ORIGINAL RESEARCH PAPER
Pages: 168-183

Improving the Efficiency of Wireless Sensor Networks Using Fountain Codes

S. M. Mirrezaei
Faulty of Electrical Engineering, Shahrood University of Technology, Shahrood, Iran
E-mail: sm.mirrezaei@shahroodut.ac.ir
Corresponding author: S. M. Mirrezaei

DOI:10.22070/jce.2021.5334.1156

***Abstract*- Fountain codes are erasure codes that are characterized by their rateless property and their global acknowledgment. The larger the network size, the more efficient the fountain codes are degraded because of multi-hops causing an overflow. The optimization of wireless communication is also a focus of study exciting and an important issue always to maximize performance, the lifetime of the sensor nodes, and to reduce the consumption of energy. Estimation becomes one of the attractive topics in wireless sensor networks nowadays. In this paper, I consider a distributed estimation scheme composing of a sensor member and a fusion center, which is the cluster head. To minimize the number of transmissions as well as the impact of overflow, I determine the optimal minimal number of encoded packets needed for successful decoding. Sensor observations are encoded using fountain codes, and then messages are collected at the cluster head where a final estimation is provided with a classification based on Bayes rule. The main goal of this paper is to estimate the total number of received packets using the Bayes rule so that it is possible to minimize the overflow and extend the network lifetime.**

Index Terms- Wireless Sensor Networks, Fountain codes, Estimation, Bayes rule, Naive Bayes.

I. INTRODUCTION

Wireless Sensor Network (WSN) consists of a large number of geographically distributed sensor nodes. Research in this field is very attractive, and scientists, as well as researchers, must be ingenious, inventive, and anticipatory to propose new techniques to reduce energy consumption [1]-[3]. It performs activities in several dimensions, for instance, identifying the neighborhood, presence of targets, or monitoring environmental factors (motion, temperature, humidity, sound, and other physical variables) [4,5]. However, owing to limited battery power, sensor networks need resolution with energy savings to increase sensor network performance. Prolonging battery life is a principal objective in the design of wireless sensor networks due to the difficulty and high cost associated with replacing or recharging exhausted batteries in a deployed network. There is, therefore, a problem of

Manuscript received 15-March-2020 and revised 25-May-2020,
Accepted on 17-June-2020
P- ISSN: 2322-4088
E- ISSN: 2322-3936

energy preservation. At the same time, a node in this network can be a transceiver. Given the limited communication range of sensor nodes, sending data from a node to the sink typically results in sets of hops through the network.

On the other hand, transmissions using radio channel can generate a low level of communication reliability. Indeed, the transmitted information can be subject to errors caused by the radio channel itself, fading, or because of the coexistence of multiple transmissions. To protect data against channel failures, two solutions can be envisaged. Either acknowledgment mechanisms or retransmission techniques can be used. However, the major disadvantage of these techniques is still the high energy cost.

Recently, appeared erasure codes known as fountain codes. Fountain codes are the object of a renewed interest aroused by numerous research projects due to their potential capacity to make the transmission more reliable while limiting the use of the feedback channel. The greater the network size, the more fountain codes efficiency degrades because of the multi-hops generating useless data known by overflow. Decoding and re-encoding information raised a significant degradation problem of the network efficiency by increasing latency and energy consumption. Besides, the re-encoding by intermediate nodes modifies the degree distribution of the information at the receiver side. Generally, one of the most efficient ways to deploy WSN is to organize sensor networks into clustered architectures. Clustering is one of the basic approaches for designing energy-efficient, robust and highly scalable distributed sensor networks. Using clusters reduces the communication overhead, thereby decreasing the energy consumption and interference among the sensor nodes. To exploit correlation and eliminate redundancy that often exists among sensors, clustering is required in many applications. Through data aggregation at the cluster centers, called cluster heads(CHs), the total amount of data sent to the sink can be significantly reduced, saving energy and bandwidth resources.

Data collection is considered an essential process for preserving sensor energy. Indeed, data are collected from the sensor and sent to the CH and then to sink for analysis [6]. Each elected CH must anticipate the amount of encoded packets collected. This anticipation aims at acknowledging the source and stop generating encoded data whenever their reception is enough. In this paper, using sets of information provided by each sensor, I realize a distributed estimation of needed data. Many distributed estimation algorithms [7]-[10] have been proposed. In these schemes, each sensor transmits its measurements using few bits to the CH directly. One way to reduce energy consumption is to process the data locally, such as encoding data at local sensors. Sensors perform local data encoding and send resulting messages to CH that combines received messages to produce a final estimation of the desired parameter. In [11] the authors discussed and compared their proposed distributed estimation scheme with some machine learning-based methods for data classification. This study shows two main objectives:(i) that deep learning models offer the potential for fountain codes in terms of data classification. (ii) a comparative analysis of the most principal machine learning classification algorithm in WSN. Their proposed model looks advantageous over other methods due to its performance to accurately classify data with error rate notably low. Various studies and algorithms have been proposed for distributed estimation in wireless sensor networks [12]-[15].

