

A Packet-Interleaving Scheme Using RS Code for Burst Errors in Wireless Sensor Networks

Tsang-Ling Sheu, Yen-Hsi Kuo, Zi-Tsan Chou*

Department of Electrical Engineering, National Sun Yat-Sen University, Kaohsiung Taiwan
Email: sheu@ee.nsysu.edu.tw, ysk@atm.ee.nsysu.edu.tw, *ztchou@mail.ee.nsysu.edu.tw

Received 27 May 2015; accepted 10 July 2015; published 13 July 2015

Copyright © 2015 by authors and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY).
<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

In this paper, we propose a packet-interleaving scheme (PIS) for increasing packet reliability under burst errors in wireless sensor networks (WSN). In a WSN, packet errors could occur due to weak signal strength or interference. These erroneous packets have to be retransmitted, which will increase network load substantially. The proposed PIS, encoding data using Reed-Solomon (RS) codes, can classify data into two different types: high-reliability-required (HRR) data and non-HRR data. An HRR packet is encoded with a short RS symbol, while a non-HRR packet with a long RS symbol. When an HRR and a non-HRR packet arrive at a sensor, they are interleaved on a symbol-by-symbol basis. Thus, the effect of burst errors (BE) is dispersed and consequently the uncorrectable HRR packets can be reduced. For the purpose of evaluation, two models, the uniform bit-error model (UBEM) and the on-off bit-error model (OBEM), are built to analyze the packet uncorrectable probability. In the evaluation, we first change the lengths of BE, then we vary the shift positions in a BE period, and finally we increase the number of correctable symbols to observe the superiority of the proposed PIS in reducing packet uncorrectable probability.

Keywords

WSN, RS Code, Burst Errors, Interleaving, Packet Uncorrectable Probability

1. Introduction

Along with the increasing requirements for living quality and home security, sensors have been widely deployed inside or outside a building to collect environmental information, such as temperature, humidity, image, motion

*Corresponding author.

picture, etc. To effectively deliver the collected data back to a control center for further analysis, a wireless sensor network (WSN) [1]-[3] is usually built. However, packet transmission over a WSN may encounter intermittent errors due to weak signals or interferences. The erroneous packets, if comprising of text or numbers, such as temperature or humidity, would require packet re-transmission, which surprisingly increases network load. Thus, the motivation of this paper is to increase transmission reliability over a WSN, which has recently attracted many researchers' attention.

Basically, previous researches on transmission reliability over a WSN can be divided into three major categories: reliable routing, information coding, and network coding. In the first category, to increase the transmission reliability after data are collected by a sensor node, relay nodes (RNs) are employed. For examples, H. Chebbo, *et al.* [4] modified IEEE 802.15.4 MAC frames. The authors added one bit in the frame control field, with which whether it is necessary to build a tree by RNs or just build a simple star, can be determined. Moreover, R. Sampangi, *et al.* [5] utilized RNs to divide sensors into several cluster networks. Since the distance from a sensor to its cluster head is reduced, the quality of data transmission is greatly improved. To protect the routing path, S. Kim, *et al.* [6] utilized both coding and retransmission schemes once the established path fails. However, in these schemes, it is inevitable that end-to-end packet delay will increase accordingly due to multiple-hop forwarding.

Thus, in the second category, instead of developing reliable routing, the authors switch their interests to information coding. For examples, E. Byrne, *et al.* [7] designed a coding scheme which can increase the probability of successful decoding based on graph theory. Y. Hamada, *et al.* [8] proposed a scheme to reduce packet error rate by using Luby Transform (LT) codes [9]. Their proposed schemes have achieved small complexity of $O(n)$; yet too many packets are required for retransmission when bit error rate is high. Thus, K. Ishibashi, *et al.* [10] proposed embedded forward error control (FEC) technique which utilizes RS (Reed Solomon) code to reduce packet error rate. Similarly, M. Busse, *et al.* [11] can recover lost chunk by using Fountain code and Raptor code. To increase data reliability and processing speed, K. Yu, *et al.* [12] designed new FEC which protects header and payload, respectively. Similarly, M. Srouji, *et al.* [13] proposed a reliable data transfer scheme which can adjust the lengths of redundancy code based on the successful receiving rate at the downstream node.

In the third category, to increase transmission reliability, researchers focus on network coding. For examples, to increase the probability of successfully recovering the transmitted data, G. Arrobo, *et al.* [14] proposed cooperative network coding. Similarly, A. Taparugssanagorn, *et al.* [15] proposed unequally error protection (UEP) scheme. In UEP, data are classified into two types: medical and non-medical. Only medical data are EX-ORed, while non-medical data are not. However, in UEP, it is not guaranteed that a downstream sensor node can always decode data successfully. Thus, I. Salhi, *et al.* [16] proposed another reliable coding scheme where a relay node can produce much better packet combination through the exchange of neighboring table and topology interference information. Finally, Z. Kiss, *et al.* [17] tried to combine two coding techniques, RS coding and network coding. Once a relay node receives data packets transmitted from all the downstream sensor nodes, it begins to chop every data packet into smaller segments. Each segment is encoded into two symbols using RS codes. The two symbols are then converted to two checking blocks before data transmission. A receiving node can therefore increase the probability of correct data by decoding these two checking blocks.

Unlike the previous research work, where the authors focused on reliable routing, information coding, and network coding, in this paper, we propose a packet interleaving scheme (PIS) to reduce the impact of burst errors (BE) on high-reliability-required (HRR) data in WSNs. BE is defined as a sequence of transmitted bits which may encounter errors in continuous bit pattern. Although Reed-Solomon (RS) codes [18], which encode data in a block-by-block fashion, may correct bit errors in a certain constraints, it may not be economically worthy in dealing with burst errors when the number of consecutive bit errors exceeds a threshold. In other words, it makes the overhead of redundancy codes become unacceptable. Hence, to increase packet correctable probability in a WSN, the proposed PIS first classifies the collected data into two different types: HRR data and non-HRR data. An HRR packet is encoded with a short RS symbol, while a non-HRR packet with a long RS symbol. When an HRR and a non-HRR packet arrive at a sensor, they are interleaved on a symbol-by-symbol basis. The noticeable benefit from the packet interleaving is that the burst errors are dispersed and the uncorrectable probability of HRR packets is significantly reduced.

The remainder of this paper is organized as follows. In Section 2, we introduce RS codes employed in a WSN. In Section 3, the proposed PIS and its operations are described. In Section 4, an analytical model is built using two bit-error models, the uniform bit-error model (UBEM) and the on-off bit-error model (OBEM). Finally,

conclusions are drawn in Section 5.

2. Reed-Solomon Codes

2.1. Forward Error Correction

Forward Error Correction (FEC) codes have been widely used in today’s unreliable communication system, such as wireless networks, to reduce bit error rate (BER). The basic principle behind FEC is to generate redundancy code by encoding the user data; the former is then appended to the latter before transmission. At the receiver, user data together with the redundancy code are examined so that a certain pattern of bit errors can be corrected. FEC is composed of two categories, convolution code (CC) and block code (BC) [3]. The former has been applied to the transmission for continuous data, such as video streaming. Encoding process in CC requires relatively larger memory to store the previous, the current, and the upcoming information. As shown in **Figure 1(a)**, each symbol in the k symbols, $(m_0, m_1, \dots, m_{k-1})$ is Exclusive-ORed with the previous or the upcoming symbols. As a result, n symbols $(c_0, c_1, \dots, c_{n-1})$ are produced from k symbols. BC, on the other hand, has been used in encoding discrete data on the basis of block by block. Thus, BC is more applicable to a sensor network, where data, such as temperature, moisture, etc., are collected discretely in a block size. The basic idea behind BC is to encode k -symbol information with a division operation. Redundancy codes with $n-k$ symbols, $(r_0, r_1, \dots, r_{n-k-1})$, are generated after the division operation. Before the transmission to an upper stream node, the $n-k$ symbols are attached to the original k symbols, as shown in **Figure 1(b)**.

2.2. RS Encoding

1) It is well known that Reed-Solomon (RS) code is one of the most popular block codes. RS Code is usually represented by $GF(2^m)$, where GF is the Galois Field [19] and m denotes the number of bits in a symbol. Alternatively, RS code can be represented by (n, k) , where $n = 2^m - 1$ denotes the total number of symbols encoded, k is number of symbols for user data, and $(n-k)$ denotes the number of redundancy symbols. Hence, $t = \frac{n-k}{2}$ denotes the number of erroneous symbols that can be corrected.

2) Let us briefly review the operations of RS codes. Before a message is transmitted using RS encoding, the message is segmented into a number of m -bit symbols. The k m -bit symbols can be represented by a polynomial: $M(X) = \sum_{i=0}^{k-1} m_i X^i = m_{k-1} X^{k-1} + m_{k-2} X^{k-2} + \dots + m_1 X^1 + m_0$. A generator polynomial, $G(X)$, is defined as $G(X) = \prod_{i=0}^{2t-1} (X - \alpha_i) = (X - \alpha_0)(X - \alpha_1) \dots (X - \alpha_{2t-1})$, which is used to calculate a parity polynomial, $P(X)$; *i.e.*, $P(X) = M(X) X^{2t} \text{ mod } G(X)$. Finally, codeword, $C(X)$, is calculated by $C(X) = M(X) X^{2t} + P(X)$. The coefficients of polynomial $C(X)$ are the final encoded data to be transmitted.

