Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2014, Article ID 698125, 9 pages
http://dx.doi.org/10.1155/2014/698125

Research Article

Energy Efficient and High Speed Error Control Scheme for Real Time Wireless Sensor Networks

Ali Barati,[1] Ali Movaghar,[2] and Masoud Sabaei[3]

[1] *Department of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran*
[2] *Department of Computer Engineering, Sharif University of Technology, Tehran, Iran*
[3] *Computer Engineering and Information Technology Department, Amirkabir University of Technology, Tehran, Iran*

Correspondence should be addressed to Ali Barati; abarati@iaud.ac.ir

Received 5 October 2013; Revised 24 January 2014; Accepted 2 February 2014; Published 26 May 2014

Academic Editor: Yong Jin

Reliability and energy consumption are two of the main constraints in wireless sensor networks (WSNs). In this paper, a novel energy efficient and high speed error control scheme is introduced that is based on the Redundant Residue Number System (RRNS) allowing real-time application of WSNs. The proposed approach employs a new 3-moduli set $\{2^{2n+1}, 2^{2n+1}-1, 2^n-1\}$ and an efficient reverse converter which relies on the Mixed Radix Conversion (MRC) algorithm and achieves significant improvements both in terms of conversion delay and hardware design. In order to obtain error controllability, two-redundant-moduli set $\{2^{3n+1}-1, 2^{4n+1}-1\}$ is also added to the main 3-moduli set. The theoretical results backed by simulation tests confirm that the solution put forward in this paper outperforms popular error control methods for WSNs in terms of error controllability, energy efficiency, and reduction of end-to-end delay.

1. Introduction

Wireless sensor networks with a large number of multifunctional low power sensor nodes have a broad range of applications, such as battlefield surveillance, environmental monitoring, disaster relief, and healthcare [1–5]. The main task of wireless sensor networks is to collect the sensed data from environment and send them back to the sink node for further processing. Because of the dynamic, lossy, and low-power nature of sensor networks, two of the most important constraints that designers have to overcome in these networks are energy consumption and reliability [6]. Moreover, high speed operation is another critical issue in real time wireless sensor networks applications. In-network data aggregation which is proposed for energy conservation [7] and error control schemes such as Automatic Repeat reQuest (ARQ) and Forward Error Correction (FEC) is used to improve reliability. In ARQ-based schemes, the receiver must detect lost packets and then request the sender to retransmit packets [8]. In FEC codes, to transmit data from sensor to the sink node, each aggregator must decode the received data, aggregate them with new data, and encrypt them again before sending to the sink node. These operations cause a significant end-to-end delay and high energy consumption, which leads to a decrease in the network lifetime. Therefore, it is necessary to develop a new solution that offers both low power consumption and guaranteed reliability as well as low end-to-end delay in real-time wireless sensor networks.

In this paper, authors utilize RRNS to design a reliable, energy-efficient, and high speed algorithm for real time applications of wireless sensor networks. The proposed RRNS-based model offers desirable features including robust security, an enhanced error detection, and correction capability. Furthermore, the solution allows parallel operations and thus making real-time functionality as well as the use of low-power components possible. By employing a 3-moduli set $\{2^{2n+1}, 2^{2n+1}-1, 2^n-1\}$ and exploiting the MRC algorithm, an efficient reverse converter is designed. Two redundant-moduli set $\{2^{3n+1}-1, 2^{4n+1}-1\}$ is subsequently added to the mentioned three main RNS moduli to create a 5-moduli RRNS set $\{2^{2n+1}, 2^{2n+1}-1, 2^n-1, 2^{3n+1}-1, 2^{4n+1}-1\}$. This

inclusion provides the new system with an effective error control capability.

The rest of the paper is organized as follows: a few of related works are discussed in Section 2. In Section 3 a full description of the proposed scheme is presented. This is followed in Section 4 by a detailed explanation of an efficient reverse converter design and derivation of the necessary expressions. In Section 5, error controllability of the proposed scheme is evaluated through simulation. Finally, the paper is concluded in Section 6.

2. Related Works

As was discussed in the introduction section, error control methods in WSNs are divided into two general categories, namely, Automatic Repeat reQuest (ARQ) and Forward Error Correction (FEC). The ARQ method is used only for error detection and once a corrupted data is sensed, a request for information retransmission is made. This approach is based on Cyclic Redundancy Check (CRC). To set specifications for a CRC code it requires a generator of polynomial to be defined. This polynomial resembles the divisor in a polynomial long division, which takes the message as the dividend and in which the quotient is discarded and the remainder becomes the result, with the important distinction that the polynomial coefficients are calculated according to the carry-less arithmetic of a finite field. In order to make checksum, attach a polynomial generator to the packet before it is transmitted. At the receiver, if the division of the bit stream by $G(x)$ generates no remainder, then this is proof that transmission has been free of error. For each error-free packet received a positive acknowledgment (ACK) is sent back by the receiver and for each corrupted packet received a negative acknowledgment (NACK) is sent back by the receiver [9, 10].

By adding redundancy to the sent data, the FEC codes let the receiver detect the errors and correct them by itself. One of these codes is BCH which is represented by (n, k, t), where "n" refers to the length of the total sent bits, "k" denotes the length of the main data, and "t" stands for the maximum number of the errors that could be corrected.

In the age of modern wireless communications, the need for end-to-end reliable data transfer is growing incessantly. Fortunately, to ensure error free transmission, a variety of popular error control techniques devoted to error detection and correction have been published in the literature. In what follows, a number of these reported methods are examined to ascertain how they achieve their objectives. In [6, 11] for example, we note that error control schemes in Bluetooth sensor networks were explored, that is, a low cost wireless technology designed to facilitate the formation of ad hoc networks. In [11], authors studied an energy efficient model for adaptive and custom error control scheme and also investigated energy consumption and reliability constraints of WSNs. For custom coding, they introduced new packet types using some error control strategies. These new packets exploited CRC for error detection (without ARQ), BCH code with and without CRC, and the Hamming code with and without CRC. They also unveiled two adaptive techniques and a packet selection strategy that was based on channel state. The result of their analysis showed that error recovery was energy efficient provided that the channel conditions were known and a suitable control scheme was employed. This meant that with good channel conditions, there was no need to vary energy consumption and the use of packets with little or no error protection resulted in an efficient scenario. However, for low values of signal to noise ratio (SNR), the BCH packet offered the most effective tool due to its ability to correct more errors, in spite of higher energy consumption. Thus, under the circumstances where the error occurrence probability was high, a more robust error recovery strategy had to be employed for the network.