In [16-17], a new routing algorithm based on a hierarchical clustering is proposed for wireless sensor networks. In each level of this hierarchical clustering, the appropriate nodes are selected as cluster heads and then formed the clusters. The proposed algorithm is distributed and selects cluster heads and forms the clusters with the smallest number of the control messages.

The main contributions of this paper are the introduction of:

- The use of fountain codes is even more effective if they are associated with other techniques, like the self-organization structure. The primary purpose of the self-organization is to minimize the transfer in the network and improve the performance of the sensors. Clustering is a self-organizing strategy based on the division of the network into small areas called clusters.
- Designing an efficient distributed estimation scheme to collect fountain codes data packets.
- Estimate and determine the minimum encoded packets needed to recover information based on Bayes rules.
- I propose and adopt a new distributed estimation scheme composing a sensor member and a CH based on Bayesian rules. Each CH anticipates and acknowledges the source to stop the generation of data, whenever there is a sufficient number of collected encoded packets.

The remainder of this paper is organized as follows: in section 2, I present a preliminary study to describe the general context of my work. Section 3 presents the model and problem formulation. In section 4 the system model for estimating unknown target parameters is formulated. To evaluate and demonstrate the performance of the proposed model, an example of classification using Bayes rule was presented in section 5 with an analysis of different simulations and results. Section 6 gives finally concluding remarks.

II. PRELIMINARY

Fountain codes are an attractive and interesting class of erasure codes. The source first splits the message to K fragments or input symbols and then starts sending practically unique encoded packets [18]. When the receiver has acquired enough encoded packets to recuperate original data, it sends an acknowledgment (Ack) to stop the generation of encoded packets. LT code (Luby Transform code) was the first practical example of the erasure code fulfilling the fountain coding principle [17]. For simplicity of explanation, I suppose that the message M comprises K equally input symbols, $M = \{f_1, f_2, \dots, f_K\}$. The encoded packet is obtained by first choosing a random degree d from the degree distribution, which is the Robust Soliton Distribution RSD [19,20] and then XOR-ing the randomly chosen input symbols. Soliton distribution was a probability distribution used with erasure codes presented by Michael Luby. Luby introduced in his paper [21] a detailed study of the two forms of

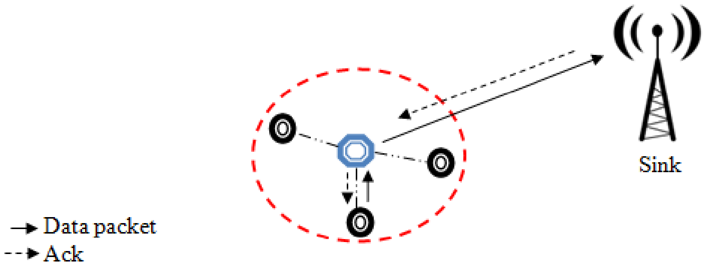

Fig. 1. The model of a wireless sensor network

distribution; Ideal Soliton Distribution (ISD) [22] and Robust Soliton Distribution (RSD). RSD is $\mu(.)$ defined as follows. Let:

$$R = c \times ln(\frac{K}{\delta})\sqrt{K} \tag{1}$$

is the average number of degree 1. Some suitable constant c > 0 and δ is the success decoding probability. Let δ be the allowable failure probability of the decoder to recover the data for a given number K of encoded symbols. The idea here is to design the distribution so that the expected ripple size is about $ln(\frac{K}{\delta})\sqrt{K}$ throughout the process. The intuition is that the probability of a random walk of length K deviates from its mean by more than $ln(\frac{K}{\delta})\sqrt{K}$ is at most δ.

$$\varphi(i) = \begin{cases} \frac{R}{iK} & for\ 1 \leq i \leq \frac{K}{R} - 1 \\ \frac{Rln(\frac{R}{\delta})}{K} & for\ i = \frac{K}{R} \\ 0 & for\ i > \frac{K}{R} \end{cases} \tag{2}$$

and $\rho(i)$ is the Ideal Soliton Distribution defined by the expression (3) below:

$$\rho(i) = \begin{cases} \frac{1}{K} & for\ i = 1 \\ \frac{1}{i(i-1)} & for\ i = 2, \ldots, K \end{cases} \tag{3}$$

where *i* indicates the symbol number being coded.

Add the Ideal Soliton distribution $\rho(i)$ to $\varphi(i)$ and normalize to obtain $\mu(.)$:

$$\mu(i) = \frac{\rho(i) + \varphi(i)}{\beta} \tag{4}$$

where $\beta = \sum_i(\rho(i) + \varphi(i))$.

More studies and analyzes of soliton distribution are detailed in divers research [21-25].

