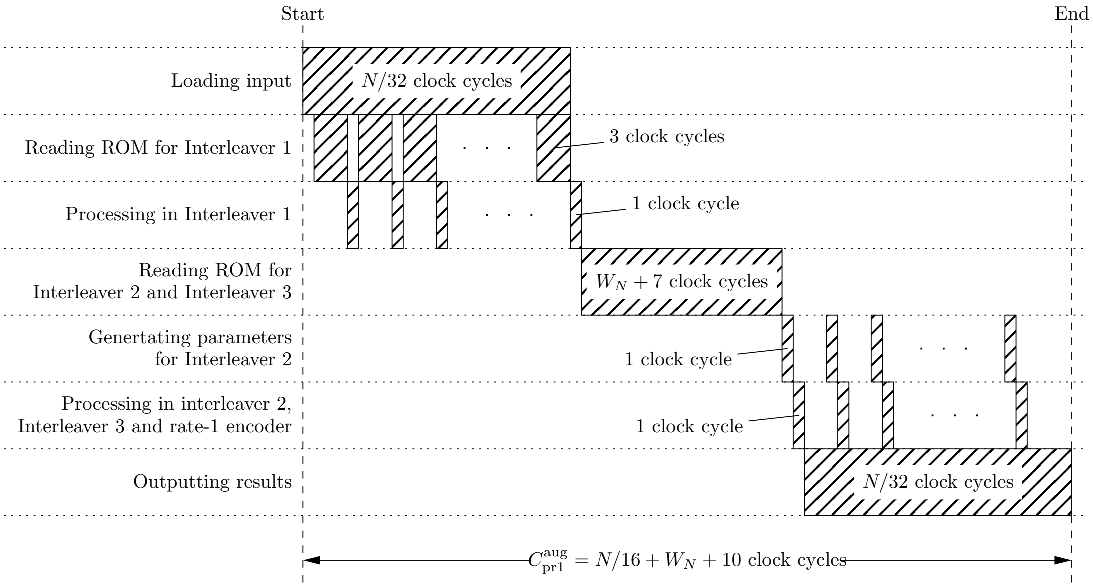
Start
End
Loading input
$N/32$ clock cycles
Reading ROM for Interleaver 1
3 clock cycles
Processing in Interleaver 1
1 clock cycle
Reading ROM for Interleaver 2 and Interleaver 3
$W_N + 7$ clock cycles
Genertating parameters for Interleaver 2
1 clock cycle
Processing in interleaver 2, Interleaver 3 and rate-1 encoder
1 clock cycle
Outputting results
$N/32$ clock cycles
$C_{\mathrm{pr1}}^{\mathrm{aug}} = N/16 + W_N + 10$ clock cycles

Fig. 6. Timing diagram of the proposed hardware implementation.

hops that are required, as discussed in Section III. The 128-chip input and output buffers shown in Figure 5 are required owing to the different port widths employed inside and outside of the datapath block.

As described in Section III, the three interleaving stages are parameterised as described in Table I. In the proposed module, these parameters are stored in the ROM block shown in Figure 5. Combinational logic is employed to convert these parameters into the chip indices j_i, according to (1), (2) and (3) for Interleaver 1, Interleaver 2 and Interleaver 3, respectively. Here, the complexity of calculating (2) can be reduced, if it is implemented recursively according to

$$j_i = (j_{i-1} + p_N) \bmod N, \tag{4}$$

where $i \in [1, N-1]$ and $j(0) = s_N$. As shown in Figure 5, Interleaver 2 is required to interleave 128 chips in parallel. Note that 128 adders could be chained together to perform the required calculations of (4). However, the necessary combinational logic path would be too long to be performed within a single clock cycle. For this reason, we employed eight parallel adder chains, each employing 16 adders to determine a different subset of the 128 chip indices j_i in a single clock cycle, striking a trade-off between chip area and speed.

Let us now describe the working flow of the proposed module, which is directed by the control block shown in Figure 5. As described above, 128-chip subsets of the sequence $\mathbf{b}$ are processed by Interleaver 1 as and when they are supplied to the input buffer, at a rate of 32 chips per clock cycle. However, 128 elements from the arrays $\mathbf{r}_u$ of Table I are required in order to calculate (1) for each 128-chip subset of $\mathbf{b}$. For this reason, a multiport ROM is employed to provide these 128 parameters in just three clock cycles, as shown in the timing diagram of Figure 6. In a fourth clock cycle, the 128 chips that have been loaded into the input buffer during the previous four clock cycles are interleaved and stored in the register bank of Interleaver 2. Note that during the one of four clock cycles in which Interleaver 1 is operated, it is necessary to both read from and write to the input buffer of Figure 5. When this process completes, the register bank of Interleaver 2 will store the chip sequence $\mathbf{c}$, as shown in Figure 4.