3) For example, a RS code has $m = 3$ and $k = 3$ or $(n, k) = (7, 3)$; *i.e.*, a message is segmented into a num-

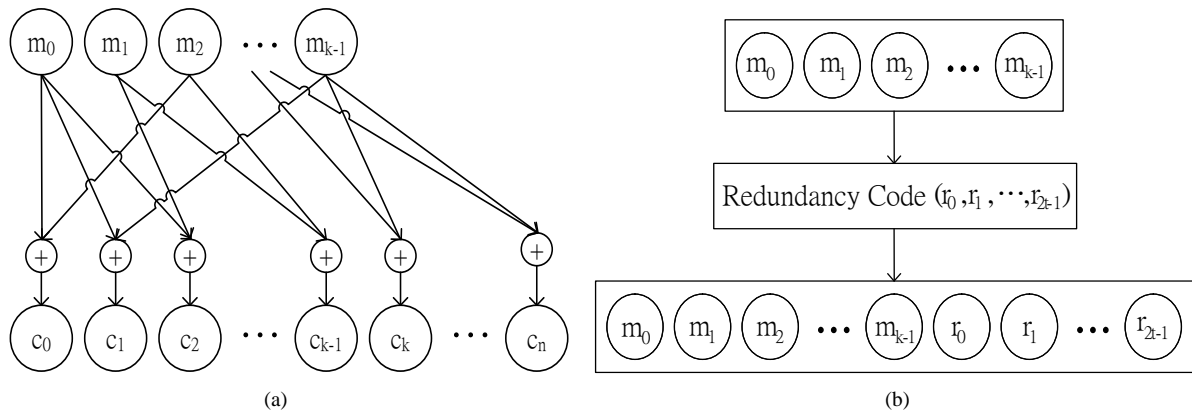


Figure 1. Convolution code versus block code. (a) Convolution Code; (b) Block Code.

ber of 3-bit symbols. The 8 different symbols and their binary representations are shown in **Table 1**.

We assume the original user data is 9 bits: 001 010 011. Since a symbol consists of 3 bits and $t = (n - k) / 2 = (7 - 3) / 2 = 2$. From **Table 1**, the three symbols are represented by $\alpha_1 \alpha_2 \alpha_3$. Thus, we have

$$M(X) = \sum_{i=0}^2 m_i X^i = \alpha_1 X^2 + \alpha_2 X^1 + \alpha_3 \quad \text{and}$$

$$G(X) = \prod_{i=0}^3 (X - \alpha_0)(X - \alpha_1)(X - \alpha_2)(X - \alpha_3) = X^4 + \alpha_3 X^3 + \alpha_0 X^2 + \alpha_1 X + \alpha_3.$$

Parity polynomial is computed as $P(X) = M(X)X^4 \bmod G(X) = 0X^3 + 0X^2 + \alpha_0 X + \alpha_3$. Thus, $C(X) = M(X)X^4 + P(X) = \alpha_1 X^6 + \alpha_2 X^5 + \alpha_4 X^4 + \alpha_0 X^3 + \alpha_0 X^2 + \alpha_1 X + \alpha_4$. From **Table 1**, we know the encoded data to be transmitted are 001 010 100 000 000 001 100.

2.3. RS Decoding

Due to noise or interference, data may encounter bit errors over wireless link. Let $R(X)$ be the data received, $C(X)$ be the data transmitted, and $E(X)$ be the data in error. Thus, we have, $R(X) = C(X) + E(X)$. To determine whether or not $R(X)$ requires any corrections, we have to compute $E(X)$ by their syndromes: $(S_1, S_2, \dots, S_{2t})$. From Equation (1), if $S_i = 0$, then $E(\alpha_i) = 0$, which implies $R(\alpha_i) = C(\alpha_i)$; *i.e.*, no bit errors occur. Otherwise, bit errors occur and $E(X)$ is computed.

$$S_i = R(\alpha_i) = C(\alpha_i) + E(\alpha_i), \quad i = 1, 2, \dots, 2t \quad (1)$$

If bit errors do occur, error locations and error values are required computation. Assume there are v errors appear at the locations, $X^{l^1}, X^{l^2}, \dots, X^{l^v}$. That is, $E(X) = X^{l^1} + X^{l^2} + \dots + X^{l^v}$ and their syndromes can be expressed as in Equation (2).

$$\begin{cases} S_1 = E(\alpha_1) = \alpha_1^{l^1} + \alpha_1^{l^2} + \dots + \alpha_1^{l^v} \\ S_2 = E(\alpha_2) = \alpha_2^{l^1} + \alpha_2^{l^2} + \dots + \alpha_2^{l^v} \\ \vdots \\ S_{2t} = E(\alpha_{2t}) = \alpha_{2t}^{l^1} + \alpha_{2t}^{l^2} + \dots + \alpha_{2t}^{l^v} \end{cases} \quad (2)$$

Next, if we can find $(\alpha^{l^1}, \alpha^{l^2}, \dots, \alpha^{l^v})$ such that all syndromes are equal to zero, then the errors can be located from the exponents of α . Let $\beta^i = \alpha^{l^i}, \forall i = 1, 2, \dots, v$. The syndromes can be expressed as in Equation (3).

$$\begin{cases} S_1 = \beta_1^1 + \beta_1^2 + \dots + \beta_1^v = \sum_{i=1}^v \beta_1^i \\ S_2 = \beta_2^1 + \beta_2^2 + \dots + \beta_2^v = \sum_{i=1}^v \beta_2^i \\ \vdots \\ S_{2t} = \beta_{2t}^1 + \beta_{2t}^2 + \dots + \beta_{2t}^v = \sum_{i=1}^v \beta_{2t}^i \end{cases} \quad (3)$$

Table 1. An example of RS code with 8 symbols.

Symbols	$b_2 b_1 b_0$
α_0	000
α_1	001
α_2	010
α_3	011
α_4	100
α_5	101
α_6	110
α_7	111

Thus, the polynomial of error locations can be expressed as shown in Equation (4).

$$\sigma(X) = \prod_{i=1}^v (1 + \beta^i) = \sigma_v X^v + \sigma_{v-1} X^{v-1} + \cdots + \sigma_1 X + 1 \quad (4)$$

To compute σ_i , we let $X = (\beta^i)^{-1}$ and substitute it into Equation (4). Next, we let $\sigma(X) = 0$, which results to Equation (5), where $i = 1, 2, \dots, v$ and $l = 1, 2, \dots, 2t - v$. From Equation (5), we can derive Equation (6).

$$\sigma_v (\beta^i)^l + \sigma_{v-1} (\beta^i)^{l+1} + \cdots + (\beta^i)^{l+v} = 0 \quad (5)$$

$$\sum_{i=1}^v \left(\sigma_v (\beta^i)^l + \sigma_{v-1} (\beta^i)^{l+1} + \cdots + (\beta^i)^{l+v} \right) = 0 \quad (6)$$

Expanding Equation (6), the result is shown in Equation (7).

$$\sum_{i=1}^v \sigma_v (\beta^i)^l + \sum_{i=1}^v \sigma_{v-1} (\beta^i)^{l+1} + \cdots + \sum_{i=1}^v (\beta^i)^{l+v} = 0 \quad (7)$$

From Equation (3) and Equation (7), we can derive Equation (8).

$$\sigma_v S_l + \sigma_{v-1} S_{l+1} + \cdots + S_{l+v} = 0 \quad (8)$$

Substituting $l = 1, 2, \dots, 2t - v$ to Equation (8), we have derive Equation (9).

$$\begin{cases} S_1 \sigma_v + S_2 \sigma_{v-1} + \cdots + S_v \sigma_1 = -S_{v+1} \\ S_2 \sigma_v + S_3 \sigma_{v-1} + \cdots + S_{v+1} \sigma_1 = -S_{v+2} \\ \vdots \\ S_{2t-v} \sigma_v + S_{2t-v+1} \sigma_{v-1} + \cdots + S_{2t-1} \sigma_1 = -S_{2t} \end{cases} \quad (9)$$

Equation (9) can be expressed by a matrix equation as shown in Equation (10). By applying the operation of inverse matrix, we can compute σ_i .

$$\begin{bmatrix} S_1 & S_2 & \cdots & S_v \\ S_2 & S_3 & \cdots & S_{v+1} \\ \vdots & \vdots & \ddots & \vdots \\ S_v & S_{v+1} & \cdots & S_{2t-1} \end{bmatrix} \begin{bmatrix} \sigma_v \\ \sigma_{v-1} \\ \vdots \\ \sigma_1 \end{bmatrix} = \begin{bmatrix} -S_{v+1} \\ -S_{v+2} \\ \vdots \\ -S_{2t} \end{bmatrix} \quad (10)$$

Substituting σ_i to Equation (4), we can find β^i ; i.e., the error locations. To compute error values, we define them as $e'_v, e'_{v-1}, \dots, e'_1$ as shown in Equation (11).

$$R(X) = C(X) + E'(X) = C(X) + e'_v X^v + e'_{v-1} X^{v-1} + \cdots + e'_1 X \quad (11)$$

Let $X = \alpha_1, \alpha_2, \dots, \alpha_v$. By substituting X to Equation (11), we have derive Equation (12).

$$\begin{cases} e'_v \alpha_1^v + e'_{v-1} \alpha_1^{v-1} + \cdots + e'_1 \alpha_1 = S_1 \\ e'_v \alpha_2^v + e'_{v-1} \alpha_2^{v-1} + \cdots + e'_1 \alpha_2 = S_2 \\ \vdots \\ e'_v \alpha_v^v + e'_{v-1} \alpha_v^{v-1} + \cdots + e'_1 \alpha_v = S_v \end{cases} \quad (12)$$