III. MODEL AND PROBLEM FORMULATION

I consider a WSN divided into clusters. Each cluster group is a set of *N* sensor member managed by a CH, as it is shown in Fig.1. I assume that sensors are distributed and there are no inter-sensor communications. For most wireless sensor networks, sensors suffer from limited battery power as well as communication capacity [26]. Therefore, encoding local data seems to be needed at each sensor to reduce the communication requirement between sensors and CH. Using fountain codes, each sensor has *K* input symbols in a vector $M = \{f_i;\ 1 \leq i \leq K\}$, encodes its selected blocks $f_i, 1 \leq i \leq d$ to p_i using fountain code function and transmits its encoded packets to the CH. The encoded packet $p_i = \oplus_{i=1}^{d} f_i$ is the XOR of the *d* input symbols transmitted and aggregated at the corresponding CH.

Encoded packets that contain only one fragment (with degree 1) are released first to recuperate their single fragment. Released fragments are saved in buffers named ripple for later using one by one when decoding data until recovering all input symbols.

The processing of each input symbol in the ripple is as follows:

1. It is removed from the ripple.
2. It is removed as a neighbor from all encoded symbols that have it as a neighbor.
3. For each encoded symbol with exactly one remaining neighbor, its remaining neighbor is released; this operation is called a symbol release. For each encoded symbol with zero remaining neighbors, its neighbors have been released before; this operation is called a symbol recovery.
4. Newly released input symbols previously unrecovered are added to the ripple.

Ripple size is the main criterion of decoding success. Decoding crashes if the ripple is empty before all input symbols are released (there are no more fragments left for decoding). Consequently, for the convergence of the decoding, it is necessary to ensure an impressive size of the ripple to avoid blocking. On the other hand, this size must be kept low to avoid the redundancy of the encoded symbol. The evolution of ripple size is the criterion of decoding success.

Decoding is successful when all input symbols have been recovered. If the ripple size is equal to zero before the end of decoding, decoding fails. This hints that the well-performing LT codes should ensure a large ripple size during the decoding process. However, when an encoded symbol is released, there is a risk that the encoded symbol is redundant. Hence, to minimize the risk of redundancy, the ripple size should be kept low.

The decoding success is judged based on the size and the evolution of the ripple size.

Release and Ripple Add Probability: is the probability that a symbol of original degree *d* is released and added to the ripple, when *L* out of the *K* input symbols remain unprocessed, given that the ripple size is *R* at the point of release. This probability is expressed as follows:

$$q(1, K, 0) = 0 \tag{5}$$

$$q(d,K,R) = \frac{d(d-1)(L-R+1)\prod_{i=0}^{d=3}(K-(L+1)-i)}{\prod_{i=0}^{d-1}(K-i)}$$

$$for \quad d = 2,\ldots,K-R+1, \qquad L = R,\ldots,d+1$$

Redundancy Probability: is the probability that a symbol of original degree d is redundant in the ripple. Assuming a constant ripple size *R*, the redundant probability is defined by the expression below:

$$r(d,R) = 1 - \sum_{L=R}^{K-d+1} q(d,L,R) \tag{6}$$

$$for \quad d = 2,\ldots,K-R+1, \qquad R = 1,\ldots,K-1.$$

IV. BAYESIAN METHOD AND DECISION

The probabilistic model is a conditional model, based on Bayes rule, which takes the following form: the probability of realization of an event *A* given all the events $B_i, 1 \le i \le n$; is given by the ratio between the probability of occurrence of events B_i given *A* and the probability that B_i has occurred [27]. This definition takes this form:

$$\begin{aligned} P(B_1,\ldots,B_n|A) * P(A) \\ &= P(A) * P(B_1|A) * P(B_2,\ldots,B_n|A,B_1) \\ &= P(A) * P(B_1|A) * P(B_2|A,B_1) * P(B_3,\ldots,B_n|A,B_1,B_2) \\ &= P(A,B_1,\ldots,B_n) = P(A)\prod_{i=1}^{n} P(B_i|A) \end{aligned} \tag{7}$$

Using the network architecture proposed and presented in Fig. 1, each sensor member sends its observation to the corresponding CH. The fusion center, which is the CH in this case, collects received data packets and under constraints, it estimates if the received number of encoded packets is enough or not [28]. The Bayesian method which is a classifier, focuses on the update of the probability estimate using the probability of the data under a given hypothesis (likelihood). Then it deduces the probability of an event from other events already evaluated called prior [27]. Based on the expression (7), the problem can be modelized as follows: suppose I have two categories of packets Cp_i defining released packet category and unreleased packet category. I determine the affiliation probability of encoded packet p_i to a category using the Bayes rule presented [29] in the expression (8):

$$P(Cp_i|p_i) = \frac{P(p_i|Cp_i) * P(Cp_i)}{P(p_i)} \tag{8}$$

In the expression (8):

- $P(Cp_i|p_i)$ is *a posteriori* affiliation probability of a packet p_i to the category Cp_i.
- $P(p_i|Cp_i)$ defines the probability that for a given category, the fragments of the packet p_i

are added to Cp_i

- $P(Cp_i)$ is the probability that associates the encoded packet p_i with the category Cp_i.
- $P(p_i)$ is the likelihood of observing the packet p_i.