Notice that the results obtained in [11] were only applicable to networks, where designers had access to information on the channel conditions. Moreover, power consumption of the WSN had to be taken into account. It is clear that in situations where channel conditions are very vulnerable a robust error control scheme is the best choice. But the feasibility of this choice must be considered against energy constraints of the given WSN. Another work that also focused on reliability in Bluetooth sensor networks was presented by Khodadoustan et al. [6] which included an attempt to establish reliability/energy tradeoff in Bluetooth error control scheme. The paper investigated a combination of ARQ and a particular CRC code known as CRC-CCITT and also carried out an analysis of the data packets by using different BCH codes. Furthermore, by conducting simulations, authors examined the proposed coding techniques' performance as a result of changes in the number of hops and their effect on SNR. The results of this study can help network designers to decide on suitable packet types and error control schemes. Turning to [8, 20], we find that researchers' main interests were in reliability for underwater wireless sensor networks (UWSNs). Sensors in underwater wireless sensor networks are often mobile and the channels are acoustic. Such WSNs have low available bandwidth, large propagation delays, high error probability, and highly dynamic network topology [20]. Thus, error recovery is a major challenge in underwater wireless sensor networks.

Guo et al. in [8] suggested the use of network coding in multipath for UWSNs to achieve efficient error recovery in the presence of high error probability. An analytical evaluation of the performance of this scheme against several other error-recovery systems revealed that this method was very efficient in terms of error recovery and energy consumption.

Segmented Data Reliable Transfer protocol (SDRT) was another solution that authors reported for UWSNs in [20]. SDRT was a hybrid approach based on ARQ and FEC techniques which adopted efficient erasure codes whereby the packets were transmitted block-by-block and hop-by-hop. Moreover, a mathematical model was subsequently put forward which achieved further energy saving and enabled designers to set an appropriate size for the block for each packet. By using this model, the number of needed packets could be accurately predicted.

Iyer et al. in [21] offered STCP protocol which is a reliable end-to-end transport protocol. This protocol provides different levels of reliability in accordance with application; that is,

it provides two levels of reliability for data flows for event-driven and continuous data flow applications. When packets are transmitted, for event-driven applications and continuous data flow applications, ACK packets and NACK packets are used to indicate that the packets have reached the destination, respectively. Before transporting packets over STCP, sensor nodes transmit a Session Initiation packet to the base station. This Session Initiation packet sends data such as the type of data flow, transmission rate, and number of flows to the base station. The base station then stores this packet and sets timer and other parameters of this data flow. When data is received, the base station sends a ACK message to make connection. When sensor node receives ACK, it starts to transmit data. The base station transmits ACK or NACK message, based on the type of the flow. STCP transfers packets according to the size of the window. Also, this method has a congestion detection mechanism and employs occupancy monitoring of the queue of intermediate nodes. The assumption in this paper is that all network intermediate nodes have clocks which are synchronized by sink.

Marchi et al. in [22] suggested a reliable transport protocol called DTSN. This method provides two levels of reliability and is based on full reliability and differential reliability. If it is required that all packets are received at the destination accurately, full reliability is used; otherwise differential reliability is employed. DTSN relies on Selective Repeat ARQ and uses ACK and NACK to provide full reliability. Packets transmit data packets to destination in accordance with the size of Acknowledgment Window (AW). Then Explicit Acknowledgement Request is transmitted to destination. If the destination has received all the packets, ACK packet will be sent; otherwise NACK packet will be sent. DTSN protocol adds Forward Error Correction (FEC) mechanism to provide differential reliability. In this method, a session is a source/destination relationship univocally identified by the tuple ⟨source address, destination address, application identifier, session number⟩ designated the session identifier. The number of the session is selected randomly and the packets of a session are numbered continuously. Intermediate nodes store packets in their buffers and then transmit them, so they are able to transmit them repeatedly if necessary. However, when a large volume of data is transmitted, a large memory is needed, which is a problem.

In all of the aforesaid schemes, there is the need for data decryption and encryption during the aggregation phase using secret key. Thus, these algorithms are very energy and time deficient. Moreover, the security and confidentiality of data are in great peril because of decryption of the received packet in each aggregator, aggregation of the recovered message with the new data and, finally, encryption of the new data by the aggregator that is to be sent to the sink. In the proposed scheme, security is preserved in each aggregator and there is no need to decrypt and then encrypt the data again in each aggregator, which results in notable conservation of energy and reduction of transmission time.

In addition to these, none of the above schemes considered the "delay" parameter. Although WSNs are immensely popular but for real-time applications, end-to-end delay becomes a critical design issue. In such circumstances, if delay turns out to be more than a fixed value, the results and information will be useless. It is clear, therefore, that methods presented in [1, 6, 8, 20] could not be adopted effectively and again the technique explored in this paper is the one that meets the requirement for high speed operations.

3. The Proposed Scheme

Minimizing energy dissipation and maximizing network lifetime are among the central concerns when designing applications and protocols for sensor networks. Clustering has been proven to be energy efficient in sensor networks since data routing and relaying are only operated by cluster heads. Besides, cluster heads can process, filter, and aggregate data sent by cluster members, and thus reducing network load and alleviating the bandwidth.

Residue number system is a nonweighted system that employs residue of numbers divided by several specific moduli to represent them [23–25]. Residue number system can be used for computational applications that need real-time processing such as digital signal processing [26], digital filtering [27], image processing [28], RSA encryption algorithm [29], and digital communications [30].