Since Equation (12) has v equations and v unknown, it is easy to compute v . Finally, errors can be corrected by adding error values v to their corresponding positions. At last, let us give an example. We assume the received data = 001 010 000 011 000 001 011, which can be represented by a polynomial $R(X) = \alpha_1 X^6 + \alpha_2 X^5 + \alpha_0 X^4 + \alpha_3 X^3 + \alpha_0 X^2 + \alpha_1 X + \alpha_3$. We then calculate the 4 syndromes, $S_1 = R(\alpha_1) = \alpha_2$, $S_2 = R(\alpha_2) = \alpha_1$, $S_3 = R(\alpha_3) = \alpha_6$, and $S_4 = R(\alpha_4) = \alpha_6$. By substituting them into Equation (10), we obtain the error locations as shown in Equation (13).

$$\begin{bmatrix} \alpha_3 & \alpha_5 \\ \alpha_5 & \alpha_6 \end{bmatrix} \begin{bmatrix} \sigma_2 \\ \sigma_1 \end{bmatrix} = \begin{bmatrix} \alpha_6 \\ 0 \end{bmatrix} \quad (13)$$

$$\begin{bmatrix} \sigma_2 \\ \sigma_1 \end{bmatrix} = \begin{bmatrix} \alpha_1 & \alpha_0 \\ \alpha_0 & \alpha_5 \end{bmatrix} \begin{bmatrix} \alpha_6 \\ 0 \end{bmatrix} = \begin{bmatrix} \alpha_0 \\ \alpha_6 \end{bmatrix} \quad (14)$$

By multiplying an inverse matrix on both sides in Equation (13), we have derive Equation (14). Next, if we substitute σ_2 and σ_1 to an error-location polynomial, we have $\sigma(X) = \alpha_0 X^2 + \alpha_6 X + \alpha_0$. Now, we can derive $\sigma(\alpha_0) = \alpha_6$, $\sigma(\alpha_1) = \alpha_2$, $\sigma(\alpha_2) = \alpha_6$, $\sigma(\alpha_3) = 0$, $\sigma(\alpha_4) = 0$, $\sigma(\alpha_5) = \alpha_2$, and $\sigma(\alpha_6) = \alpha_0$. Hence, the two error locations are found, which show us the third and the fourth symbols are in error. To compute these two error values, we need to compute S_1 and S_2 first, as shown in Equation (15). Thus, the two error values can be computed from Equation (16).

$$\begin{bmatrix} \alpha_1^3 & \alpha_1^4 \\ \alpha_2^3 & \alpha_2^4 \end{bmatrix} \begin{bmatrix} e'_1 \\ e'_2 \end{bmatrix} = \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} \quad (15)$$

$$\begin{bmatrix} e'_1 \\ e'_2 \end{bmatrix} = \begin{bmatrix} \alpha_2 & \alpha_5 \\ \alpha_0 & \alpha_4 \end{bmatrix} \begin{bmatrix} \alpha_2 \\ \alpha_1 \end{bmatrix} = \begin{bmatrix} \alpha_3 \\ \alpha_3 \end{bmatrix} \quad (16)$$

Since $E(X) = \alpha_3 X^4 + \alpha_3 X^3$, the received data are corrected as,

$C'(X) = R(X) + E(X) = \alpha_1 X^6 + \alpha_2 X^5 + \alpha_3 X^4 + \alpha_0 X^3 + \alpha_0 X^2 + \alpha_1 X + \alpha_3$, which in binary representation is 001 010 011 000 000 001 011.

3. Packet Interleaving Scheme

3.1. WSN with Multi-Hop Tree Structure

In a wireless sensor network (WSN), a coordinator is the sink which gathers all the data collected from other distant sensor nodes. To facilitate data gathering from all the sensor nodes, it is very constructive that the coordinator and the sensor nodes will collaborate to build a multi-hop tree structure (MTS), as shown in **Figure 2**. In an MTS-based WSN, data collected by a sensor node will be forwarded hop-by-hop to the coordinator. Thus, by fully utilizing a branch node of the MTS, in this paper, we propose a packet interleaving scheme (PIS) based on RS codes to reduce the impact of burst errors on packet uncorrectable probability.

3.2. Packet Interleaving

In the proposed PIS, packets collected by a sensor node are classified into two different types: high-reliability required (HRR) packet and non-HRR packet. An HRR packet is defined as a packet which requires for retransmission, if uncorrectable burst errors exist. Payload in an HRR packet consists of numerical data, such as temperature, moisture, luminance, etc. This type of packet has relatively shorter data length (usually, a couple of bytes) and each uncorrectable HRR packet requires for retransmission. Hence, it is better to employ a shorter-length symbol (in this paper, we use $m = 4$) to encode an HRR packet with shorter data length. On the other

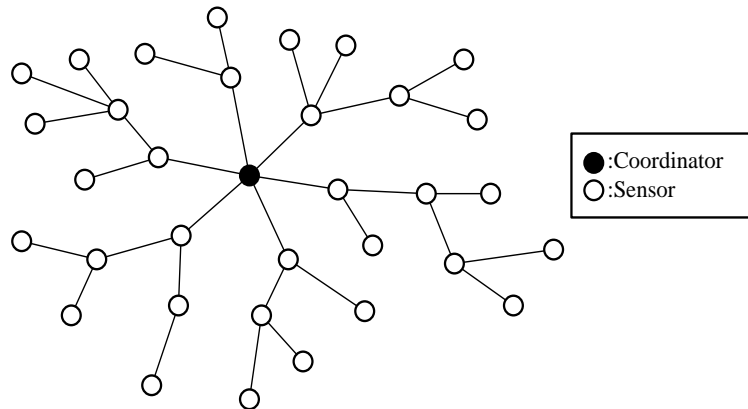


Figure 2. WSN with multi-hop tree structure.

hand, payload in a non-HRR packet consists of non-numerical data, such as video, audio, etc. This type of packet has relatively longer data length (usually, in the order of kilo bytes) and each uncorrectable non-HRR packet may not require for retransmission. Thus, it is better to employ longer-length symbol (in this paper, we use $m = 8$) to encode a non-HRR packet with longer data length.

As it is illustrated in **Figure 3**, an HRR packet is encoded with a shorter-length RS symbol (i.e., $m = 4$). First, an M-byte MAC-layer header and payload is converted to $\frac{(M \times 8)}{4} = 2M$ symbols on the basis of 4 bits per symbol. Thus, we have the length of a codeword is n , where $n = 2^m - 1 = 2^4 - 1 = 15$, the number of symbols for user data in a codeword is k , where $k = n - 2 \times t = 15 - 2 \times t$, and the number of symbols for redundancy code in a codeword is $2 \times t$. Let $N = \left\lceil \frac{2M}{k} \right\rceil$, which denotes the number of codeword required for encoding the M-byte packet (header plus payload). The total redundancy code (or FCS) is therefore equal to $N \times 2 \times t$ symbols, or $N \times 2 \times t \times 4 (= N \times t \times 8)$ bits.

Similarly, as it is illustrated in **Figure 4**, a non-HRR packet is encoded with a longer-length symbol ($m = 8$). An M-byte MAC-layer header and payload is converted to $\frac{(M \times 8)}{8} = M$ symbols on the basis of 8 bits per symbol. Since the length of a codeword, $n = 2^m - 1 = 2^8 - 1 = 255$ bytes, which is much greater than the maximum length of a packet, which is 127 bytes in a WSN, a codeword is sufficiently enough to encode a MAC-layer packet in a WSN. Thus, the length of redundancy code (or FCS) is equal to $2 \times t$ bytes.

When both an HRR and a non-HRR packet are received by a branch node in a tree-structured WSN, these two packets are interleaved on a symbol-by-symbol basis, as shown in **Figure 5**. The interleaved packet is then forwarded to an upper stream node, which performs decoding and correction process. However, as shown in **Figure 6**, an interleaved packet may not be correctable, if it encounters burst errors where the number of total errors is greater than t . In the proposed PIS, by separating the interleaved single packet back to their original two packets, each individual packet may become correctable. This is because the number of errors in each separated packet is highly possible to be smaller than t .

4. Mathematical Model and Analysis

In RS codes, whether or not a coded packet is correctable is mainly determined by the number of erroneous symbols. In this section, to study the robustness and the error-correction capability of the proposed PIS, we build two mathematical models for comprehensive numerical simulations. The first one is referred to as uniform bit error model (UBEM), while the second one is referred to as on-off bit error model (OBEM). The first model assumes the errors occur evenly on the coded packets. The second model assumes the errors may occur continuously

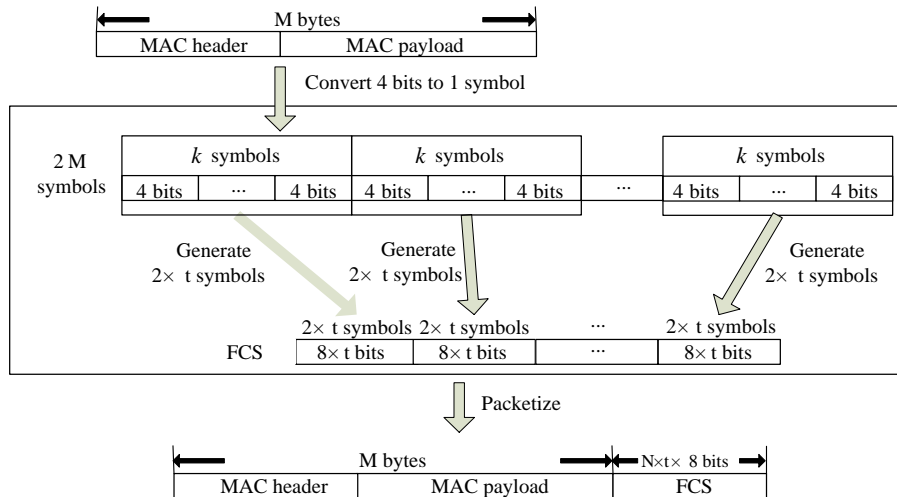


Figure 3. An HRR packet encoded with a shorter-length symbol ($m = 4$).

