COMPRESSION SCHEME FOR WIRELESS SENSOR NETWORKS USING A DISTRIBUTED SOURCE CODING ALGORITHM BASED ON RAPTOR CODE

Dragoş Mihai OFRIM[1], Dragoş Ioan SĂCĂLEANU[2], Vasile LĂZĂRESCU[3]

Această lucrare adresează problematica compresiei datelor în re elele de In this paper, the problem of distributed source coding for data compression in wireless sensor networks (WSN) is addressed. To achieve high levels of compression, a complete solution for network architecture, data correlation model and distributed source coding (DSC) algorithm is proposed. DSC is implemented using Raptor codes, the newest class of fountain codes. Rigorous tests proved better performance, in terms of compression rate, of the proposed solution compared to DSC schemes using LDPC or Turbo Codes. As tests also revealed, the differences in architecture between the proposed systematic version and the non-systematic version of Raptor code enable the implementation of DSC in a wide range of WSN applications.

Keywords: wireless sensor networks, Raptor codes, distributed source coding, data compression, data correlation

1. Introduction

Wireless sensor networks (WSN) have generated a lot of research during the past decade. Their main challenge is designing robust, low power devices that operate in industrial environments for a long period of time. Because they are battery powered and their processing capability is reduced, optimizations in data processing and data transmissions are needed to enhance the lifetime and throughput of the network.

Data compression is essential in reducing the amount of information sent over the wireless channel, thus reducing many cost functions of interest, like energy spent by sensor nodes, processing capabilities, data routing delay and efficiency. WSN compression schemes implementing algorithms based on source codes like Huffman [1] and Shannon-Fano-Elias [1] codes exploit internally the redundancy of data for each sensor node. More efficient approaches explore the

[1] PhD. Student, Faculty of Electronics, Telecommunicatiins and Information Technology, University POLITEHNICA of Bucharest, Romania, e-mail:, e-mail: dragosofrim@yahoo.com
[2] Faculty of Electronics, Telecommunicatiins and Information Technology, University POLITEHNICA of Bucharest, Romania, e-mail: dragos_sacaleanu@yahoo.com
[3] Prof., Faculty of Electronics, Telecommunicatiins and Information Technology, University POLITEHNICA of Bucharest, Romania, e-mail: vl@elia.pub.ro

inter-correlation of data between the sensors. These are based on the groundbreaking theory developed by Slepian and Wolf [2], which allows distributed source coding (DSC) of correlated sources. The sensor nodes are modeled as correlated sources of information using mathematical models that exploit the dependency in the measured data at each sensor, thus enabling data compression. Successful DSC schemes have been implemented using block codes (including LDPC – *Low Density Parity Check*) [3] and Turbo Codes [4].

This paper proposes a new compression scheme for WSN implemented using Raptor codes, the newest class of rateless codes. The advantage of using Raptor codes in real DSC applications is that they offer a low complexity of the encoder and a flexible code rate that can always be adjusted to match the changes is the data correlation parameters. Moreover, as the tests will reveal further in the paper, Raptor code, with both its systematic and non-systematic version, enables a great range of WSN applications, from simple environment monitoring, to video sensors.

2. Fundamentals of distributed source coding

During this paper, random variables are noted using capital letters, e.g., X, Y. Vectors are denoted by lower-case letters, e.g., **x**, **y** and matrices by bold upper-case letters, e.g. $\boldsymbol{G_{k \times n}}$.

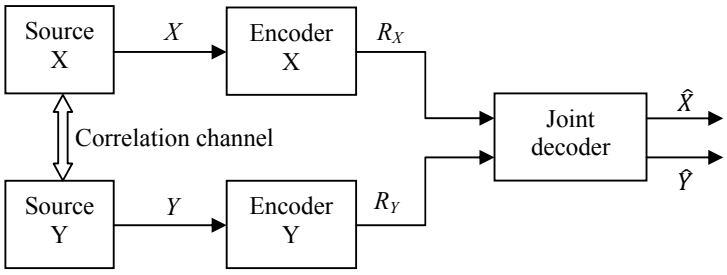

Fig.1 Distributed compression of two correlated, i.i.d., discrete random sequences, X and Y

In WSN, sensor nodes are modeled as independent, identically distributed (i.i.d) sources of information. Let (X, Y) be a pair of two correlated i.i.d sources. One way to model the correlation between these two sources is via a "correlation channel", where $X = Y + N$ and N is a random variable characterizing the noise of the correlation channel. Thus, one of the two sources is modeled as a noisy version of the other.