Determining $P(Cp_i|p_i)$ returns to determine $P(p_i|Cp_i) * P(Cp_i)$. With encoding LT, each encoded packet p_i is a combination of d fragments. So, determining $P(p_i|Cp_i)$ returns to determine the affiliation of fragments f_i in each encoded packet p_i to a category. I obtain the following expression:

$$P(p_i|Cp_j) = P(Cp_j)\prod_{i=1}^{n} P(f_i|Cp_j), \qquad j = 1,2 \tag{9}$$

Following the collection of the received encoded data and using the Bayesian model based on the conditional probability, each CH must anticipate the last encoded packet received to acknowledge the source and stop the generation process [30]. Let c_j be the probability of the input symbols which are not discovered at the iteration j. It is expressed as follows:

$$c_j = 1 - \sum_{i=1}^{j} P(f_i|Cp_k), \qquad k = 1,2 \tag{10}$$

The generation of the encoded packets stops when there are no undecoded input symbols. The main objective is to be able to estimate the minimum average number of encoded packets needed to decode the primary message. The optimal number m is the solution of the following optimization problem:

$$\begin{aligned} &\underset{m}{\operatorname{argmax}} \sum_{i=1}^{m} P(Cp_j|p_i), \qquad j = 1,2 \\ &s.t. \qquad\qquad c_i = 0 \end{aligned} \tag{11}$$

Considering the set of possible actions $A = \{a_1, \ldots, a_n\}$ defining the actions to perform. Typically, actions consist of assigning a packet to a specific category and, therefore, the number of actions is equal to the number of categories. In my case, since I have two categories, Cp_1 and Cp_2, the actions consist of either attribute the encoded packet p_i to one of two categories or not (respectively action a_1 and a_2). Bayes decision takes the following form:

$$D_{Bayes} = \begin{cases} 1 & if \quad P(Cp_i|p_i) > 1/2 \\ 0 & \qquad Otherwise \end{cases} \tag{12}$$

From the expression presented in (12), any packet with an assignment probability less than the predefined decision values, is placed into Cp2.

The necessary number of encoded packets to recuperate initial information, with K input symbols used in LT encoding, is [21,31]:

$$N = K + O(\sqrt{K}log^2K) \tag{13}$$

Thus, N can be rewritten as follows:

$$K \leq N \tag{14}$$

Based on the expression (14), I suppose the initial value of the generated packets is $n_{init} = K$.

Each CH checks whether the stop conditions are verified with this initial value; if so, it

Algorithm 1

```
for each p_i do
    Compute P(Cp_i|p_i)
    if P(Cp_i|p_i) ≥ 1/2 then
Cp_1 ← p_i
m=m+1
        if m ≥ K and c_j = 0 then
m_opt = m
        else
            Do not acknowledge the source
        end if
    else
Cp_2 ← p_i
    end if
end for
```

Fig. 2. Estimation on the number of encoded packets

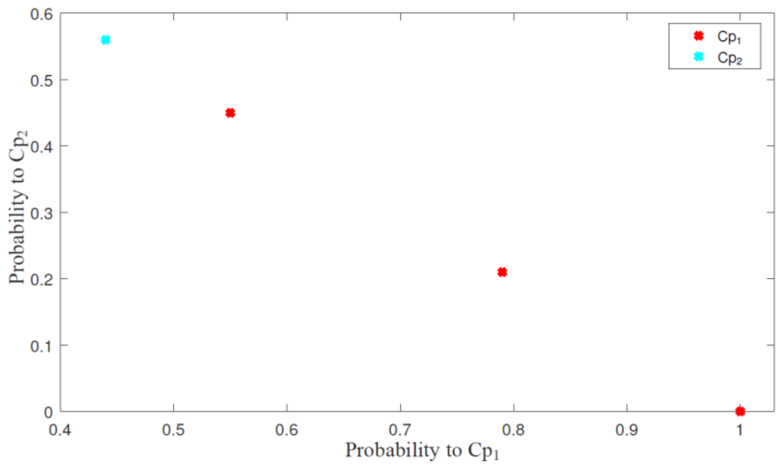

Fig. 3. Categories formation

acknowledges the source and the optimal number of packets is then $m_{opt} = n_{init}$.

This classifier works by calculating an assignment probability of a generated packet to one of two categories. Packet partially released (packets with degree greater than 1 after releasing some fragments) is supposed released and placed in the category Cp_1 of the *released packet* category. The distribution of packets degrees is the Robust Soliton Distribution (RSD).

The principle of the method of estimating the number of encoded packets needed for decoding is summarized in Algorithm (1) in Fig. 2. For each packet, I must collect data and update my knowledge to compute the posterior probability using the Bayes formula. By receiving n_{init} encoded packets, each CH checks the verification of stop conditions. If the CH does not receive the needed number of encoded packets, the source is still generating encoded packets.