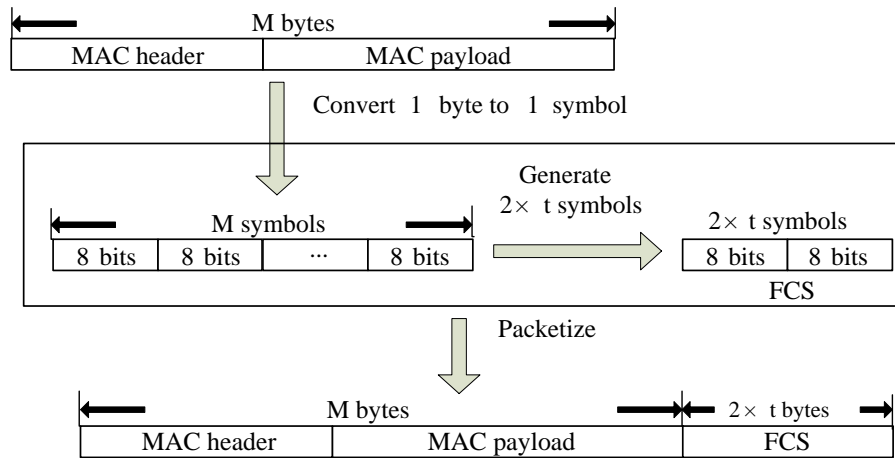


Figure 4. A non-HRR packet encoded with a longer-length symbol ($m = 8$).

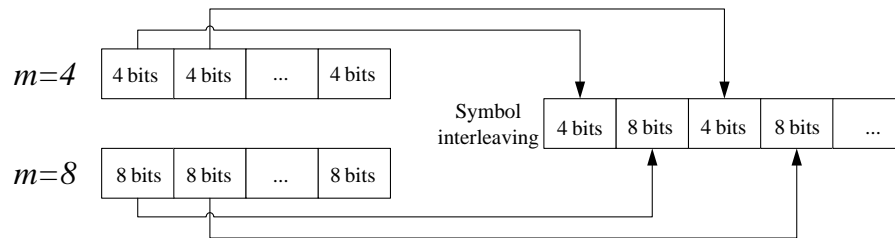


Figure 5. Packet interleaving on a symbol-by-symbol basis.

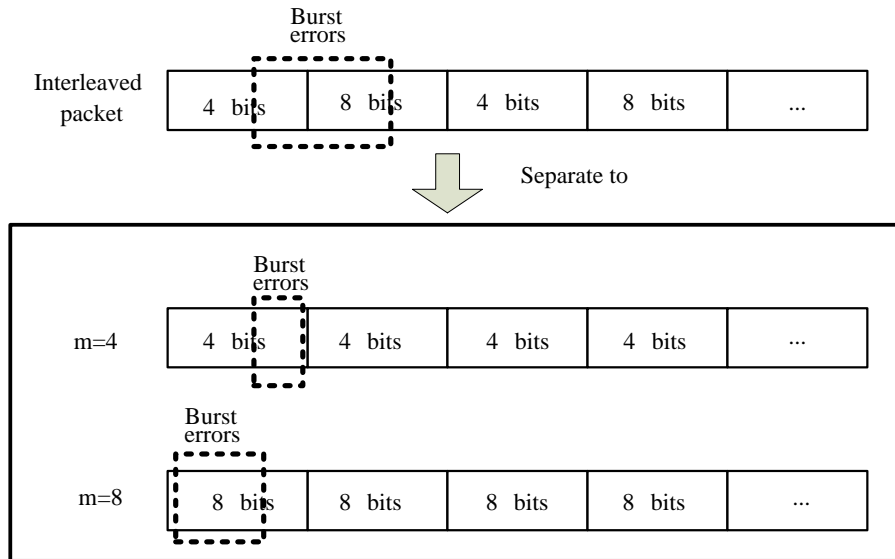


Figure 6. Burst errors dispersed on two symbols.

in a burst length.

4.1. UBEM

Let $P_{UB_{be}}$ and $P_{UB_{se}}$ denote the probability of bit errors and the probability of symbol errors, respectively. Since any bit errors occur in a symbol may result in a symbol error and each symbol has m bits, we can derive

P_{UB_se} directly from P_{UB_be} , as shown in Equation (17).

$$P_{UB_se} = 1 - (1 - P_{UB_be})^m \quad (17)$$

Next, let us define two more parameters, N_s and N_c . The first parameter denotes the number of symbols in a codeword and the second parameter denotes the number of codeword in a packet. Hence, the uncorrectable probability of a codeword (P_{UB_cuc}), can be derived as shown in Equation (18).

$$P_{UB_cuc} = 1 - \sum_{i=0}^t \binom{N_s}{i} \times (P_{UB_se})^i \times (1 - P_{UB_se})^{N_s-i} \quad (18)$$

In Equation (18), we know in a codeword if the number of symbol errors is smaller than t , then the codeword is correctable. Thus, the uncorrectable probability of a codeword can be summed up from $i = 0$ to t , since there are $\binom{N_s}{i}$ different types of errors. Next, let us define P_{UB_puc} as the packet uncorrectable probability.

Since there are N_c codeword in a packet, we can derive P_{UB_puc} as shown in Equation (19).

$$P_{UB_puc} = 1 - (1 - P_{UB_cuc})^{N_c} \quad (19)$$

4.2. OBEM

The on-off bit error model (OBEM) is illustrated in **Figure 7**. All the parameters used in the analysis are defined in **Table 2**. A burst error (BE) period is defined as two consecutive bit error intervals where high bit errors ap-

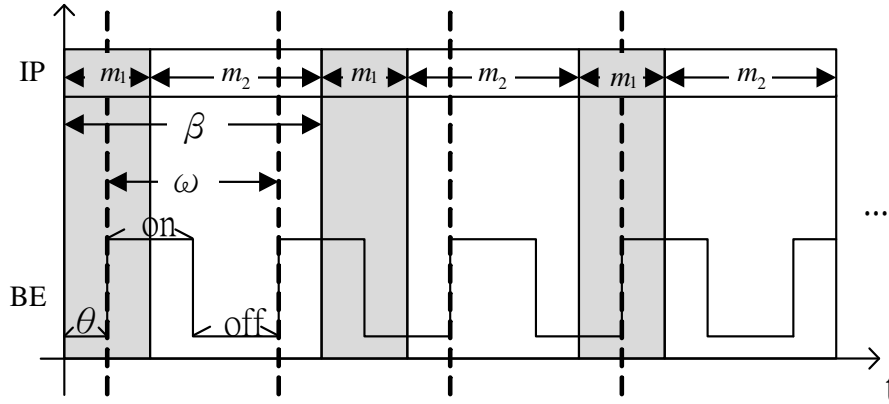


Figure 7. On-Off bit error model.

Table 2. Parameters used in OBEM.

m_1	Number of bits in a symbol of HRR packet
m_2	Number of bits in a symbol of non-HRR packet
β	Packet interleaved length in bits ($m_1 + m_2$)
on	Length of high bit errors (in bits)
σ	Error probability of high bit errors
off	Length of low bit errors (in bits)
ρ	Error probability of low bit errors
ω	Burst length ($on + off$) (in bits)
θ_0	Length of initial right-shift position (in bits)

pear first and then followed by low bit errors. Notice that θ_0 is defined as the length of right-shift position for an initial BE period; $\theta_0 = 0$ implies that no gap exists between the beginning of an interleaved packet and the beginning of the first BE period.

First, we define P_{OB_se} as the symbol-error probability in OBEM. An interleaved packet (IP) and a burst error (BE) may have different lengths; here we assume the former has a length of β bits ($\beta = m_1 + m_2$) and the later has a length of ω bits ($\omega = on + off$). Since every symbol in an IP may encounter different positions of bit errors, we have to analyze the bit error positions of a symbol before we can compute the symbol-error probability. To compute the error probability of the α^{th} symbol, we define (i) m_{1s} = the distance between the first bit of m_1 and the first bit of an IP, and (ii) m_{1e} = the distance between the last bit of m_1 and the first bit of an IP. Similarly, we define m_{2s} and m_{2e} for m_2 . Thus, m_{1s} , m_{1e} , m_{2s} , and m_{2e} can be computed as shown in Equation (20), (21), (22), and (23), respectively.

$$m_{1s} = \alpha \times \beta \quad (20)$$

$$m_{1e} = \alpha \times \beta + m_1 - 1 \quad (21)$$

$$m_{2s} = \alpha \times \beta + m_1 \quad (22)$$

$$m_{2e} = (\alpha + 1) \times \beta - 1 \quad (23)$$

For simplicity, the four parameters, m_{1s} , m_{1e} , m_{2s} , and m_{2e} are generalized to m_{ij} , where $i = 1, 2$ and $j = s, e$. Let θ denote the length of right-shift position between an IP and a BE at the α^{th} symbol. Let $\Omega_{m_{ij}}$ denote the length of right-shift position for m_{1s} , m_{1e} , m_{2s} , and m_{2e} at the α^{th} symbol. Thus, we can compute θ and $\Omega_{m_{ij}}$ as shown in Equation (24) and (25), respectively.

$$\theta = \left(\theta_0 + \left(\left\lceil \frac{\text{header length}}{\omega} \right\rceil \times \omega - \text{header length} \right) \right) \bmod \omega \quad (24)$$

$$\Omega_{m_{ij}} = \left(\left\lceil \frac{m_{ij} + 1}{\omega} \right\rceil - 1 \right) \times \omega + \theta \quad (25)$$