Next, the registers that were used to store the parameters of Interleaver 1 are reused for storing the parameters of Interleaver 2 and Interleaver 3. Since the register bank of Interleaver 2 can store the chip sequence $\mathbf{c}$ indefinitely, there is no need to use multiport ROM accesses, when loading the parameters of Interleaver 2 and Interleaver 3. Hence, as shown in Figure 6, $W_N + 7$ clock cycles are used to load the parameters s_N, p_N and W_N, as well as the W_N elements of the array $\mathbf{w}_N$ of Table I.

Following this, the chips of the sequence $\mathbf{c}$ are processed by Interleaver 2, Interleaver 3 and the rate-1 encoder in order to obtain the sequence $\mathbf{f}$, as shown in Figure 4. Here, 128 chips are processed and written into the output buffer of Figure 5 at a time. As shown in Figure 6, 'just-in-time' processing is employed to populate the output buffer at the same rate that it supplies chips to the O-QPSK modulator of Figure 1, namely at 32 chips per clock cycle. Hence, one clock cycle is employed to recursively perform the calculations of (4), one clock cycle is employed to process the 128 chips and two idle clock cycles are employed.

As shown in Figure 6, the proposed module employs a total of $C_{\mathrm{pr2}}^{\mathrm{aug}} = N/16 + W_N + 10$ clock cycles to perform interleaving and rate-1 encoding.

VI. Energy consumption analysis

In this section we shall consider the energy consumption of the module detailed in Section V. We shall estimate the effect that integrating this module into the Chipcon CC2430 hardware [16] would have upon its total energy consumption. This estimation will be compared with that of [14], which considered the implementation of the interleaver and rate-

1 encoder of Figure 1 in software running on the Chipcon CC2430's 8051 processor, as described in Section I.

The Synopsys Design Complier was employed to synthesise a gate-level implementation of the module detailed in Section V. Our synthesis additionally employed a STMicroelectronics 0.12 μm technology standard cell library, resulting in a 1.6 mm^2 chip area, including the ROM. Synopsys PrimeTime was employed to determine the resultant implementation's average current consumption, which was found to be $I_{\mathrm{pr2}}^{\mathrm{aug}} = 222.3$ μA. We assume that our proposed module consumes no current, when it is deactivated by placing a logic zero upon its En port, as shown in Figure 5. Hence, the duration $t_{\mathrm{pr2}}^{\mathrm{aug}}$ for which the proposed module consumes current is given by $t_{\mathrm{pr2}}^{\mathrm{aug}} = C_{\mathrm{pr2}}^{\mathrm{aug}}/f_{\mathrm{pr2}}^{\mathrm{aug}}$, where $C_{\mathrm{pr2}}^{\mathrm{aug}} = N/16 + W_N + 10$ and $f_{\mathrm{pr2}}^{\mathrm{aug}} = 62.5$ kHz, as described in Section V. Finally, the proposed module's energy consumption is given by $E_{\mathrm{pr2}}^{\mathrm{aug}} = I_{\mathrm{pr2}}^{\mathrm{aug}} \cdot V \cdot t_{\mathrm{pr2}}^{\mathrm{aug}}$, where a supply voltage of $V = 3$ Volts is assumed, like in Section II.

Figure 3 provides the energy consumed by the proposed module $E_{\mathrm{pr2}}^{\mathrm{aug}}$ for payloads comprising various numbers N of chips. These energy consumptions are 76.7 – 83.4% lower than the $E_{\mathrm{pr1}}^{\mathrm{aug}}$ values estimated in [14] for the case where interleaving and rate-1 encoding are performed in software running on the 8051 processor of a Chipcon CC2430. As a result, the energy savings $[E_{\mathrm{tx}}^{\mathrm{std}} - (E_{\mathrm{tx}}^{\mathrm{aug}} + E_{\mathrm{pr2}}^{\mathrm{aug}})]$ afforded by employing the augmented PHY are increased from 17.4 – 23.3% to 24.8 – 31.4% of $E_{\mathrm{tx}}^{\mathrm{std}}$, as shown in Figure 3. Indeed, the energy $E_{\mathrm{pr2}}^{\mathrm{aug}}$ consumed by the proposed module only erodes 4.8 – 8.3% of the transmission energy reduction $(E_{\mathrm{tx}}^{\mathrm{std}} - E_{\mathrm{tx}}^{\mathrm{aug}})$ that it facilitates, which we consider to be an attractive engineering trade-off.

VII. Conclusion

In this paper, we have considered an augmentation of the IEEE 802.15.4 PHY of [14], which significantly reduces the transmission energy required to achieve a desirable PER, at the cost of requiring some additional processing within the sensor nodes. We have proposed a dedicated hardware module for performing this processing and detailed its design, parameterisation, implementation and energy consumption analysis. In particular, our novel design facilitates desirable operation for all possible payload lengths without imposing an excessive memory requirement. Furthermore, a novel GA was employed to parameterise the proposed design, facilitating desirable FEC performance. A novel implementation was proposed, which employs parallel and 'just-in-time' processing to achieve a low processing latency and energy consumption. Finally, our analysis of this energy consumption revealed that it is modest compared with the transmission energy saving that it facilitates. For this reason, we can conclude that sophisticated FEC techniques are desirable for future WSN PHY standards.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, August 2002.