A residue number system is defined in terms of relatively prime modulus set $\{P_1, P_2, \ldots, P_n\}$ whose $\gcd(P_i, P_j) = 1$ for $(i \neq j)$. In this system a weighted number X can be represented as $(x_1, x_2, \ldots, x_n)$, where

$$x_i = X \mod P_i = |X|_{P_i}, \quad 0 \leq x_i < P_i. \tag{1}$$

Such a representation is unique for any integer X in the range $[0, \mathrm{M})$, where $M = \prod_{i=1}^{n} P_i$ is the dynamic range of the modulus set $\{P_1, P_2, \ldots, P_n\}$ [31].

Redundant Residue Number system is specified by moduli set $\{m_1, m_2, m_3, \ldots, m_h, m_{h+1}, \ldots, m_{h+r}\}$ and $m_i > m_{i-1}$ for $i = 2, 3, \ldots, h + r$. If all the moduli are set pair wise prime, the dynamic range of the system is $[0, M = \prod_{i=1}^{h+r} m_i)$.

In the Redundant Residue Number system with $h + r$ modulo, X where $\alpha \leq X < \alpha + M$ with $h + r$ residue, is represented as $(x_1, x_2, \ldots, x_h, x_{h+1}, \ldots, x_{h+r})$ [32–34].

Having studied a variety of literature on the issue of error control in WSNs and the challenges it poses, the proposed scheme has come up with a solution that is based on Redundant Residue Number System (RRNS). The algorithm offers many advantages and overcomes the weakness seen in other approaches. Here, each sensor in a cluster computes remainders of the sensed data using a predefined modulus set and sends the result, that is, data which is a smaller number compared to original data, to the cluster head where the received numbers are then aggregated. It is clear that since aggregation is being performed on remainders rather than on the entire sensed data, it allows for more compact processing units to be utilized at reduced power consumption in cluster heads and also given that all aggregation operations take place in parallel; this has the benefit of increasing the speed noticeably. It should also be noted that unlike other methods studied, where preaggregation decoding and postencoding are required, in the proposed scheme these processes are unnecessary and only cluster head aggregation operation is

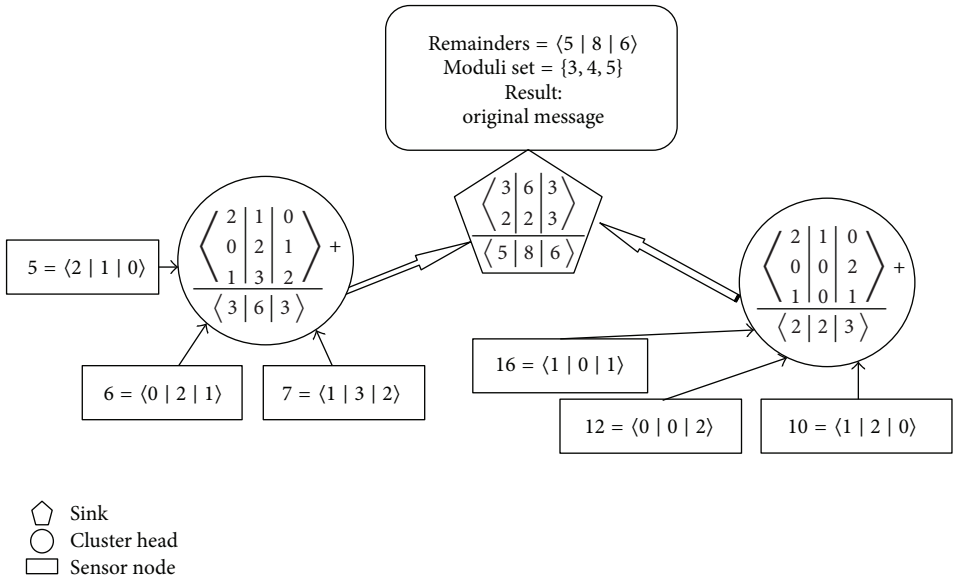

Figure 1: An example of data aggregation functionality using the proposed scheme.

required. It is therefore seen that this solution results in a significant drop in end-to-end delay which makes it suitable for real-time wireless sensor network applications.

The proposed method for error control could be applied to tree-like network topologies and cluster based network topologies. In tree-like network, children transmit the residue of their sensed data, that has been divided by the moduli set, to their parent and their parent will aggregate the received residues and his own residues. Next, the parent transmits the result to his own parent. This will continue until the residues reach sink. In a cluster-based network, the members of the cluster transmit the residue of their sensed data divided by moduli set to head cluster. Then the head cluster aggregates the received residues and its own residues and transmits the result to the next head clusters.

In this study, we considered 5 moduli, 3 main and 2 redundant. Since wireless sensor networks are limited in terms of energy, processing power, size, memory,. . .., selection of the number and value of moduli was made in a way that their reverse converters led to the lowest energy consumption and delay. How we select the main moduli determines the value of dynamic range (obtained by product of main moduli). In fact, if we increase the dynamic range, it will allow us to represent larger numbers using the proposed method. On the other hand, increasing the number of redundant moduli results in higher capability of error detection and correction. Thus, if t redundant modulo is considered, t modulo containing error can be detected and $\lfloor t/2 \rfloor$ residue can be corrected. However, the increase in the number of the moduli leads to an increase in reverse converter cost.

An example illustrating the basic theme of the proposed scheme is shown below, consisting of six sensors located in two clusters. Each cluster has three sensors and a cluster head. In this example, aggregation is a summation operation where the remainders of 5 based on the moduli $\{3, 4, 5\}$ are $\{2, 1, 0\}$ and the remainders of other sensed data are calculated in the same way. In the aggregators, where summation takes place, all of the results received from the first modulo are added together also as all of the results from the second modulo and so on for the third modulo. This feature offers the added advantage of requiring lower space for the arithmetic block. As can be seen, in the sink node, the final decrypted remainders are $\{5, 8, 6\}$ and the modulus set is $\{3, 4, 5\}$.

Now, in order to achieve a lower power consumption and as compact a design as possible, a new moduli set $\{2^{2n+1}, 2^{2n+1} - 1, 2^n - 1\}$ replaces $\{3, 4, 5\}$ in Figure 1. Moreover, the sink node using a reverse converter can decode the message and recover the original data.