Let us take the example of a set of encoded packets generated in the following way:

$p_1 = \{d = 1, f_1\}$

$p_2 = \{d = 3, f_1, f_2, f_3\}$

Table I. Classification probabilities of Encoded packets.

Encoded packets	Probability (Cp_1)	Probability (Cp_2)	Cp_1 or Cp_2
P_1	1	0	Cp_1
P_2	0.55	0.45	Cp_1
P_3	0.44	0.56	Cp_2
P_4	0.79	0.21	Cp_1

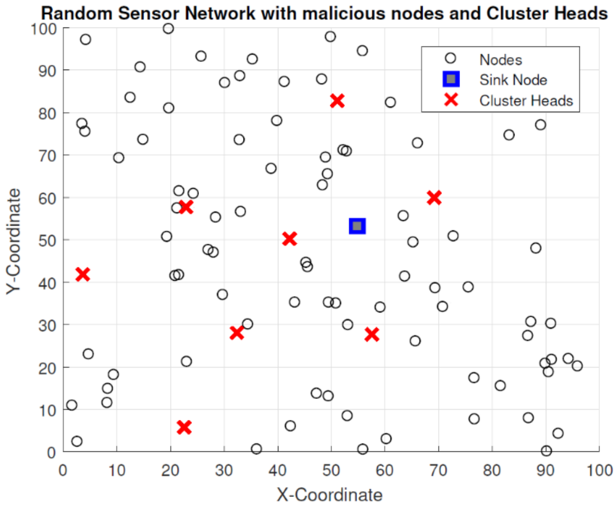

Fig. 4. distribution of nodes with the election of CH

$p_3 = \{d = 2, f_2, f_3\}$

$p_4 = \{d = 2, f_1, f_2\}$

I have assumed in this section that a packet with degree *1* is automatically added to the first category Cp_1. Fig. 3, illustrates the classification of different packets generated using the Bayesian method with their corresponding probabilities. The estimate of the posterior probability of an encoded packet p_i is related to the fragments that construct it and all the packets already generated.

As illustrated in Table I, the initial number of encoded packets is supposed to be equal to 3 input symbols (K = 3). The CH checks whether there are still undecoded input symbols (if $c_j = 0$). In this example, $c_2 \approx 0.2$, hence more encoded packets are needed, and p_4 is generated. At the 4th iteration, I have $c_4 = 0$ and so the required number of encoded packets is $m_{opt} = 4$.

V. SIMULATION AND RESULTS

In this section, I present different simulations and analyzes of results. For the simulation parameters, Table II lists their configuration details. By using the network topology with mobile nodes, I have different node's position as it is shown in Fig. 4 with the number of nodes varying between 80 and 200. Nodes' mobility is arbitrary, in all possible directions with waypoint mobility model [32,33] and different mobility levels. I performed simulations on the proposed approach using

Table II. Simulation parameters.

Parameters	Values
E_0	0.5 nJ/bit
E_{Aggr}	5 nJ/bit/message
Network area	100m × 100m
Weighting factors	a_1 =0.3, a_2 =0.4, a_3 =0.3
Number of nodes	80 - 200
Data packet size	4000 bits

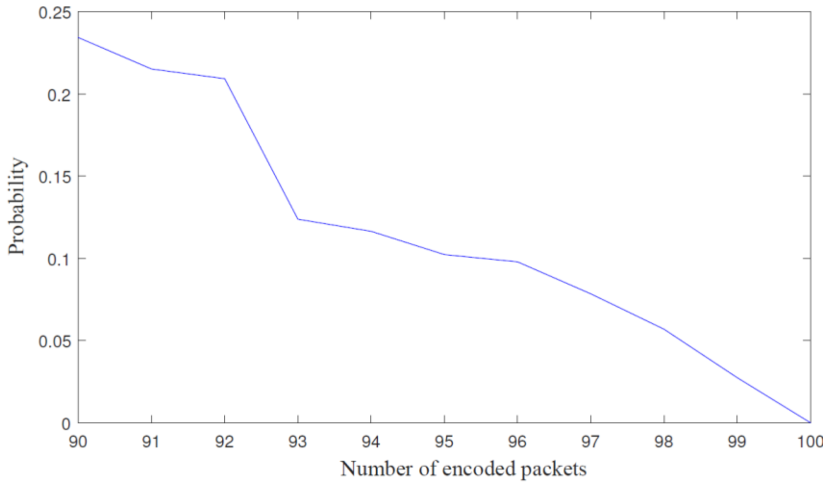

Fig. 5. Stop condition vs the number of generated packets

MATLAB R2018b. In a 100 * 100 m^2 network area, I considered several nodes, N varying between 80 and 200 nodes in the network, with a sink node and different elected CH as shown in Fig. 4.

For my simulation, I have considered the architecture presented in Fig. 1 that nodes are directly connected to the elected CH based on the clustering algorithm.