Slepian-Wolf theorem [2] states that two correlated sources, X and Y, can be coded separately and decoded jointly (Fig. 1). Each source encodes the information at a certain rate and sends in further to the decoder. The rate bounds

which ensure a lossless recovery of both sources at the decoder are [2]: $R_X \geq H(X|Y)$, $R_Y \geq H(Y|X)$, and $R_X + R_Y \geq H(X,Y)$. In practice, the asymmetric rate bounds are used for simplicity of implementation: $R_X \geq H(X|Y)$ and $R_Y \geq H(Y)$. This way, one source, Y, is perfectly known at the decoder, whiles the other one, X, needs to be perfectly recovered at the decoder side. Compression is achieved for source X, as the information rate R_X is lower than the source entropy $H(X)$.

The key in implementing distributed source coding in WSN is using channel codes. Although classic compression schemes use other types of codes which eliminate the source's redundancy, channel codes add redundancy to the source information. This redundancy is used to recover the possible errors which occur during the transmission over a communication channel. In the DSC case, the communication channel is represented by the correlation channel and source X represents a noisy version of the source Y.

Based on the Slepian-Wolf theorem [2], Wyner [5] developed a practical approach to implement DSC. Each possible sequence of source X is indexed using parity-check bits from a systematic channel code. To encode a source word $\mathbf{x}$ of k bits, a systematic codeword $(k + s, k)$ is needed, generated using the generator matrix $\mathbf{G}_{k\times(k+s)} = [\mathbf{I}_k|\mathbf{P}_{k\times s}]$. The s-tuple $\mathbf{p} = \mathbf{xP}$, which are the parity bits, represents the compressed information that is sent to the decoder. Here, the $(k + s)$-bits codeword $\mathbf{g} = [\mathbf{y}_{1\times k}|\mathbf{p}]$ is created by attaching the side information $\mathbf{y}_{1\times k}$, from source Y, to the received parity information $\mathbf{p}$ from source X. By decoding $\mathbf{g}$, the original codeword of source X is estimated. To ensure a lossless recovery, $R_X = {}^{s}/_{k} \geq H(X|Y)$.

3. Distributed source coding using Raptor code

Raptor codes [6] are the newest family of *fountain* codes, also called *rateless*, which are erasure protection codes. Raptor codes are based on the previous version of rateless codes, the Luby Transform – LT [7]. An LT code with parameters $(k, C, \Omega(x))$ generates an infinite number of encoded symbols from the source symbols using a distribution $\Omega(x)$ [7]. To generate an encoded symbol, the encoder samples a degree d from the distribution and then randomly chooses d source symbols which are then XORed to form an encoded symbol. This process can be represented using a Tanner graph [8], in which the source nodes are called *variable nodes* and the encoded symbols are named *check nodes*. An edge connecting a variable node to a check nodes means that the corresponding source symbol is among the symbols XORed to form the corresponding encoded symbol. To ensure a succsessful decoding, the associated Tanner graph of the LT code must have at least $ck\log(k)$ edges.

One of the disadvantages of the LT code is that the encoding and decoding costs are not constant. To relax the condition of having at least $ck\log(k)$ edges,

Amin Shokrollahi developed the Raptor code [6]. Raptor code is the concatenation of an LT code with a precode, a high-rate systematic linear code, like LDPC code. The precode ensures that the encoding and decoding complexity of the LT code varies only lineary with the number k of source symbols. Thus, the LT code is required to recover only a constant ratio of the source symbols, while the precode recovers the remaining symbols.

The ability to continuously sample from the distribution $\Omega(x)$ assures the rateless characteristic of the Raptor code. In the DSC case, this is a great advantage, as the same code architecture can match any desired rate R_X, even when it is dynamically changed at runtime. Moreover, the low complexity Raptor encoder best suites the common applications of WSN, where, due to the low capabilities of the wireless sensor nodes, low processing necesities are required.