Hence an efficient reverse converter is needed. There are two basic approaches that convert a number from RNS to its binary equivalent. These are the Chinese Reminder Theorem (CRT) and Mixed Radix Conversion (MRC) [6], with MRC algorithm being the preferred choice in this paper. Considering also that reliability in data delivery is an important issue in almost all wireless sensor network applications, two-redundant-moduli set $\{2^{3n+1} - 1, 2^{4n+1} - 1\}$ is added to the new 3-moduli set $\{2^{2n+1}, 2^{2n+1} - 1, 2^n - 1\}$, to further enhance error control capability of the proposed scheme. It should be pointed out that the proposed moduli set acts as secret key which is used by the sink node to decrypt the received remainders and obtain the original message. If an adversary that has no knowledge of the moduli set could acquire the transmitted packet, it would not be able to decrypt the message. It is clear that redundant residue number allows transmit data to be encrypted by employing symmetric key and thus provides a powerful tool for error bit detection as well as an effective means for data correction.

4. Design and Implementation of Proposed Residue to Binary Converter

Various methods can be used to design a reverse converter. The most popular methods are MCR and CRT. The use of CRT or MRC depends on selected moduli since multiplicative inverse should be computed in both methods. Computation of multiplicative inverse is the most difficult part of reverse converter. Therefore, it is important to look for multiplicative inverses that allow the formulation of the most simple equations. Certain moduli set can be designed well and at less cost by employing MCR since their multiplicative inverse is computed more easily through simplified (less complex) equations. As a result, the volume of required hardware and consumption of energy as well as network delay are all reduced. In this study we applied MRC. Using this method the values of multiplicative inverse become so simpler, and the design of the reverse converter costs so much less.

The residue to binary conversion can be performed using the MRC algorithm as follows:

$$X = V_n \prod_{i=1}^{n} P_i + \cdots + V_3 P_2 P_1 + V_2 P_1 + V_1. \tag{2}$$

The coefficients $V_i P$ can be obtained from residues by

$$\begin{aligned} V_1 &= x_1, \\ V_2 &= \left| (x_2 - x_1) \left| P_1^{-1} \right|_{P_2} \right|_{P_2}, \end{aligned} \tag{3}$$

$$V_3 = \left| \left((x_3 - x_1) \left| P_1^{-1} \right|_{P_3} - V_2 \right) \left| P_2^{-1} \right|_{P_3} \right|_{P_3}. \tag{4}$$

In the general case, we have

$$\begin{aligned} V_n = {} & \left(\left((x_n - V_1) \left| P_1^{-1} \right|_{P_n} - V_2 \right) \left| P_2^{-1} \right|_{P_n} \right. \\ & \left. - \cdots - V_{n-1} \right) \left| P_{n-1}^{-1} \right|_{P_n} \Big|_{P_n}, \end{aligned} \tag{5}$$

where $|P_i^{-1}|_{P_j}$ denotes the multiplicative inverse of P_i modulo P_j. In mathematics, a multiplicative inverse for a number X is a number that when multiplied by X yields the multiplicative identity, 1. The modular multiplicative inverse of a modulo m can be derived from the extended Euclidean algorithm.

According to (2)–(5), the proposed reverse converter can be designed for the new 3-moduli set $\{2^{2n+1}, 2^{2n+1} - 1, 2^n - 1\}$ as follows.

Consider the 3-moduli set $\{P_1, P_2, P_3\} = \{2^{2n+1}, 2^{2n+1} - 1, 2^n - 1\}$ with three corresponding residues (x_1, x_2, x_3). In order to design a residue to binary converter, we first need to obtain the multiplicative inverse values and substitute these values with the modulus set in the conversion algorithm formulas. The resultant equations should then be simplified by using the arithmetic properties. The final conversion stage involves the implementation of these simplified equations using hardware components such as full adders and logic gates. The following propositions are used to obtain the closed form expressions, which will be the means to compute the multiplicative inverses based on the MRC algorithm.

Proposition 1. *The multiplicative inverse of* (2^{2n+1}) *modulo* $(2^{2n+1} - 1)$ *is* $k_1 = 1$.

Proof. One has

$$\left| 2^{2n+1} \right|_{2^{2n+1}-1} = 1. \tag{6}$$

□

Proposition 2. *The multiplicative inverse of* (2^{2n+1}) *modulo* $(2^n - 1)$ *is* $k_2 = 2^{n-1}$.

Proof. One has

$$\left| 2^{n-1} \times 2^{2n+1} \right|_{2^n-1} = \left| 2^{3n} \right|_{2^n-1} = 1. \tag{7}$$

□

Proposition 3. *The multiplicative inverse of* $(2^{2n+1} - 1)$ *modulo* $(2^n - 1)$ *is* $k_3 = 1$.

Proof. One has

$$\left| 2^{2n+1} - 1 \right|_{2^n-1} = 1. \tag{8}$$

Therefore, let the values $< k_1 = 1$, $k_2 = 2^{n-1}$, $k_3 = 1$, $P_1 = (2^{2n+1})$, $P_2 = (2^{2n+1} - 1)$, and $P_3 = (2^n - 1) >$ in (2)–(4) and thus we have

$$\begin{aligned} X &= x_1 + P_1 (V_2 + V_3 P_2) \\ &= x_1 + \left(2^{2n+1}\right) \left(V_2 + \left(2^{2n+1} - 1\right) V_3\right), \end{aligned} \tag{9}$$

$$V_1 = x_1, \tag{10}$$

$$V_2 = \left| (x_2 - x_1) \left| P_1^{-1} \right|_{P_2} \right|_{P_2} = \left| x_2 - x_1 \right|_{2^{2n+1}-1}, \tag{11}$$

$$\begin{aligned} V_3 &= \left| \left((x_3 - x_1) \left| P_1^{-1} \right|_{P_3} - V_2 \right) \left| P_2^{-1} \right|_{P_3} \right|_{P_3} \\ &= \left| 2^{n-1} \times ((x_3 - x_1) + (-V_2)) \right|_{2^n-1}. \end{aligned} \tag{12}$$