[2] G. Lu, B. Krishnamachari, and C. S. Raghavendra, "Performance evaluation of the IEEE 802.15.4 MAC for low-rate low-power wireless networks," in *Proceedings of the IEEE International Conference on Performance, Computing and Communications*, Phoenix, AZ, USA, April 2004, pp. 701–706.

[3] O. Omeni, A. Wong, A. J. Burdett, and C. Toumazou, "Energy Efficient Medium Access Protocol for Wireless Medical Body Area Sensor Networks," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 2, no. 4, pp. 251–259, December 2008.

[4] S. Drude, "Requirements and application scenarios for Body Area Networks," in *Proceedings of the IST Mobile and Wireless Communications Summit*, Budapest, Hungary, July 2007, pp. 1–5.

[5] T. Ahonen, R. Virrankoski, and M. Elmusrati, "Greenhouse Monitoring with Wireless Sensor Network," in *Proceedings of the IEEE/ASME International Conference on Mechtronic and Embedded Systems and Applications*, Beijing, China, October 2008, pp. 402–408.

[6] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, "Energy-aware wireless microsensor networks," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 40–50, March 2002.

[7] R. Min, M. Bhardwaj, S.-H. Cho, E. Shih, A. Sinha, A. Wang, and A. Chandrakasan, "Low-power wireless sensor networks," in *Proceedings of the International Conference on VLSI Design*, Bangalore, India, January 2001, pp. 205–210.

[8] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," in *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, Seattle, WA, USA, August 1999, pp. 174–185.

[9] J. Kulik, W. R. Heinzelman, and H. Balakrishnan, "Negotiation-based protocols for disseminating information in wireless sensor networks," *Wireless Networks*, vol. 9, no. 2–3, pp. 169–185, March 2002.

[10] S. Bandyopadhyay and E. J. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks," in *Proceedings of the Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, San Francisco, CA, USA, March 2003, pp. 1713–1723.

[11] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie, "Protocols for self-organization of a wireless sensor network," *IEEE Personal Communications*, vol. 7, no. 5, pp. 16–27, October 2000.

[12] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, New York, NY, USA, June 2002, pp. 1567–1576.

[13] T. van Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the International Conference on Embedded Networked Sensor Systems*, Los Angeles, CA, USA, November 2003, pp. 171–180.

[14] R. G. Maunder, A. S. Weddell, G. V. Merrett, B. M. Al-Hashimi, and L. Hanzo, "Iterative decoding for redistributing energy consumption in wireless sensor networks," in *IEEE International Conference on Computer Communications and Networks*, St. Thomas, U.S. Virgin Islands, August 2008, pp. 1–6.

[15] *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*, IEEE Std. 802.15.4, September 2006.

[16] *A True System-on-Chip Solution for 2.4 GHz IEEE 802.15.4 / ZigBee(TM) Datasheet*, Chipcon, June 2007.

[17] S. Crozier and P. Guinand, "High-performance low-memory interleaver banks for turbo-codes," in *Proceedings of the IEEE Vehicular Technology Conference*, vol. 4, Atlantic City, NJ, USA, October 2001, pp. 2394–2398.

[18] D. Divsalar, S. Dolinar, and F. Pollara, "Serial concatenated trellis coded modulation with rate-1 inner code," in *Proceedings of the IEEE Global Telecommunications Conference*, vol. 2, San Francisco, CA, USA, November 2000, pp. 777–782.

[19] S. Benedetto and G. Montorsi, "Iterative decoding of serially concatenated convolutional codes," *Electronics Letters*, vol. 32, no. 13, pp. 1186–1188, June 1996.

[20] T. Okuma, M. Muroyama, and Y. H, "Reducing access energy of on-chip data memory considering active data bitwidth," in *Proceedings of the International Symposium on Low Power Electronics and Design*, Monterey, CA, USA, August 2002, pp. 88–91.

[21] Z. Wang and Q. Li, "Very low-complexity hardware interleaver for turbo decoding," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 54, no. 7, pp. 636–640, July 2007.

[22] J.-H. Kim and I.-C. Park, "Double-binary circular turbo decoding based on border metric encoding," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 55, no. 1, pp. 79–83, January 2008.

[23] M. Martina, M. Nicola, and G. Masera, “A flexible UMTS-WiMax turbo decoder architecture,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 55, no. 4, pp. 369–373, April 2008.

[24] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.

[25] R. Dobkin, M. Peleg, and R. Ginosar, “Parallel interleaver design and VLSI architecture for low-latency MAP turbo decoders,” *IEEE Transactions on Very Large Scale Integration Systems*, vol. 13, no. 4, pp. 427–438, April 2005.