As mentioned before, the DSC arhitecture requires channel codes with error correcting capabilities, while fountain codes were designed for errasure protection [6,7]. To fully benefit of the Raptor code advantages in the DSC case, the decoding algorithm must enable the correction of the „correlation errors". This paper proposes the implementation of a *soft decoding* algorithm like *belief propagation* [9], which enables error correction capabilities for the Raptor code.

A.Encoding

To fully expose the advantages of DSC with Raptor code in WSN, this paper proposes a systematic and a non-systematic version of Raptor code. Both structures use a systematic high-rate LDPC code as precode, defined by the $(k+s)$ x k generator matrix $\mathbf{G}_{LDPC}$. The LT code is characterised by a N x $(k+s)$ generator matrix $\mathbf{G}_{LT}$, with N not fixed, depending on the desired rate R_X. Let $\mathbf{x} = \{x_1,x_2,x_3,\ldots,x_k\}$ be a realization for k source symbols of X.

In the proposed non-systematic version of Raptor code, the source symbols are first encoded with a systematic LDPC code, resulting a vector of $k+s$ LDPC symbols $\mathbf{x'} = \mathbf{G}_{LDPC}\ \mathbf{x}^{T}$, $\mathbf{x'} = \{x_1,x_2,x_3,\ldots,x_k,x'_{k+1},\ldots,x'_{k+s}\}$. Then, the LDPC symbols are coded using the LT code, resulting a number of $N=p$ coded symbols $\mathbf{x}^{*} = \mathbf{G}_{LDPC}\ \mathbf{x'} = \{x_1^*,\ldots, x_p^*\}$ which are sent to the decoder. According to Slepian-Wolf theorem [2], $R_X = {}^{p}/_{k} \geq H(X|Y)$. In the proposed non-systematic version, the source symbols are not among the coded symbols, as the LT code randomly XORes the LDPC symbols to calculate the coded symbols.

The proposed systematic Raptor encoder assures that the source symbols are among the coded symbols. For that, a vector of $k+s$ intermediate symbols is generated at first

$$x^* = \mathrm{A}^{-1} \times d^T$$

$$A_{(k+s)\times(k+s)} = \begin{bmatrix} G\_LDPC_{s\times k} & I_{s\times s} \\ G\_LT_{k\times(k+s)} \end{bmatrix}; \quad d = (0,0,....,0, x_1, \dots, x_k) \qquad (1)$$

where **A** is a full rank (k+s) x (k+s) matrix and d is a (s+k)-tuple formed by s zeros and the k source simbols :

These intermediate symbols are then encoded using a (k+p) x (k+s) $\mathbf{G}_{LT}$ generator matrix, resulting an output vector $\mathbf{x}^* = \{x_1, x_2, \dots, x_k, x^*_{k+1}, \dots, x^*_{k+p}\}$ containing, in the first k positions, the source symbols. In this case $N = k + p$, and the rate is $R_X = {}^{p}/_{k} \geq H(X|Y)$, because in the systematic version only the $\{x^*_{k+1}, \dots, x^*_{k+p}\}$ output symbols are sent to the decoder.

B. Decoding

The joint decoder of the DSC scheme receives R_X and R_Y bits from the two correlated sources X and Y. Source symbols $\mathbf{y} = \{y_1, y_2, y_3, \dots, y_k\}$ of Y are perfectly known at the decoder and are considered side information for decoding the original source symbols **x**, along with the R_X parity bits. To properly decode the original source symbols of X, the proposed *belief propagation* [9] algorithm is used. The side information **y** is interpreted as a noisy version of **x**.