In order to design an efficient reverse converter, expressions (9) and (11)-(12) can be rewritten as follows:

$$\begin{aligned} V_2 &= \left| (x_2 - x_1) \left| P_1^{-1} \right|_{P_2} \right|_{P_2} = \left| x_2 - x_1 \right|_{2^{2n+1}-1} \\ &= \left| x_2 \right|_{2^{2n+1}-1} + \left| -x_1 \right|_{2^{2n+1}-1} = V_{21} + V_{22}, \end{aligned} \tag{13}$$

where

$$\begin{aligned} V_{21} &= \left| x_2 \right|_{2^{2n+1}-1} = \underbrace{x_{2,2n}\, x_{2,2n-1} \cdots x_{2,0}}_{(2n+1)\text{bits}}, \\ V_{22} &= \left| -x_1 \right|_{2^{2n+1}-1} = \underbrace{\bar{x}_{1,2n}\, \bar{x}_{1,2n-1} \cdots \bar{x}_{1,0}}_{(2n+1)\text{bits}}. \end{aligned} \tag{14}$$

Now, in order to implement V_3 on the basis of (12), we have

$$\begin{aligned} V_3 &= \left| \left((x_3 - x_1) \left| P_1^{-1} \right|_{P_3} - V_2 \right) \left| P_2^{-1} \right|_{P_3} \right|_{P_3} \\ &= \left| 2^{n-1} \times (x_3 - x_1) - V_2 \right|_{2^n-1} \\ &= V_{31} + V_{32} + V_{33}, \end{aligned} \tag{15}$$

where

$$V_{31} = \left|2^{n-1} \times x_3\right|_{2^n-1} = \underbrace{x_{3,0}}_{1\text{bit}}\ \underbrace{x_{3,n-1} \cdots x_{3,1}}_{(n-1)\text{bits}}$$

$$V_{32} = \left|-2^{n-1} \times x_1\right|_{2^n-1} = \left\{ \begin{array}{c} \bar{x}_{1,0}\ \bar{x}_{1,n-1} \cdots \bar{x}_{1,1} \\ \bar{x}_{1,n}\ \bar{x}_{1,2n-1} \cdots \bar{x}_{1,n+1} + \\ \underbrace{\bar{x}_{1,2n}}_{1\text{bit}}\ \underbrace{1 \cdots 1 \cdots 1 \cdots 1}_{(n-1)\text{bits}} \end{array} \right\} \quad (16)$$

$$V_{33} = \left|-V_2\right|_{2^n-1} = \left\{ \begin{array}{c} \bar{V}_{2,n-1} \cdots \bar{V}_{2,1} \bar{V}_{2,0} \\ \bar{V}_{2,2n-1} \cdots \bar{V}_{2,n+1} \bar{V}_{2,n} + \\ \underbrace{1 \cdots 1 \cdots 1 \cdots 1}_{(n-1)\text{bits}} \underbrace{\bar{V}_{2,2n}}_{1\text{bit}} \end{array} \right\}.$$

Finally, to obtain X based on (9), we have

$$\begin{aligned} X &= x_1 + P_1 \left(V_2 + V_3 P_2\right) \\ &= x_1 + \left(2^{2n+1}\right) \underbrace{\left(V_2 + \left(2^{2n+1} - 1\right) V_3\right)}_{C} \\ &= x_1 + \left(2^{2n+1}\right) C \\ C &= V_2 + \left(2^{2n+1} - 1\right) V_3 \end{aligned}$$

$$C = \left\{ \begin{array}{cc} \overbrace{V_{3,n-1} \cdots V_{3,0}}^{n\text{bits}} & \overbrace{V_{2,2n} V_{2,2n-1} \cdots V_{2,n}}^{(n+1)\text{bits}} \quad \overbrace{\bar{V}_{3,n-1} \cdots \bar{V}_{3,0}}^{n\text{bits}} + \\ \underbrace{1 \cdots 1 \cdots 1 \cdots 1 \cdots 1 \cdots 1 \cdots 1 \cdots 1}_{(2n+1)\text{bits}} & \underbrace{V_{2,n-1} \cdots V_{2,0}}_{n\text{bits}} \end{array} \right\}$$

$$X = x_1 + \left(2^{2n+1}\right) C = \text{Concatenation}\left(x_1, C\right). \quad (17)$$

□

The area and delay specifications for the proposed reverse converter are shown in Table 1.

From this Table, we have

$$\begin{aligned} \text{Total area} &= (7n+3)\, A_{\text{FA}} + (5n+2)\, A_{\text{NOT}} \\ &\quad + (4n-1)\, A_{\text{OR}} + (4n-1)\, A_{\text{XNOR}} \quad (18) \\ \text{Total delay} &= (9n+8)\, t_{\text{FA}} + 3t_{\text{NOT}}. \end{aligned}$$

Now let us discuss comparison of the proposed reverse converter's performance for the new moduli set $\{2^{2n+1}, 2^{2n+1} - 1, 2^n - 1\}$ against competing models that have either the same or a lower dynamic range in terms of hardware requirement and speed of operations. It should be noted that in any reverse converter most of the delay is associated with full adders which also occupy the largest area allocated to components assembly. Thus for the sake of comparison it is sufficient to investigate only A_{FA} and t_{FA}; see Table 2.

5. Performance Evaluation

In this section, the performance of the proposed moduli set $\{2^{2n+1}, 2^{2n+1}-1, 2^n-1, 2^{3n+1}-1, 2^{4n+1}-1\}$ is evaluated in terms of error detection and error correction capabilities. Evaluation of the error control capability requires an investigation of the effectiveness of the employed "error correction" approach. In general, an error control technique is divided into two parts "error detection" and "error detection and correction". For simplicity's sake, here we refer to "error correction" to mean both error detection and correction noting that error bits must first be detected and located before they can be corrected. Referring to the residue number system, one of its features is that it can correct "burst errors."

This means that if error bits are residing in one modulo, the error control algorithm can easily detect and correct them. However, when error bits are located in more than one modulo, they cannot be corrected although it may be possible to detect them. So, the aim now is to discover what the average percentage of error detection is when error bits reside in more than one modulo by employing the proposed RNS based solution.