The adopted clustering algorithm developed in [34] contains two kinds of nodes: sink nodes (static node) and other nodes (with energy restriction and different speed mobility). The pause time is the duration when a node stays in the stationary state. For the weighting factors, they should be chosen depending on the importance of each parameter used to determine the weight of each node. It is possible to affect the biggest value to the most important metric, and it is possible to assign the same value for each parameter if all of them are considered to have the same importance.

Fig. 5 shows the variation of the stop condition. Indeed, the verification of the states to stop the generation of encoded packets is started at n_{init} = K = 80. In this figure, the probability decreases until it equals 0 with a necessary number of packets around 100. This means that for maximum decoding, at least 100 encoded packets from the sender are required for decoding of 80 packets. According to the degree distribution presented in Fig. 6, where I have a high degree and low degree values to guarantee success decoding, I classify encoded packets into the defined categories Cp_1 or Cp_2. Fig. 6

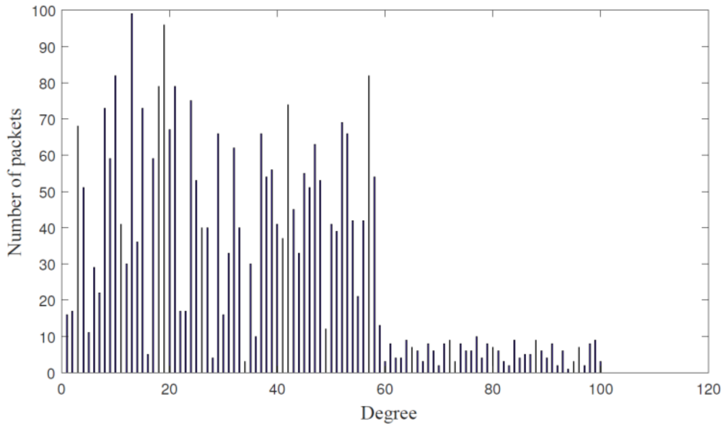

Fig. 6. Degree distribution of different packets

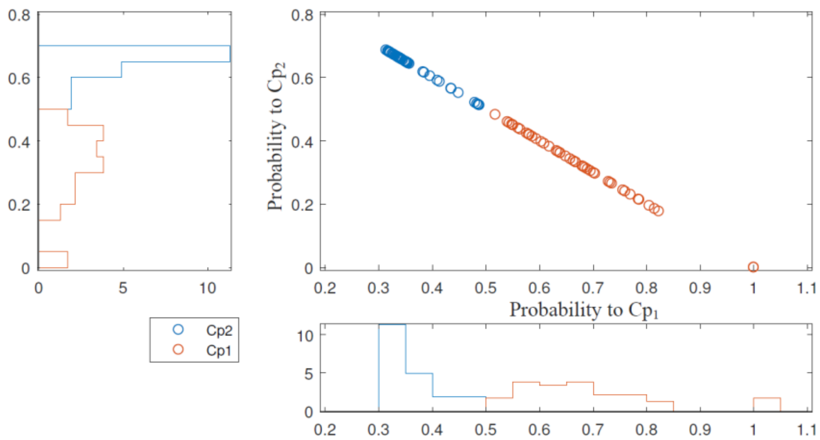

Fig. 7. Marginal distribution and classification of the encoded packet into Cp_1 and Cp_2

shows how many packets have what degree of distribution. In Fig. 7, encoded packets are classified into Cp_1 or Cp_2 based on the Bayesian method. The highest probabilities are affected by the first category of released packets Cp_1.

Since the probability of successful decoding depends on K value and also this dependence on the overhead of the sent encoded packets is very high, so for this purpose, a comparison between the simulation results and the theory mentioned above has been performed for this parameter.

Figs. 8 and 9 show the margin of difference between the values obtained according to theoretical study and simulations in two cases for $K \approx 80$ and $K \approx 190$. These two figures illustrate the efficiency of the estimates made to determine the number of packets necessary for the success of the decoding. Indeed, with a number of fragments around $K = 190$, it is essential to receive $N \approx 200$ packets to recover initial information. Using the estimation so that it is possible to determine the needed number of encoded packets, a vital minimization of energy consumption was obtained. This means that as the number of encoded packets transmitted in each step of the data transmission increases, the probability of decoding the sent packets increases, as shown in Fig. 8. When about 10 more coded packets are