The *belief propagation* is a message passing algorithm based on sending *belief* messages between the nodes of the graph. As stated before, the Raptor scheme is represented using a Tanner graph [8]. For every source symbol x_i the aposteriori probability $\Pr(x_i = 1|\mathbf{y})$ that the bit has value 1, knowing the side information **y**, is calculated. In reality, as a measure of trust, the *log-likelihood ratio (LLR)* $L(x_i)$ is used

$$L(x_i) \stackrel{\text{def}}{=} \ln \frac{\Pr(x_i = 0|\mathbf{y})}{\Pr(x_i = 1|\mathbf{y})} \qquad (2)$$

There are two types of nodes in a Tanner graph: *variable nodes* and *check nodes*. Using general notation, a message $q_{ij}(x)$ from a variable node x_i to a check node u_j represents the propability that the variable node x_i has a certain value, knowing all the extrinsic information received by the variable node x_i from all the check nodes it is connected to, except u_j. The message $r_{ji}(x)$ from a check node u_j to a variable node x_i represents the probability that the u_j parity is checked, knowing x_i and the distributions of the other variable nodes connected to u_j (other than x_i), depicted by their corresponding messages sent to u_j. The following formulas apply to these messages:

$$\tanh\left(\frac{r_{j\to i}}{2}\right) = \prod_{i' \neq i} \tanh\left(\frac{q_{i'\to j}}{2}\right)$$

(3)

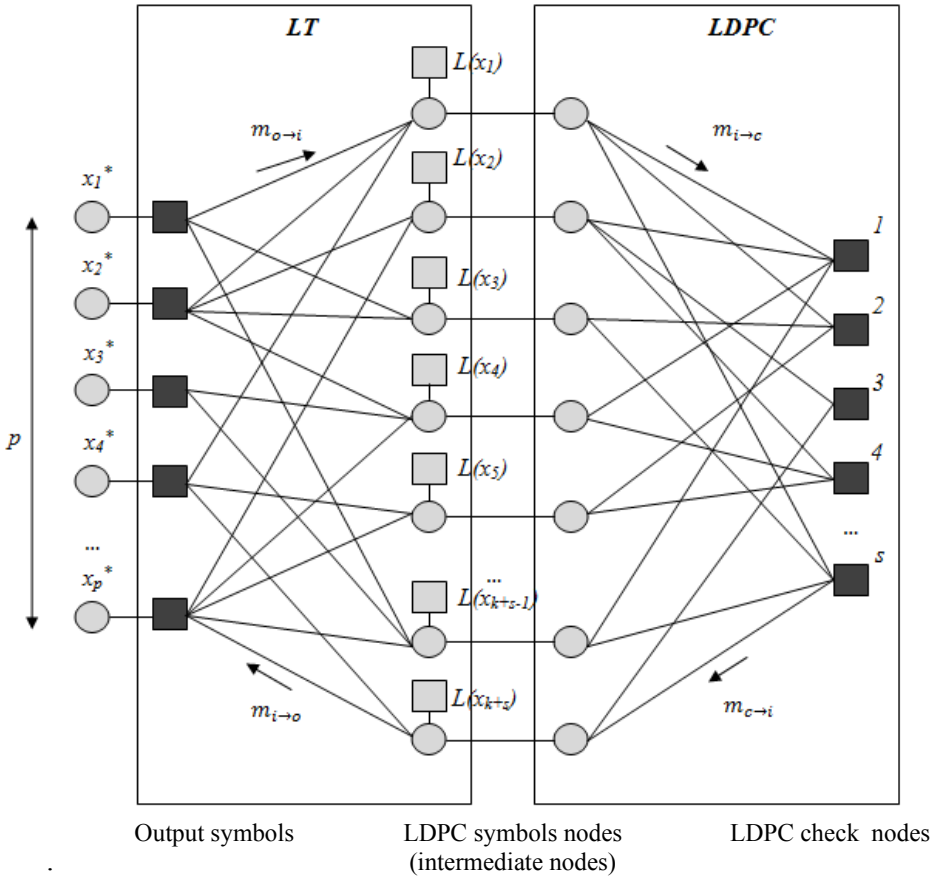

Fig. 2 Tanner graph associated to the non-systematic Raptor code and belief propagation algorithm for decoding the source symbols

For the proposed non-systematic version, the associated graph is depicted in Fig. 2. Variable nodes are marked as circles, while the check nodes are represented by squares. The LLRs for the source symbols are calculated using (2), while the LLRs for the LDPC parity symbols are initilized with zero, as there is no apriori information about them.

The messages respect the rules in (3) and the following notations are used:

- and – messages sent from the intermediate nodes to the output nodes and, respective, from the output nodes to the intermediate nodes, inside the LT graph, at iteration *l*.