In the proposed moduli set, $\{2^{2n+1}, 2^{2n+1}-1, 2^n-1, 2^{3n+1}-1, 2^{4n+1}-1\}$, $\{2^{2n+1}, 2^{2n+1}-1, 2^n-1\}$ are the main three and $\{2^{3n+1}-1, 2^{4n+1}-1\}$ are the redundant moduli, respectively. Now, the first, second, and third main moduli are $(2n+1)$, $(2n+1)$, and (n) bits, respectively. Given this architecture, the error bits can only be detected and corrected if (a) they are

Table 1: Characterization of each part of the proposed reverse converter.

Parts	FA	NOT	AND/XOR	OR/XNOR	Delay
OPU 1	—	$(2n+1)$	—	—	t_{NOT}
CPA 1	$(2n+1)$	—	—	—	$(4n+2)t_{\text{FA}}$
CSA 1	N	—	—	—	t_{FA}
CSA 2	1	—	—	$(n-1)$	t_{FA}
OPU 2	—	$(2n+1)$	—	—	t_{NOT}
CSA 3	N	—	—	—	t_{FA}
CSA 4	N	—	—	—	t_{FA}
CSA 5	1	—	—	$(n-1)$	t_{FA}
CPA 2	N	—	—	—	$(2n)t_{\text{FA}}$
OPU 3	—	N	—	—	t_{NOT}
R– CPA	N	—	—	$(2n+1)$	$(3n+1)t_{\text{FA}}$

Table 2: Area and Delay comparison.

Reverse converter	Area (A_{FA})	Delay (t_{FA})
[12]	$8n+2$	$12n+5$
[13]	$(5n^2+43n)/6+\ 16n-1$	$18n+7$
[14]	$10n+5$	$13n+1$
[15]	$12.5n+6$	$12n+6$
[16]	$10n+5$	$12n+1$
[17]-1	$9n+5$	$11.5n+6$
[17]-2	$n^2+12n+12$	$16n+22$
[17]-3	$9n+10$	$11n+14$
[18]	$n^2/2+\ 11n+14$	$11n+8$
[19]-CICE	$2.5n^2+25.5n+12$	$18n+23$
[19]-CIHS	$2.5n^2+\ 37.5n+28$	$12n+15$
[19]-C2CE	$20n+17$	$13n+22$
[19]-C3CE	$23n+11$	$16n+14$
Proposed	$7n+3$	$9n+8$

Table 3: Area and delay comparison.

N	Main modulus	Redundant Modulus	Maximum error correction	Error detection capability	
				Error bits in one modulo	Error bits in two modulus
2	{32, 31, 3}	{127, 511}	5 bits	100%	99.37%
3	{128, 127, 7}	{1023, 8191}	7 bits	100%	100%
4	{512, 511, 15}	{8191, 131071}	9 bits	100%	100%

placed either in the first, the second, or the third modulo (b); maximum error bit count does not exceed $(2n+1)$, $(2n+1)$, and (n), respectively. Hence, the maximum achievable error correction capability is equal to $(2n+1)$ error bits occurring in the first or the second modulo. On the other hand, the minimum correction capability that can be achieved using the proposed modulus set is equal to one error bit within any modulo of the received packet.

In the proposed error control method, first sensed data are divided by the proposed moduli set and then the residues are computed. Next, the nodes transmit the residues instead of sensed data. After receiving residues, sink computes the original data using 3 main moduli and their received residues. When the residue of the computed number has been divided by redundant moduli and the received residues for those redundant moduli are not equal, an error is detected and error correction procedure is called for it. Sink can reconstruct the original data using 3 residues out of the 5 residues and MRC reverse converter. There are only 10 possible conditions for selecting 3 residues out of 5 residues ($C_3^5 = \binom{5}{3} = (5!/(2! * 3!)) = 10$). Sink computes the original data for all of these possible conditions. Given the fact that during data transfer some residues may have contained errors, not all sink computed numbers are accurate and equal. Thus, among these 10 numbers those out of dynamic range are removed. After a voting process, those numbers that have received the greatest votes are selected as correct numbers.

In order to gauge the error detection capability of the proposed set in a situation where errors occur in more than one modulo and, where therefore, no correction is possible, we simulated the behavior of the error control algorithm using C++ programming language. The experiment runs $30'000'000$ different error bit states by setting $n = \{2, 3, 4\}$. Substituting these values for n in the set we obtain $\{32, 31, 3, 127, 511\}$, $\{128, 127, 7, 1023, 8191\}$, and $\{512, 511, 15, 8191, 131071\}$, respectively. The test results are displayed in Table 3. Note that the maximum error correction is equal to $(2n+1)$ bits, as was mentioned before, and the three error states considered are bits which are located in one of the main modulus (correction is possible and detection is 100%), bits which are located in two moduli, and finally the case where bits are resided in three moduli.

With reference to Table 3 there are several individual conditions for error detection: among the 5 received residues only 1 residue contains error; 2 residues contain error; 3 residues contain error. Since 2 redundant moduli have been considered for this moduli set, error correction of a received residue is possible. It makes no difference whether just one or all bits of the modulo are damaged; this can be corrected. Based on the damaged bits of the residues we will have different conditions (e.g., it is possible that the first and second bits, the first and third bits, or any other combination of bits are damaged); simulation in C++ language considers all these conditions. All possible conditions are simulated for $n = 2$ (i.e., moduli set $\{32, 31, 3, 127, 511\}$), for $n = 3$ (i.e., moduli set $\{128, 127, 7, 1023, 8191\}$), and for $n = 4$ (i.e., moduli set $\{512, 511, 15, 8191, 131071\}$ and each time a part of data was damaged intentionally to investigate detection and correction capability of the method. Table 3 displays error detection percentage for different values of n. It also gives the most number of bits of a modulus that are corrected for various n values.

We evaluated the proposed scheme through NS-2 simulation tool [35]. We considered a randomly deployed network, that is, one hundred nodes that were initially spread over an area of 100 square meters in a random manner. The radio coverage of each sensor node and sink was assumed to be a circular area of 100 m in diameter. The antenna model deployed was omnidirectional. Also, the initial energy of sensor nodes set to 0.1 Joule.