After we found the right-shift position between an IP and a BE, we can categorize the symbol errors into four cases, as illustrated from **Figure 8(a)** to **Figure 8(d)**. Case 1 in **Figure 8(a)** shows whether or not a symbol may occupy one BE or two BE periods, their start bit and stop bit of a symbol all appear at the high-bit-error interval. Case 2 in **Figure 8(b)** shows only the start bit of a symbol appears at the high-bit-error interval. Case 3 in **Figure 8(c)** shows whether or not a symbol may occupy one BE or two BE periods, neither the start bit nor the stop bit appear at the high-bit-error interval. Finally, Case 4 in **Figure 8(d)** shows only the stop bit of a symbol appears at the high-bit-error interval. Actually, the four different cases of symbol errors can be constrained by eight inequalities with four parameters, m_{ij} , $\Omega_{m_{ij}}$, on , and ω . These eight inequalities (two inequalities for each case) are shown in **Table 3**. Once we identify the four cases of symbol errors, we can compute the symbol-error probability. Since one codeword consists of N_s symbols, we define $P_{OB_se}(\alpha)$, $\alpha = 0, 1, \dots, N_s - 1$, as the symbol-error probability of the α^{th} symbol. Let σ denote the error probability of high bit errors and ρ denote the error probability of low bit errors. Let N_σ represent the number of bits with errors in a symbol and N_ρ represent the number of bits without errors in a symbol. We can compute $P_{OB_se}(\alpha)$ as shown in Equation (26).

$$P_{OB_se}(\alpha) = 1 - (1 - \sigma)^{N_\sigma} \times (1 - \rho)^{N_\rho} \quad (26)$$

Now, we can compute N_σ and N_ρ for case 1 as shown in Equation (27) and (28), case 2 as shown in Equation (29) and (30), case 3 as shown in Equation (31) and (32), and case 4 as shown in Equation (33) and (34), respectively.

Case 1:

$$N_\rho = \left(\left\lceil \frac{m_{1e} + 1}{\omega} \right\rceil - 1 \right) - \left(\left\lceil \frac{m_{1s} + 1}{\omega} \right\rceil - 1 \right) \times Off = \left(\left\lceil \frac{m_{1e} + 1}{\omega} \right\rceil - \left\lceil \frac{m_{1s} + 1}{\omega} \right\rceil \right) \times Off \quad (27)$$

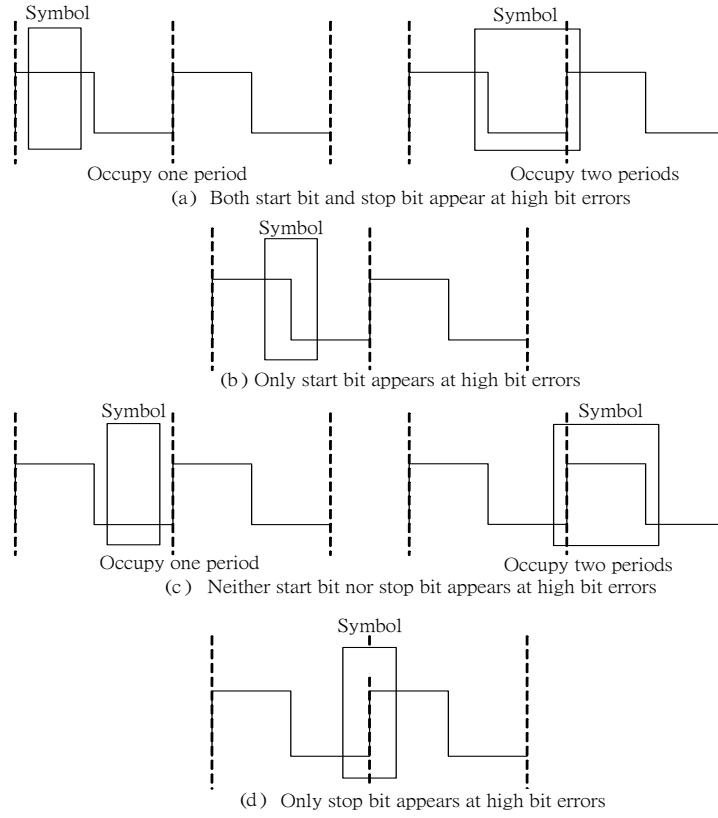


Figure 8. Four cases of symbol errors in terms of the start and stop bits.

Table 3. Four cases of symbol errors with 8 constraints.

Case	Conditions
1	$\Omega_{m_{is}} \leq m_{is} < \Omega_{m_{is}} + On$ and $\Omega_{m_{ie}} \leq m_{ie} < \Omega_{m_{ie}} + On$
2	$\Omega_{m_{is}} \leq m_{is} < \Omega_{m_{is}} + On$ and $\Omega_{m_{ie}} + On \leq m_{ie} < \Omega_{m_{ie}} + \omega$
3	$\Omega_{m_{is}} + On \leq m_{is} < \Omega_{m_{is}} + \omega$ and $\Omega_{m_{ie}} + On \leq m_{ie} < \Omega_{m_{ie}} + \omega$
4	$\Omega_{m_{is}} + On \leq m_{is} < \Omega_{m_{is}} + \omega$ and $\Omega_{m_{ie}} \leq m_{ie} < \Omega_{m_{ie}} + On$

$$N_{\sigma} = m_i - N_{\rho} = m_i - \left(\left\lfloor \frac{m_{ie} + 1}{\omega} \right\rfloor - \left\lfloor \frac{m_{is} + 1}{\omega} \right\rfloor \right) \times Off \quad (28)$$

Case 2:

$$\begin{aligned} N_{\sigma} &= \left(\left\lfloor \frac{m_{ie} + 1}{\omega} \right\rfloor - 1 \right) - \left(\left\lfloor \frac{m_{is} + 1}{\omega} \right\rfloor - 1 \right) \times On + \Omega_{m_{is}} + On - m_{is} \\ &= \left(\left\lfloor \frac{m_{ie} + 1}{\omega} \right\rfloor - \left\lfloor \frac{m_{is} + 1}{\omega} \right\rfloor \right) \times On + \left(\left\lfloor \frac{m_{is} + 1}{\omega} \right\rfloor - 1 \right) \times \omega + \theta + On - m_{is} \end{aligned} \quad (29)$$

$$N_{\rho} = m_i - N_{\sigma} = m_i - \left\lfloor \frac{m_{ie} + 1}{\omega} \right\rfloor \times On - \left(\left\lfloor \frac{m_{is} + 1}{\omega} \right\rfloor - 1 \right) \times Off - \theta + m_{is} \quad (30)$$

Case 3:

$$N_\sigma = \left(\left\lfloor \frac{m_{ie} + 1}{\omega} \right\rfloor - 1 \right) - \left(\left\lfloor \frac{m_{is} + 1}{\omega} \right\rfloor - 1 \right) \times On = \left(\left\lfloor \frac{m_{ie} + 1}{\omega} \right\rfloor - \left\lfloor \frac{m_{is} + 1}{\omega} \right\rfloor \right) \times On \quad (31)$$

$$N_\rho = m_i - N_\sigma = m_i - \left(\left\lfloor \frac{m_{ie} + 1}{\omega} \right\rfloor - \left\lfloor \frac{m_{is} + 1}{\omega} \right\rfloor \right) \times On \quad (32)$$

Case 4:

$$\begin{aligned} N_\rho &= \left(\left(\left\lfloor \frac{m_{ie} + 1}{\omega} \right\rfloor - 1 \right) - \left(\left\lfloor \frac{m_{is} + 1}{\omega} \right\rfloor - 1 \right) - 1 \right) \times Off + \Omega_{m_{is}} + \omega - m_{is} \\ &= \left(\left\lfloor \frac{m_{ie} + 1}{\omega} \right\rfloor - \left\lfloor \frac{m_{is} + 1}{\omega} \right\rfloor - 1 \right) \times Off + \left(\left\lfloor \frac{m_{is} + 1}{\omega} \right\rfloor - 1 \right) \times \omega + \theta + \omega - m_{is} \end{aligned} \quad (33)$$

$$= \left\lfloor \frac{m_{is} + 1}{\omega} \right\rfloor \times On + \left(\left\lfloor \frac{m_{ie} + 1}{\omega} \right\rfloor - 1 \right) \times Off + \theta - m_{is}$$

$$N_\sigma = m_i - N_\rho = m_i - \left\lfloor \frac{m_{is} + 1}{\omega} \right\rfloor \times On - \left(\left\lfloor \frac{m_{ie} + 1}{\omega} \right\rfloor - 1 \right) \times Off - \theta + m_{is} \quad (34)$$

By substituting N_σ and N_ρ back to Equation (26), we can now summarize $P_{OB_se}(\alpha)$ for the four different cases of symbol errors as shown in **Table 4**.