- $m_{i\to c}^{(l)}$ and $m_{c\to i}^{(l)}$ - messages sent from intermediate nodes to LDPC check nodes and, respective, from LDPC check nodes to intermediate nodes, inside the LDPC graph, at iteration *l*.

The decoding algorithm is performed in two steps: first, messages are passed through the associated graph of the LT code until no more errors can be corrected or the maximum number of iterations for this phase is reached; second, messages are passed through the LDPC decoder, using the updated LLRs from the LT decoding step. The algorithm stops when either all the source symbols are recovered, or the maximum number of iterations is reached. At each iteration, the new values of the LLRs are evaluated and the source symbol is estimated, using the following decision scheme:

$$\hat{x}_i = \begin{cases} 0 & \text{if } L(x_i) \geq 0 \\ 1 & \text{if } L(x_i) < 0 \end{cases} \tag{4}$$

For the systematic Raptor case, the associated Tanner graph is depicted in Fig. 3. Different from the non-systematic case, in this configuration the LLRs are associated to the first *k* output symbols, as the source symbols are found within the Raptor codeword. The middle nodes represent now the intermediate symbols, as resulted from (1). Besides the messages defined for the non-systematic version, two more intermediate messages are used in this systematic case:

- $m_{LT}^{(l),i}$ - messages generated by the intermediate node *i* from the LT graph
- $m_{LDPC}^{(l),i}$ – messages generated by the intermediate node *i* from the LDPC graph

The same cascading scheme is used for the decoding algorithm of the proposed systematic version: first, messages through the LT graph are passed and the new LLRs are calculated at the output symbols. At the end of the LT decoding step, occuring when the maximum number of iterations is reached or when no more errors can be corrected, the LDPC decoding starts, passing messages from the output nodes to the LDPC check nodes. At each iteration, the source symbols are estimated using (4). The algorithm ends when either all the source symbols are recovered, or the maximum number of iterations is reached.

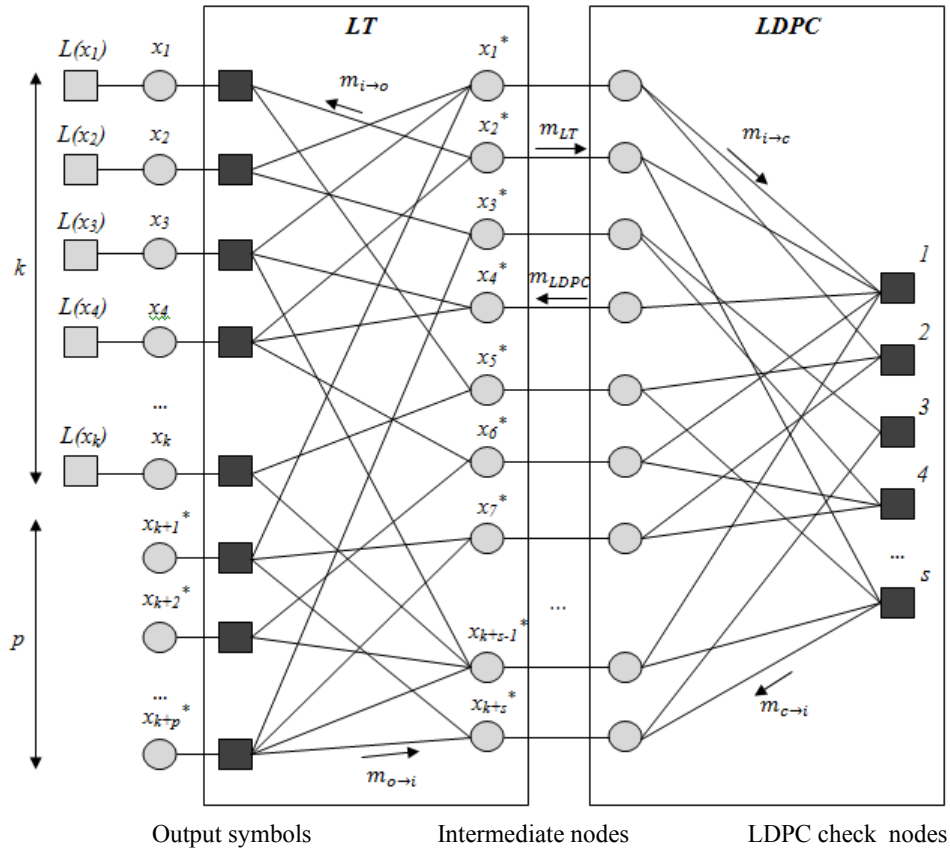

Fig. 3 Tanner graph associated to the systematic Raptor code and belief propagation algorithm for decoding the source symbols