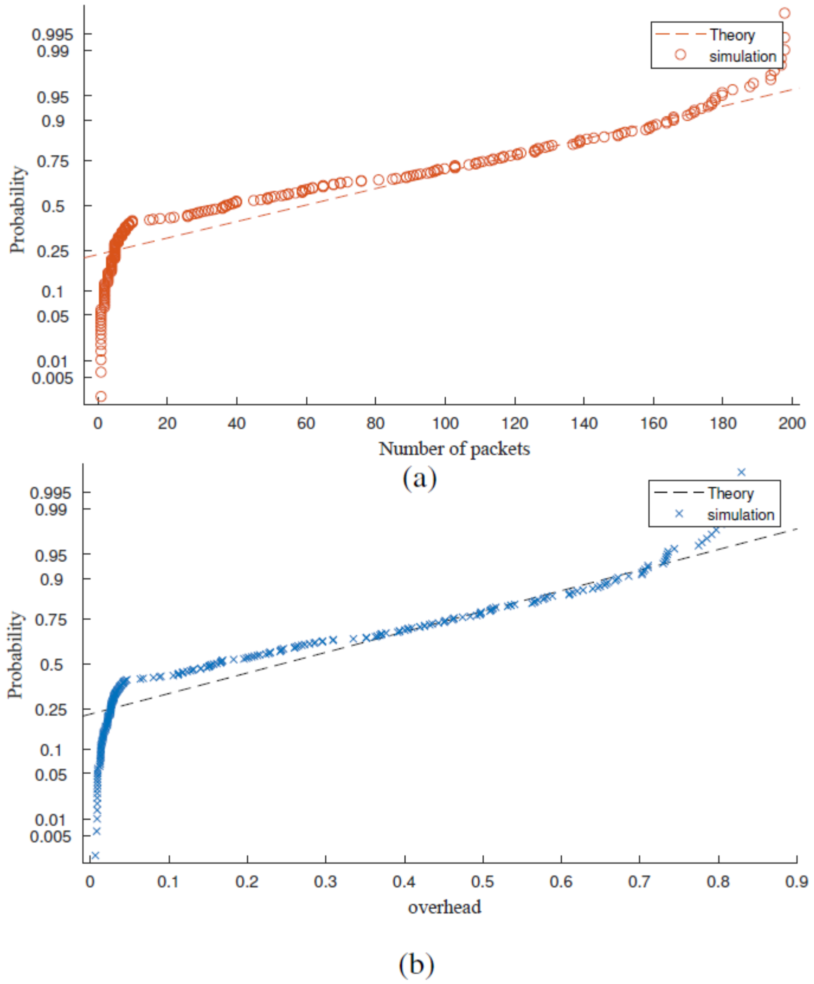

Fig. 8. (a) Comparison between simulations and theoretical number of encoded packets for K = 190, (b) Marginal distribution of the overhead for K = 190

sent, the probability of decoding the transmitted packets will increase to about 100%. Part *b* also clearly shows the percentage of these packets sent for this example. Therefore, to increase the probability of successful decoding according to the length of the data packet, it is necessary to determine the exact amount of overhead.

The generation of encoded packets is stopped based on the stopping condition presented in Fig. 5, with K = 80, it is possible to recover initial information with about N ≈ 100 encoded packets. In general, overhead refers to the extra cost in terms of time, memory, bandwidth, or any other resources used to execute instructions. More specifically, and in the case of LT codes, overhead refers to the overflow or the amount of additional encoded packets so that it is possible to decode successfully. Decoding is possible with the reception of $N = K\,(1 + \varepsilon)$, ε represents the overhead. $\varepsilon = 1$ indicates that decoding is possible with a number of additional encoded packets equal to the input symbols. Plus, the value of the overhead is weak; the better the decoding is. Indeed, the efficiency of the technique used is related to the amount of overflow observed.

The direct relationship between each fusion center (FC) or each CH and the member nodes allows