After we compute the symbol-error probability for the four different cases, our next step is to derive the uncorrectable probability of a codeword and the uncorrectable probability of a packet. We know there are N_c codeword in a packet and the uncorrectable probabilities of the N_c codeword are all different. Let $P_{OB_cuc}(l)$, $l = 0, 1, \dots, N_c - 1$, denote the uncorrectable probability of the l^{th} codeword and let P_{OB_puc} denote the uncorrectable probability of a packet. By using the combination theory of probabilities, we can derive $P_{OB_cuc}(l)$ and P_{OB_puc} as shown in Equation (35) and (36), respectively.

$$P_{OB_cuc}(l) = 1 - \sum_{i=0}^{\tau_i - 1} \left(\sum_{j=0}^{\lambda - 1} \left(\prod_{k=0}^{N_k} \binom{N_k}{r_k^{ij}} \times (P_{OB_se}^k)^{r_k^{ij}} \times (1 - P_{OB_se}^k)^{N_k - r_k^{ij}} \right) \right) \quad (35)$$

$$P_{OB_puc} = 1 - \prod_{l=0}^{N_c - 1} (1 - P_{OB_cuc}(l)) \quad (36)$$

5. Numerical Simulations

In OBEM, to study the influences of the four parameters, (i) the length of high bit errors (*i.e.*, the *On* period), (ii) the length of low bit errors (*i.e.*, the *Off* period), (iii) the right-shift position (*i.e.*, θ), and (iv) the number of correctable symbols (*i.e.*, τ), on the packet uncorrectable probability (*i.e.*, P_{OB_puc}), we perform comprehensive numerical simulation. Table V shows the parameters and their setting used in the simulation. From **Table 5**, we can observe that an *On-Off* period is fixed to 12 bits. Within the 12 bits, the *On* period is varied among three different lengths; shorter length (4 bits), equal length (6 bits), and longer length (8 bits). The length of *Off* period is varied from the other directions (*i.e.*, from longer to shorter length).

Table 4. Four cases of symbol-error probabilities.

Case	$P_{OB_se}(\alpha)$
1	$1 - (1 - \sigma)^{m_i - \left\lfloor \frac{m_{ie} + 1}{\omega} \right\rfloor - \left\lfloor \frac{m_{is} + 1}{\omega} \right\rfloor} \times (1 - \rho)^{\left\lfloor \frac{m_{ie} + 1}{\omega} \right\rfloor - \left\lfloor \frac{m_{is} + 1}{\omega} \right\rfloor} \times Off$
2	$1 - (1 - \sigma)^{\left\lfloor \frac{m_{ie} + 1}{\omega} \right\rfloor - \left\lfloor \frac{m_{is} + 1}{\omega} \right\rfloor - 1} \times Off + \theta - m_{is} \times (1 - \rho)^{\left\lfloor \frac{m_{is} + 1}{\omega} \right\rfloor - 1} \times \omega + \theta + \omega - m_{is}$
3	$1 - (1 - \sigma)^{\left\lfloor \frac{m_{ie} + 1}{\omega} \right\rfloor - \left\lfloor \frac{m_{is} + 1}{\omega} \right\rfloor} \times On \times (1 - \rho)^{\left\lfloor \frac{m_{ie} + 1}{\omega} \right\rfloor - \left\lfloor \frac{m_{is} + 1}{\omega} \right\rfloor} \times On$
4	$1 - (1 - \sigma)^{\left\lfloor \frac{m_{ie} + 1}{\omega} \right\rfloor - \left\lfloor \frac{m_{is} + 1}{\omega} \right\rfloor - 1} \times On - \left(\left\lfloor \frac{m_{ie} + 1}{\omega} \right\rfloor - 1 \right) \times Off - \theta + m_{is} \times (1 - \rho)^{\left\lfloor \frac{m_{is} + 1}{\omega} \right\rfloor - 1} \times \omega + \theta + \omega - m_{is}$

Table 5. Parameter settings.

Parameter	Setting
m_1	4 bits
m_2	8 bits
on	4/6/8 bits
off	8/6/4 bits
σ	0.1
ρ	0.0001

Prior to studying the influences of the three different *On-Off* lengths on packet uncorrectable probability, we illustrate three scenarios in **Figure 9**. When the length of *On* period is shorter, as shown in **Figure 9(a)**, an HRR packet encoded with a shorter length of symbol ($m = 4$, dark-shaded portion) completely falls in the *Off* period, while a non-HRR packet encoded with a longer length of symbol ($m = 8$, light-shaded portion) just has a few bits fall in the *Off* period. When the length of the *On* period is increased to 6 bits, which is equal to the length of the *Off* period, as shown in **Figure 9(b)**, half of the HRR packet falls in the *On* period. The error-bit coverage for an HRR packet becomes even worse when the *On* period is increased to 8 bits, as shown in **Figure 9(c)**. As it can be seen, a whole symbol of an HRR packet is now covered by the *On* period, while only half of a symbol of a non-HRR packet is covered by the *On* period.

Impact of Correctable Symbols

First, we are interested in studying the impact of increasing the number of correctable symbols when the length of high bit errors is shorter than the length of low bit errors; *i.e.*, $On = 4$ bits and $Off = 8$ bits. As shown in Figures 10(a) to 10(e), we observe the variations of packet uncorrectable probability (P_{OB_puc}) when the right-shift position (θ) is increased from 0 to 11 bits and the number of correctable symbols (t) is increased from 1 to 5.

From **Figure 10**, we can observe that both the packet uncorrectable probabilities (P_{OB_puc}) of HRR and non-HRR curves drop off very quickly, when the number of correctable symbols (t) is increased from 1 to 5. Additionally, we can observe that the curves of P_{OB_puc} with PIS (the two dashed lines) are much lower than the curves of P_{OB_puc} without using PIS (the two solid lines). The improvement of P_{OB_puc} by using PIS is more significant when t is small, which is quite beneficial for reducing packet overhead, since the number of redundancy bits in RS codes can be shorter. Another noticeable phenomenon is that although when θ is smaller than 7, the curves of HRR with PIS are completely inverted to the curves of non-HRR with PIS, the curves of HRR with PIS do drop to zero when θ is larger than 7. In other words, on average we have come out a result; *i.e.*, the packet uncorrectable probability of HRR with PIS is significantly lower than that of non-HRR with PIS.

From **Figure 11(a)** to **Figure 11(e)**, we show the packet uncorrectable probabilities when the *on* period (6 bits) is equal to the *Off* period (6 bits). It is not surprising to notice that, for the corresponding t values ($t = 1$ to 5), the four curves in **Figure 11** are much higher than those four curves in **Figure 10**. The reason is quite straightforward, since the period of high bit errors (*i.e.*, the *On* period) is increased from 4 bits to 6 bits. However, on average the proposed PIS can effectively reduce packet uncorrectable probability, as it is compared to the curves without using PIS. In particular, we can observe that, no matter what the value t is, the curves of HRR with PIS can be reduced to near zero when θ is larger than 7. Additionally, we observe that a side effect could occur. That is, the curves of non-HRR with PIS cannot be reduced very significantly, as they are compared to those curves in **Figure 10**. This is because no matter how we adjust the right-shift position (θ), a non-HRR packet will be affected by the *on* period since the latter has been increased from 4 bits to 6 bits.

From **Figure 12(a)** to **Figure 11(e)**, we show the packet uncorrectable probabilities when the *on* period (8 bits) is longer than the *Off* period (4 bits). It is interesting to notice that when t is larger than 3 and θ is smaller than 6, the packet uncorrectable probabilities of HRR with PIS are higher than those curves without PIS. Of course, when t is smaller than 3 and θ is larger than 6, the situations are completely inverted. Hence, from **Figure 12**, we have discovered that when the period of high bit errors exceeds the length of an HRR symbol ($m = 4$) and approaches to a non-HRR symbol ($m = 8$), it is better to encode packets with a small value of t ; otherwise, there

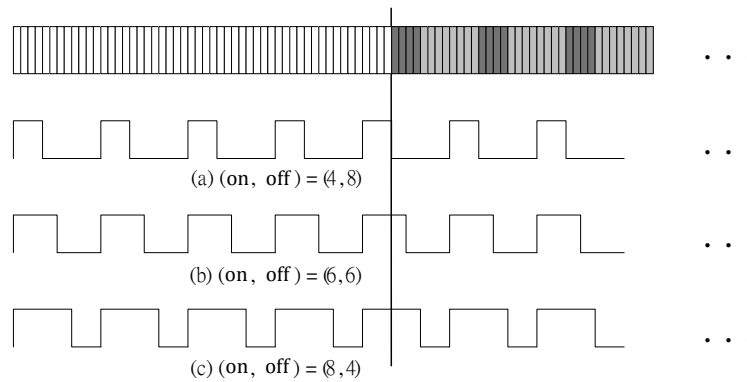


Figure 9. Influences of three *On-Off* lengths on the packet errors.