4. Data correlation model

To effectively implement the proposed DSC in a real wireless sensor network application, several tasks need to be established: first, the architecture of the WSN has to be designed for the proper implementation of the DSC scheme; second, the mathematical model for the data correlation must be feasible and reflect the actual correlation of the measured data.

A. The proposed DSC architecture for wireless sensor networks

The proposed architecture for the WSN is cluster-based, as depicted in Fig. 4. DSC is applied within each cluster as follows: the source *Y*, which constitutes the side information, is represented by the cluster-head (CH).

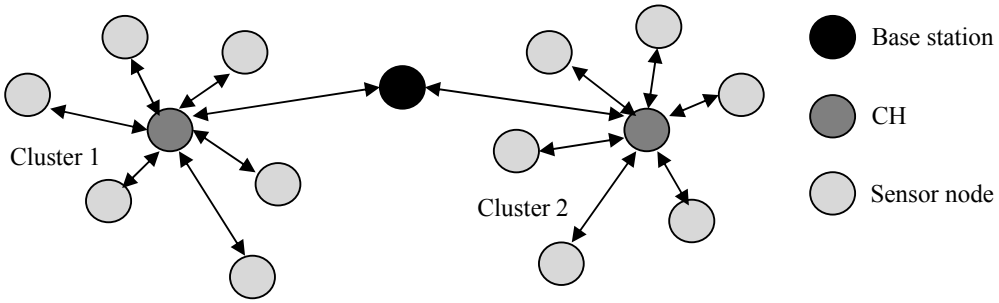

Fig. 4 Cluster based architecture of the WSN.

The other nodes of the cluster represent the sources X_i, $i = 1,\ldots,n$. The CH gathers all the information from the other nodes and sends it further to the base station, which performs the decoding. Thus, as side information, the CH has rate $R_Y \geq H(Y)$, while each sensor node has rate $R_{Xi} \geq H(X_i|Y)$.

B. Correlation model based on a binary symmetric channel

As DSC is implemented using channel codes, one way to model the correlation is by using a standard channel model, like the *binary symmetric channel* (BSC) [10]. For each received bit, p represents the probability that an error occurred during the transmission, while (1 - p) represents the probability that the bit was successfully received. Following this model in the DSC schemes, (1 – p_i) represents the probability that a bit x_{ij} from the source X_i has the same value as the corresponding bit y_j from the side information Y. Therefore, the information rates for sources X_i become [10]:

$$R_{Xi} \geq H(X_i|Y) = -p_i \log p_i - (1 - p_i) \log(1 - p_i) \tag{5}$$

At the decoder, the LLRs are calculated using the BSC model and (6):

$$L(x_{ij}) = (1 - 2y_j) ln \frac{1-p_i}{p_i} \tag{6}$$

The belief propagation decoding algorithm is then initialized. In case the decoder is unsuccessful in recovering the original source symbols $\mathbf{x_i}$, that means rate R_{Xi} was under estimated and the base station informs the sensor node X_i to increase the rate. The power of the Raptor code is that the rate can easily be adjusted, without changing the architecture of the code, by just sampling more values from the distribution $\Omega(x)$ of the LT code, thus increasing the number of generated code bits.

5. Experimental results

To justify the proposal of using Raptor code for DSC, over other channel codes, the performance over BSC correlation model of both systematic and non-systematic proposed versions has been evaluated. Comparison were made with state of the art LDPC [3] and Turbo codes [4], which are also used in distributed source coding of correlated sources.

First, the evolution of bit error rate (BER) over the source rate has been tested (Fig. 5). The source word length was 5000 bits and the compression rate was set to 2. The theoretical Slepian-Wolf (SW) limit of the rate is then $H(X|Y) = 0.5$. The entropy was continuously decreased, keeping the compression rate constant, until a low BER was reached.