The proposed scheme was analyzed in terms of energy consumption, end-to-end delay, and network lifetime using STCP [21], DTSN [22], TCP, and Reed Solomon. Table 4 shows the simulation parameters.

All simulations were repeated 10 times before calculating the average of the results. As illustrated in Figure 2, the simulation results confirm that the proposed scheme consumes significantly less energy than the other schemes investigated. In Figure 3 end-to-end delay is examined where it can be seen that the proposed scheme generates less network delay. Finally, in Figure 4 it is easy to see that the proposed scheme outperforms its counterparts in terms of network lifetime.

6. Conclusion

In this paper, authors unveiled an innovative energy-efficient algorithm for real time wireless sensor networks that was based on a new 3-moduli set $\{2^{2n+1}, 2^{2n+1}-1, 2^n-1\}$. By exploiting the MRC technique, an efficient reverse converter was also designed which overcame two-key constraints, namely, component space and high speed operation. In order to take full advantage of the error controllability property of the proposed solution, two-redundant-moduli sets were then added to the initial three main sets resulting in a combined set $\{2^{2n+1}, 2^{2n+1}-1, 2^n-1, 2^{3n+1}-1, 2^{4n+1}-1\}$.

The new method's error correction capability was found to be $(2n+1)$ bits under the ideal condition and error detection improvement reached a figure of 99.01% under the worst case scenario condition and 100% when $(n \geq 3)$.

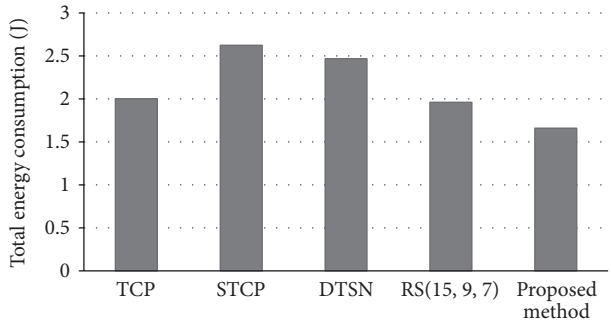

Figure 2: Total energy consumption.

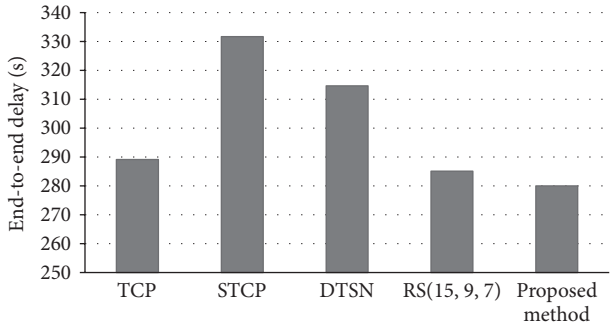

Figure 3: End-to-end delay.

Table 4: Simulation parameters.

Area of sensor field	$100 * 100\,m^2$
Antenna model	Omnidirectional
Simulation time	1000 S
Radio range of a sensor node	100 m
Number of sensor nodes	100
Initial energy	0.1 J
Interface queue type	Droptail
Interface queue (IFQ) length	50 Packets
Energy model	Battery

By simulating the proposed error control scheme, using an ns-2 network simulation tool and making comparisons against some popular methods, outperformance in terms of reductions of energy consumption and end-to-end delay as well as extension of network lifetime was clearly evident. These advantages therefore make the proposed method more desirable for real-time wireless sensor network applications where reliability, low energy consumption, and high speed operations are absolute necessities.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

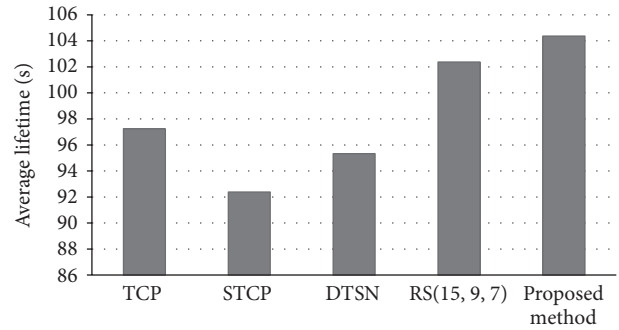

Figure 4: Average network lifetime.

References

[1] G. Song, Y. Zhou, F. Ding, and A. Song, "A mobile sensor network system for monitoring of unfriendly environments," *Sensors*, vol. 8, no. 11, pp. 7259–7274, 2008.

[2] J. Lloret, M. Garcia, D. Bri, and S. Sendra, "A wireless sensor network deployment for rural and forest fire detection and verification," *Sensors*, vol. 9, no. 11, pp. 8722–8747, 2009.

[3] M. Dunbabin and L. Marques, "Robots for environmental monitoring: significant advancements and applications," *IEEE Robotics and Automation Magazine*, vol. 19, no. 1, pp. 24–39, 2012.

[4] P. Kumar, S.-G. Lee, and H.-J. Lee, "E-SAP: efficient-strong authentication protocol for healthcare applications using wireless medical sensor networks," *Sensors*, vol. 12, no. 2, pp. 1625–1647, 2012.

[5] Y.-M. Hong, H.-C. Lin, and Y.-C. Kan, "Using wireless sensor network on real-time remote monitoring of the load cell for landslide," *Sensor Letters*, vol. 9, no. 5, pp. 1911–1915, 2011.

[6] S. Khodadoustan, F. Jalali, and A. Ejlali, "Reliability/energy trade-off in Bluetooth error control schemes," *Microelectronics Reliability*, vol. 51, no. 8, pp. 1398–1412, 2011.

[7] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[8] Z. Guo, B. Wang, P. Xie, W. Zeng, and J.-H. Cui, "Efficient error recovery with network coding in underwater sensor networks," *Ad Hoc Networks*, vol. 7, no. 4, pp. 791–802, 2009.

[9] I. F. Akyildiz and M. C. Vuran, *Wireless Sensor Networks*, John Wiley & Sons, Chichester, UK, 2010.