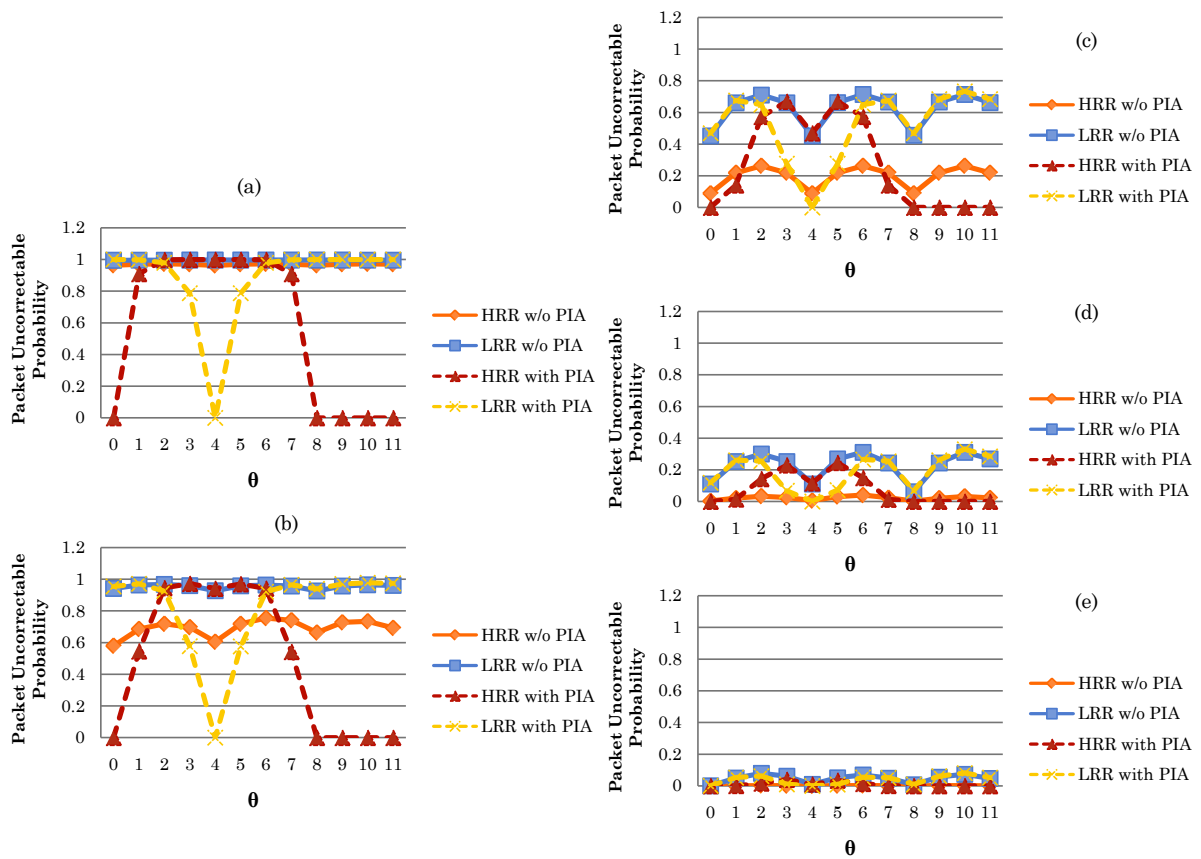


Figure 10. Variations of packet uncorrectable probability (*on* is smaller than *off*).

is no advantage achieved by using the proposed PIS.

From **Figure 13(a)** to **Figure 13(c)**, we show the comparisons in packet uncorrectable probabilities between UBEM and OBEM. First, we observe that in UBEM it does not make any big difference whether or not using the proposed PIS, since in UBEM every bit receives the same error probability. Second, we observe that packet uncorrectable probability in UBEM is relatively higher than that in OBEM. Notice that the curves of UBEM and the curves of OBEM vary along with the following two parameters: (i) when the number of correctable symbols (t) increases from 1 to 5, the curves of UBEM disserve very quickly from those of OBEM; and (ii) when the right-shift position (θ) increases from zero to 8 bits, the gap between these two models becomes smaller. Since OBEM is more reactive to a real word than UBEM, it is rewarding to know that the proposed PIS can reduce packet uncorrectable probability in OBEM more significantly than that in UBEM. Another noticeable re-

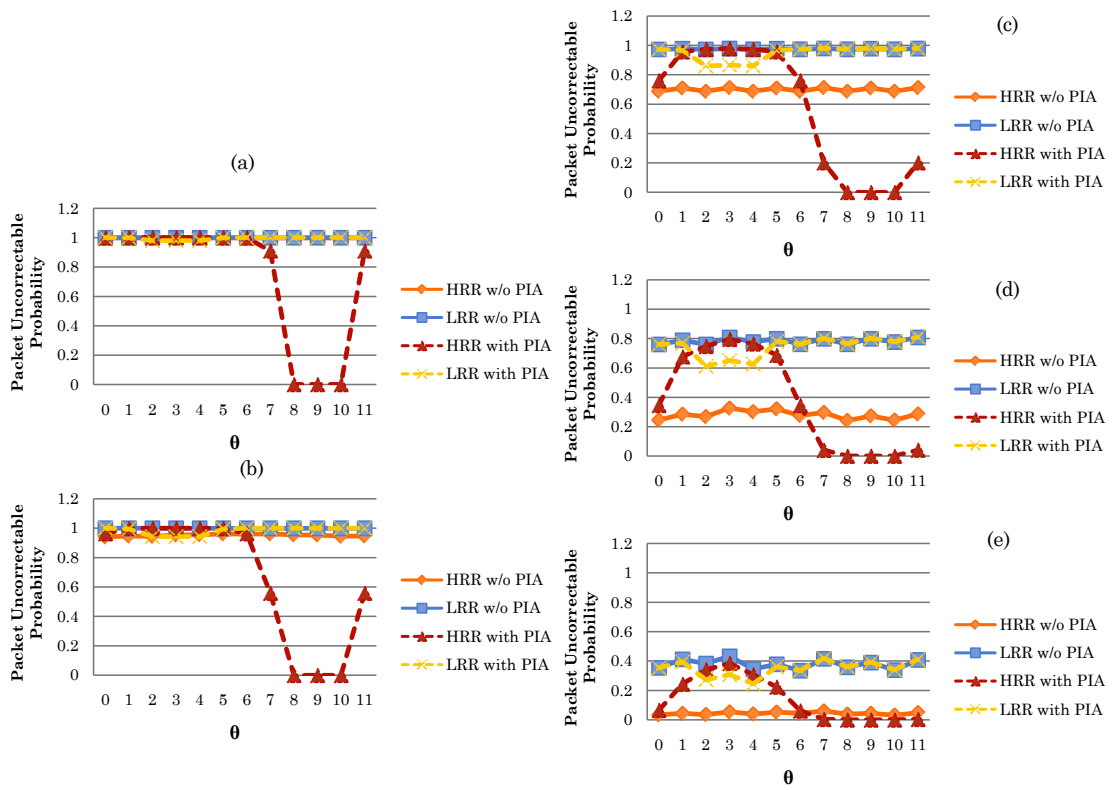


Figure 11. Variations of packet uncorrectable probability (*on* is equal to *off*).

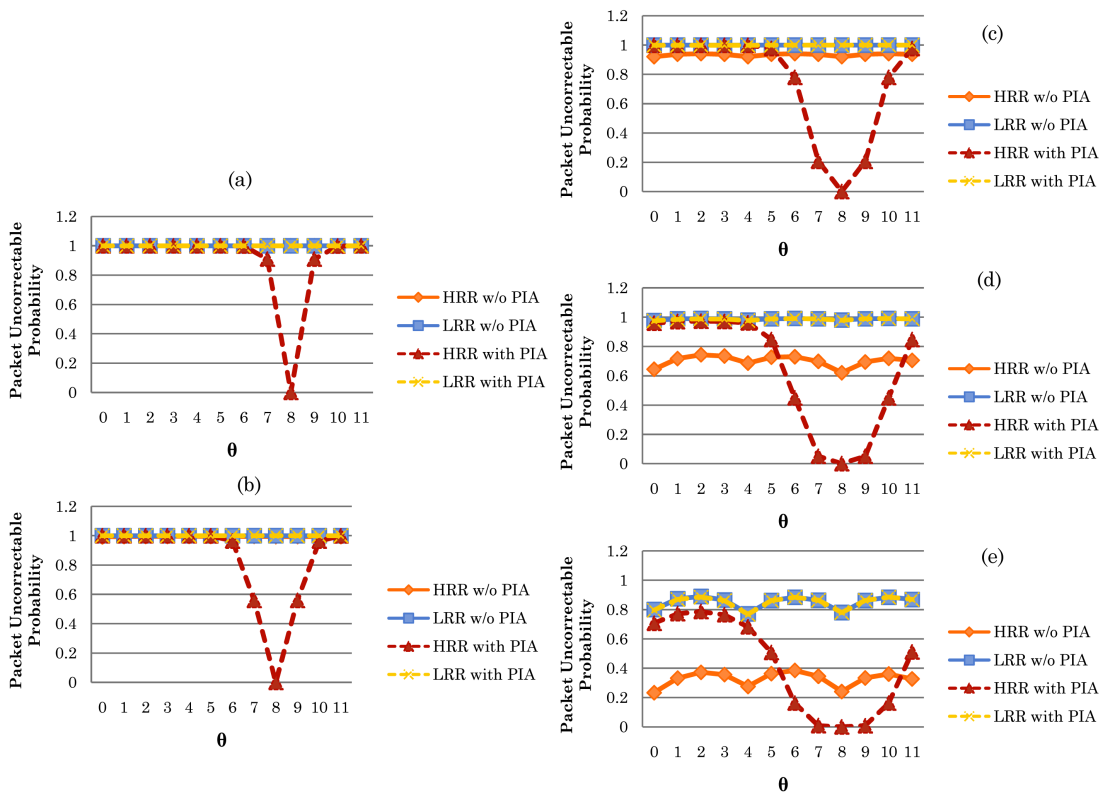


Figure 12. Variations of packet uncorrectable probability (*on* is larger than *off*).