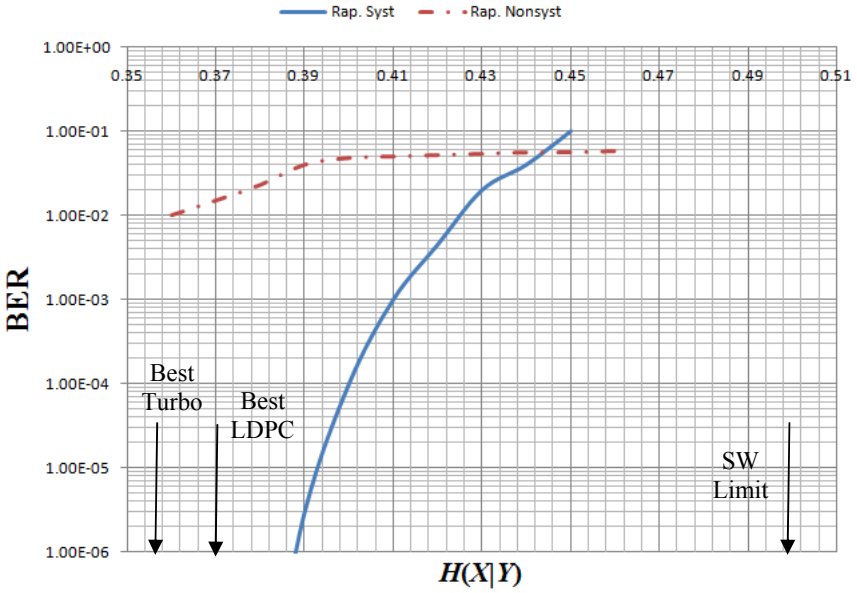

Fig. 5 BER vs. entropy for different codes

The superior performance of the proposed systematic Raptor code over the best Turbo and LDPC codes is visible in Fig. 5. The non-systematic version has the poorest performance. On the other hand, it can be observed that the theoretical SW limit cannot be reached in real implementations.

Second, the variation of the compression rate for different probabilities of error p has been studied. The case with BER very close to 0 has been considered. The source word length was 5000 bits.