[10] E. Çayırcı and C. Rong, *Security in Wireless Ad Hoc and Sensor Networks*, John Wiley & Sons, Chichester, UK, 2009.

[11] J. H. Kleinschmidt, W. C. Borelli, and M. E. Pellenz, "An energy efficiency model for adaptive and custom error control schemes in Bluetooth sensor networks," *International Journal of Electronics and Communications (AEU)*, vol. 63, no. 3, pp. 188–199, 2009.

[12] A. S. Molahosseini, K. Navi, C. Dadkhah, O. Kavehei, and S. Timarchi, "Efficient reverse converter designs for the new 4-moduli sets $\{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1\}$ and $\{2^n - 1, 2^n + 1, 2^{2n}, 2^{2n}+1\}$ based on new CRTs," *IEEE Transactions on Circuits and Systems I*, vol. 57, no. 4, pp. 823–835, 2010.

[13] B. Cao, C.-H. Chang, and T. Srikanthan, "A residue-to-binary converter for a new five-moduli set," *IEEE Transactions on Circuits and Systems I*, vol. 54, no. 5, pp. 1041–1049, 2007.

[14] A. S. Molahosseini, C. Dadkhah, and K. Navi, "A new five-moduli set for efficient hardware implementation of the reverse

converter," *IEICE Electronics Express*, vol. 6, no. 14, pp. 1006–1012, 2009.

[15] M. Esmaeildoust, K. Navi, and M. Taheri, "High speed reverse converter for new five-moduli set $\{2^n, 2^{2n+1} - 1, 2^{n/2} - 1, 2^{n/2} + 1, 2^n + 1\}$," *IEICE Electronics Express*, vol. 7, no. 3, pp. 118–125, 2010.

[16] A. S. Molahosseini and M. K. Rafsanjani, "An improved five-modulus reverse converter," *World Applied Sciences*, vol. 11, no. 2, pp. 132–135, 2010.

[17] P. V. Ananda Mohan and A. B. Premkumar, "RNS-to-binary converters for two four-moduli sets $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1\}$ and $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} + 1\}$," *IEEE Transactions on Circuits and Systems I*, vol. 54, no. 6, pp. 1245–1254, 2007.

[18] B. Cao, T. Srikanthan, and C. H. Chang, "Efficient reverse converters for four moduli sets $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1\}$ and $\{2^n - 1, 2^n, 2^n + 1, 2^{n-1} - 1\}$," *IEE Proceedings on Computers and Digital Techniques*, vol. 152, no. 5, pp. 687–696, 2005.

[19] P. V. Ananda Mohan, "New reverse converters for the moduli set $\{2^n - 3, 2^n - 1, 2^n + 1, 2^n + 3\}$," *International Journal of Electronics and Communications (AEU)*, vol. 62, no. 9, pp. 643–658, 2008.

[20] P. Xie, Z. Zhou, Z. Peng, J.-H. Cui, and Z. Shi, "SDRT: a reliable data transport protocol for underwater sensor networks," *Ad Hoc Networks*, vol. 8, no. 7, pp. 708–722, 2010.

[21] Y. G. Iyer, S. Gandham, and S. Venkatesan, "STCP: a generic transport layer protocol for wireless sensor networks," in *Proceedings of the 14th International Conference on Computer Communications and Networks (ICCCN '05)*, pp. 449–454, San Diego, Calif, USA, October 2005.

[22] B. Marchi, A. Grilo, and M. Nunes, "DTSN: distributed transport for sensor networks," in *Proceedings of the 12th IEEE International Symposium on Computers and Communications (ISCC '07)*, pp. 165–172, Aveiro, Portugal, July 2007.

[23] M. Hosseinzadeh, A. S. Molahosseini, and K. Navi, "An improved reverse converter for the moduli set $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1\}$," *IEICE Electronics Express*, vol. 5, no. 17, pp. 672–677, 2008.

[24] H. Garner, "The residue number system," *IEEE Transactions Electronic Computer*, vol. 8, pp. 140–147, 1959.

[25] K. Navi, A. S. Molahosseini, and M. Esmaeildoust, "How to teach residue number system to computer scientists and engineers," *IEEE Transactions on Education*, vol. 54, no. 1, pp. 156–163, 2011.

[26] R. Chokshi, K. S. Berezowski, A. Shrivastava, and S. J. Piestrak, "Exploiting residue number system for power-efficient digital signal processing in embedded processors," in *Proceedings of the International Conference on Compilers, Architecture, and Synthesis for Embedded Systems (CASES '09)*, pp. 19–28, October 2009.

[27] R. Conway and J. Nelson, "Improved RNS FIR filter architectures," *IEEE Transactions on Circuits and Systems II*, vol. 51, no. 1, pp. 26–28, 2004.

[28] M. Wnuk, "Remarks on hardware implementation of image processing algorithms," *International Journal of Applied Mathematics and Computer Science*, vol. 18, no. 1, pp. 105–110, 2008.

[29] J.-C. Bajard and L. Imbert, "A full RNS implementation of RSA," *IEEE Transactions on Computers*, vol. 53, no. 6, pp. 769–774, 2004.

[30] J. Ramírez, A. García, U. Meyer-Baese, and A. Lloris, "Fast RNS FPL-based communications receiver design and implementation," in *Proceeding of the 12th International Conference Field Programmable Logic*, pp. 472–481, 2002.

[31] F. J. Taylor, "Residue arithmetic: a tutorial with examples," *IEEE Computer*, vol. 17, no. 5, pp. 50–62, 1984.

[32] F. Barsi and P. Maestrini, "Error correcting properties of redundant residue number systems," *IEEE Transactions on Computers*, vol. 22, no. 3, pp. 307–315, 1973.

[33] E. Kinoshita and K.-J. Lee, "A residue arithmetic extension for reliable scientific computation," *IEEE Transactions on Computers*, vol. 46, no. 2, pp. 129–138, 1997.

[34] N. Z. Haron and S. Hamdioui, "Redundant residue number system code for fault-tolerant hybrid memories," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 7, no. 1, article 4, 2011.

[35] "Network simulator—(ns-2)," http://www.isi.edu/nsnam/ns/.