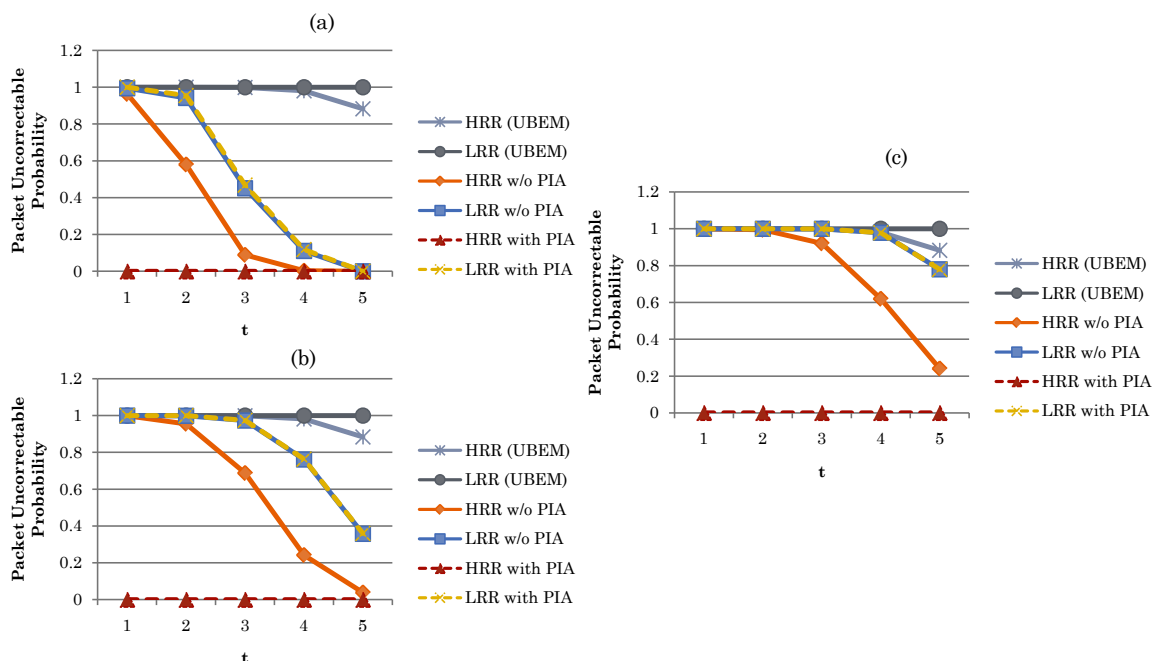


Figure 13. Packet uncorrectable probability in UBEM vs in OBEM. (a) $(On, Off, \theta) = (4, 8, 0)$; (b) $(On, Off, \theta) = (6, 6, 8)$; (c) $(On, Off, \theta) = (4, 8, 8)$

sult is that no matter how we increase the period of high bit errors (*i.e.*, we increase the *on* period from 4 bits in 13(a) to 6 bits in 13(b), and finally to 8 bits in 13(c)), HRR with PIS in OBEM always exhibits the lowest packet uncorrectable probability (near zero, in some cases). The relatively lower packet uncorrectable probability for HRR packets has demonstrated that the proposed PIS can successfully protect HRR packets from burst errors, while at the same time it does not sacrifice non-HRR packets from large uncorrectable bit errors.

6. Conclusion

In this paper, we have presented a packet interleaving scheme (PIS) using RS code to reduce packet uncorrectable probability under burst errors in wireless sensor networks. In PIS, the collected data are classified into HRR and non-HRR. An HRR packet is encoded with a shorter RS symbol, while a non-HRR packet with a longer RS symbol. One of the contributions of this paper is right in that an HRR and a non-HRR packet are interleaved on a symbol-by-symbol basis, which effectively disperses the burst errors among an interleaved packet. Consequently, the packet uncorrectable probability of an HRR packet can be significantly reduced. In the performance evaluation, we built two mathematical models, UBEM and OBEM. From the thorough numerical simulations, we have demonstrated that, no matter how we adjust the period of high bit errors, the proposed PIS behaves more resilient to burst errors in OBEM than in UBEM. Finally, we have revealed that, by carefully adjusting the period of high bit errors and the right-shift positions, the proposed PIS can reduce the uncorrectable probability of HRR packets to near zero.

Acknowledgements

The authors would like to thank the reviewers, whose valuable comments helped improve the presentation of this paper. Additionally, the authors would like to thank Prof. Tsang-Ling Sheu's assistant, Miss Yi-Ying Ke for her cautious redrawing of the figures which helped improve the quality of the paper. This study is supported under the three grant numbers: (1) NSC 102-2221-E-110-036-MY2 of MoST, Taiwan, (2) 103-EC-17-A-03-S1-214, of MOEA, Taiwan, and (3) NHRI-EX103-10142EI, Taiwan.

References

- [1] Barac, F., Yu, K., Gidlund, M., Akerberg, J. and Bjrkman, M. (2012) Towards Reliable and Lightweight Communica-

- tion in Industrial Wireless Sensor Networks. *Proceedings of the INDIN 2012: IEEE 10th International Conference on Industrial Informatics*, Beijing, 25-27 July 2012, 1218-1224. <http://dx.doi.org/10.1109/indin.2012.6300846>
- [2] Marinkovic, S. and Popovici, E. (2009) Network Coding for Efficient Error Recovery in Wireless Sensor Networks for Medical Applications. *Proceedings of the 1st International Emerging Network Intelligence*, Sliema, 11-16 October 2009, 15-20. <http://dx.doi.org/10.1109/emerging.2009.22>
- [3] Sklar, B. (2001) *Digital Communications: Fundamentals and Applications*. 2nd Edition, Prentice Hall, Upper Saddle River.
- [4] Chebbo, H., Abedi, S., Lamahewa, T.A., Smith, D.B., Miniutti, D. and Hanlen, L. (2010) Reliable Body Area Networks Using Relays: Restricted Tree Topology. *Proceedings of the 2012 International Conference on Computing, Networking and Communications (ICNC)*, Maui, 30 January-2 February 2010, 82-88. <http://dx.doi.org/10.1109/icnc.2012.6167540>
- [5] Sampangi, R.V., Urs, S.R. and Sampalli, S. (2011) A Novel Reliability Scheme Employing Multiple Sink Nodes for Wireless Body Area Networks. *Proceedings of the 2011 IEEE Symposium on Wireless Technology and Applications (ISWTA)*, Langkawi, 25-28 September 2011, 162-167. <http://dx.doi.org/10.1109/iswta.2011.6089401>
- [6] Kim, S., Fonseca, R. and Culler, D. (2004) Reliable Transfer on Wireless Sensor Network. *Proceedings of the 1st Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, Santa Clara, 4-7 October 2004, 449-459.
- [7] Byrne, E., Manada, A., Marinkovic, S. and Popovici, E. (2011) A Graph Theoretical Approach for Network Coding in Wireless Body Area Networks. *Proceedings of the 2011 IEEE International Symposium on Information Theory Proceedings (ISIT)*, Saint Petersburg, 31 July-5 August 2011. <http://dx.doi.org/10.1109/isit.2011.6034156>
- [8] Hamada, Y., Takizawa, K. and Ikegami, T. (2012) Highly Reliable Wireless Body Area Network Using Error Correcting Codes. *Proceedings of the 2012 IEEE Radio and Wireless Symposium*, Santa Clara, 15-18 January 2012, 231-234. <http://dx.doi.org/10.1109/rws.2012.6175351>
- [9] Luby, M. (2002) LT Codes. *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, Vancouver, 16-19 November 2002, 271-280. <http://dx.doi.org/10.1109/sfcs.2002.1181950>
- [10] Ishibashi, K., Ochiai, H. and Kohno, R. (2008) Embedded Forward Error Control Technique (EFFECT) for Low-Rate but Low Latency Communications. *IEEE Transactions on Wireless Communications*, **7**, 1456-1460. <http://dx.doi.org/10.1109/TWC.2008.060573>
- [11] Busse, M., Haenselmann, T. and Effelsberg, W. (2007) Energy-Efficient Data Dissemination for Wireless Sensor Networks. *Proceedings of the 5th Annual IEEE International Conference on Pervasive Computing and Communications Workshops*, White Plains, 19-23 March 2007. <http://dx.doi.org/10.1109/percomw.2007.44>
- [12] Yu, K., Barac, F., Gidlund, M., Akerberg, J. and Bjorkman, M. (2012) A Flexible Error Correction Scheme for IEEE 802.15.4-Based Industrial Wireless Sensor Networks. *Proceedings of the 2012 IEEE International Symposium on Industrial Electronics (ISIE)*, Hangzhou, 28-31 May 2012, 1172-1177. <http://dx.doi.org/10.1109/isie.2012.6237255>
- [13] Srouji, M.S., Wang, Z. and Henkel, J. (2011) RDTs: A Reliable Erasure-Coding Based Data Transfer Scheme for Wireless Sensor Networks. *Proceedings of the 2011 IEEE 17th International Conference on Parallel and Distributed Systems*, Tainan, 7-9 December 2011, 481-488. <http://dx.doi.org/10.1109/icpads.2011.104>
- [14] Arrobo, G.E. and Gitlin, R.D. (2011) Improving the Reliability of Wireless Body Area Networks. *Proceedings of the 33rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Boston, 30 August-3 September 2011, 2192-2195. <http://dx.doi.org/10.1109/iembs.2011.6090413>
- [15] Taparugssanagorn, A., Ono, F. and Kohno, R. (2010) Network Coding for Non-Invasive Wireless Body Area Networks. *Proceedings of the 2010 IEEE 21st International Symposium on Personal, Indoor and Mobile Radio Communications Workshops*, Istanbul, 26-30 September 2010, 134-138. <http://dx.doi.org/10.1109/pimrcw.2010.5670413>
- [16] Salhi, I., Ghamri-Doudane, Y., Lohier, S. and Roussel, G. (2011) Reliable Network Coding for ZigBee Wireless Sensor Networks. *Proceedings of the 8th IEEE International Conference on Mobile Ad-Hoc and Sensor Systems*, Valencia, 17-22 October 2011, 135-137.
- [17] Kiss, Z.I., Polgar, Z.A., Stef, M.P. and Bota, V. (2012) Network Coding Solution for Improving Transmission Reliability in Wireless Sensor Networks Employed in Industrial Monitoring. *Proceedings of the 35th International Conference on Telecommunications and Signal Processing (TSP)*, Prague, 3-4 July 2012, 190-195. <http://dx.doi.org/10.1109/tsp.2012.6256280>
- [18] Reed, I.S. and Solomon, G. (1960) Polynomial Codes over Certain Finite Fields. *Journal of the Society of Industrial and Applied Mathematics*, **8**, 300-304. <http://dx.doi.org/10.1137/0108018>
- [19] Lidl, R. and Niederreiter, H. (1997) *Finite Fields*. Cambridge University Press.