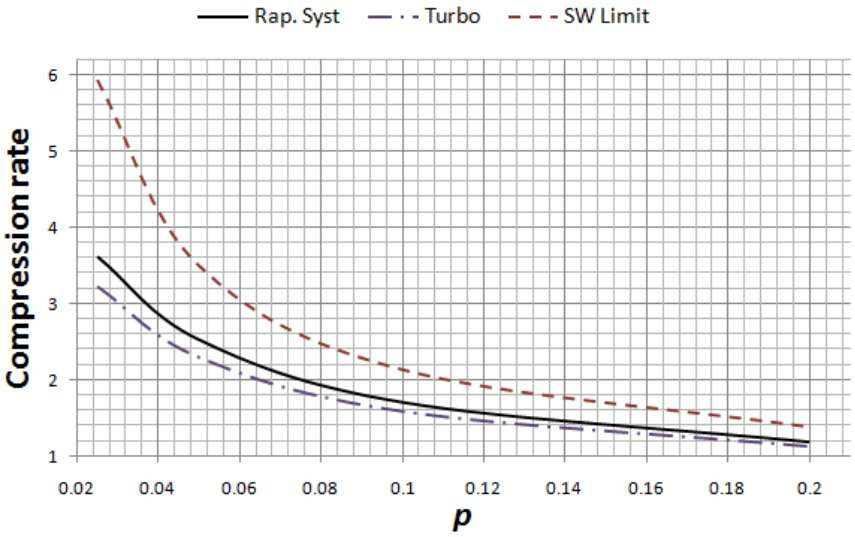

Fig. 6 Compression rate vs. *p* for Raptor systematic and Turbo code

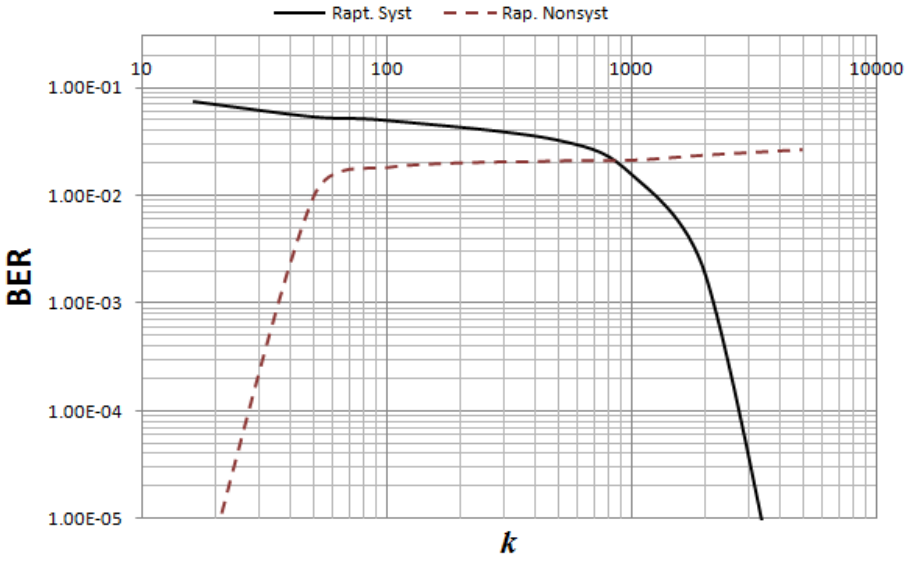

Fig. 7 BER vs. *k* for systematic and non-systematic Raptor code

The superiority of the proposed systematic Raptor code over the Turbo code can be observed in Fig. 6. The SW limit is marked as a reference. The next study was the comparison of the two version of Raptor code at different source

word lengths *k*. The compression rate was kept constant at a value of 2. It can be observed in Fig. 7 that the systematic version of Raptor code has very low BER for long source words, while the nonsystematic version has great performance for small source words.

6. Conclusions

This paper presents a novel compression scheme based on DSC, implemented using Raptor code. In the proposed WSN architecture, the systematic Raptor code outperforms the best implementations based on state of the art LDPC and Turbo codes in terms of compression rate. Compared to the non-systematic version, the systematic Raptor code has a very low BER when long source words are used, while the non-systematic code reaches BERs close to zero when small code words are used. This very important discovered feature enables the use of Raptor code in a large variety of wireless sensor networks applications, from environmental monitoring, where small code words are used, to multimedia applications, where long data streams are processed.

R E F E R E N C E S

[1] *T. M. Cover and J. A. Thomas*, Elements of Information Theory, 2nd edition, New York: Wiley, 2006,pp.118-130

[2] *D. Slepian and J. K. Wolf*, Noiseless coding of correlated information sources, IEEE, Transactions on Information Theory, vol. 19, no. 4, pp. 471-480, July 1973

[3] *A. D. Liveris, Z. Xiong, and C. N. Georghiades*, Compression of binary sources with side information at the decoder using LDPC codes, IEEE Commun. Lett., vol. 6, pp. 440–442, 2002.

[4] *A. Aaron and B. Girod*, Compression with side information using turbo codes, in Proc. Data Compression Conf. (DCC'02), Washington, DC, 2002, p. 252

[5] *A. Wyner*, Recent results in the Shannon theory, IEEE Transactions on Information Theory, vol. 20, no. 1, pp. 2-10, January 1974.

[6] *A. Shokrollahi*, Raptor codes, in IEEE International Symposium on Information Theory, *I*SIT, 2004, Chicago, IL, June-July 2004.

[7] *M. Luby*, LT-codes, in IEEE Symposium on the Foundations of Computer Science, FOCS, 2002, Vancouver, Canada, November 2002.

[8] *R. M. Tanner*, A recursive approach to low complexity codes, IEEE Transactions on Information Theory, vol. 27, no. 5, pp. 533-547, September 1981.

[9] *F.R. Kschischang, B J. Frey, and H.A.Loeliger* "Factor Graphs and the Sum-Product Algorithm," IEEE Transactions on information theory, vol. 47, no. 2, february 2001

[10] *T. M. Cover and J. A. Thomas*, Elements of Information Theory, 2nd edition, New York: Wiley, 2006,pp.187-188