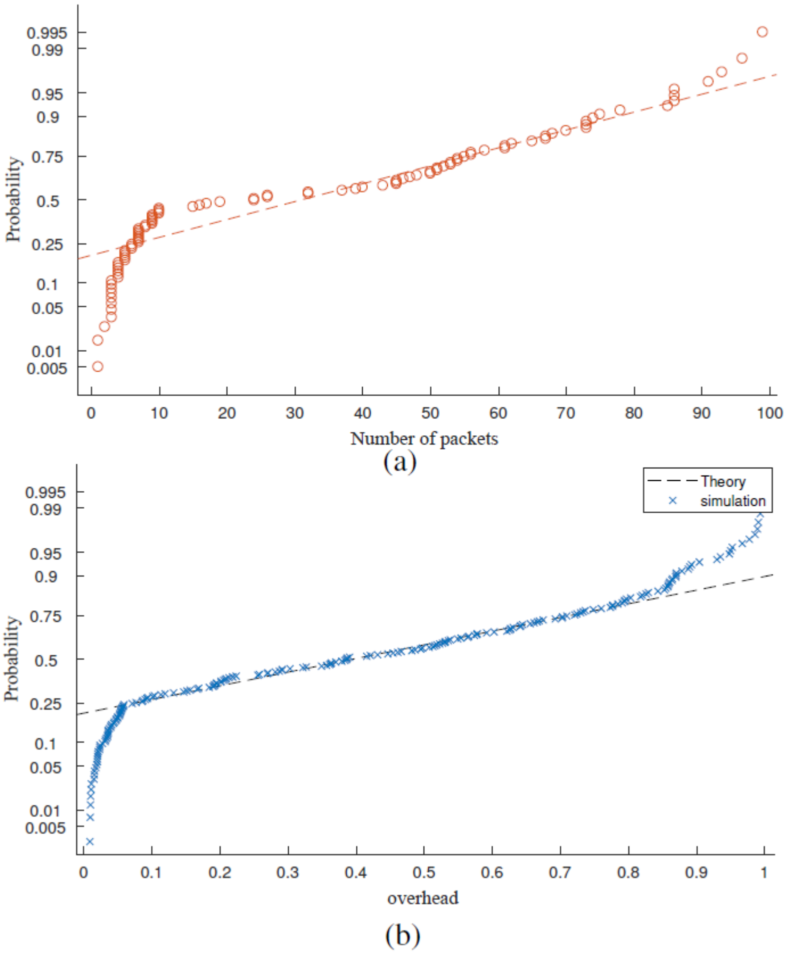

Fig. 9. (a) Comparison between simulations and theoretical number of encoded packets for K = 80, (b) Marginal distribution of the overhead for K = 80

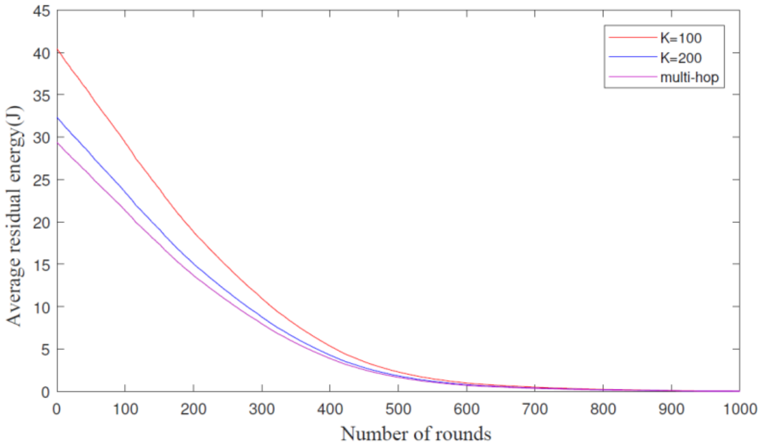

Fig. 10 Average residual energy

to minimize this overflow due to multi-hops and to minimize the energy consumed. The overhead represents the difference between the theoretical approach proposed and the results obtained following the simulations. The lower this value, the more the two results are closer. As it is shown in Fig. 10, and compared with the multi-hop situation, the average residual energy with the proposed approach is more important. The amount of residual energy due to multi-hop transmission is much greater than that of the two cases used. Also, Fig. 8 indicates the preserved energy following transmissions and reception does not exceed 28% compared to the case of the proposed estimation model where the energy reaches 32% and 40% for the two cases of simulations. I demonstrate that when fountain codes are provided with the assistance of the Bayesian method, their ability to determine the needed number of encoded packets significantly improves accurately. As a result, a significant amount of the energy consumed can be preserved, and the average residual energy after transmission and reception can be conserved.

The total energy is assumed to be equal to 1^J. The quantity of the residual energy after any communication makes it possible to judge the lifetime of the sensor node and the efficiency of the proposed algorithm. More residual energy means more efficiency. Based on the proposed algorithm, by ensuring the stability of the nodes and exploiting the Bayesian method to deduce the probability of decoding packet, an important amount of the energy consumed can be preserved.

VI. CONCLUSION

To minimize energy consumption as well as the impact of the overflow caused by the use of fountain codes, two objectives were focused on: (*i*) data collection at each elected CH and classification of the encoded packet and (*ii*) estimation of data in each cluster head. Indeed, each elected CH must anticipate the number of encoded packets collected. This anticipation aims at acknowledging the source and stopping the process of generating these data whenever their reception is enough. Naive Bayes classifier based on the probabilistic approach determines the number of encoded packets provided by nodes. It focuses on updating the probability estimate using the Bayes rule and deduces the probability of an event from other events already evaluated called prior. Each CH estimates the required data, sends it to the sink. By exploiting the adopted clustering architecture and estimation model to estimate the required number of encoded packets, I tied to check the needed packet and minimize the energy overconsumption caused by the useless data generated. Simulations show the efficiency of the proposed approach in terms of residual energy. As future work and to further characterize the benefits of the proposed method and explore its potential, I plan to compare this approach with a variety of algorithms.

REFERENCES

[1] B. Rashid and M. H. Rehmani, “Applications of wireless sensor networks for urban areas: A survey,” *Journal of network and computer applications*, vol. 60, pp. 192-219, Jan. 2016.

[2] S. Kumar, F. Zhao, and D. Shepherd, “Collaborative signal and information processing in microsensor networks,” *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 13-14, March 2002.

[3] E. Hossain, “First quarter 2016 IEEE communications surveys and tutorials,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 1-7, 1st Quarter 2016.

[4] M. Tubaishat and S. Madria, “Sensor networks: an overview,” *IEEE potentials,* vol. 22, no. 2, pp. 20-23, April-May 2003.

[5] I. Al-Anbagi, M. Erol-Kantarci, and H. T. Mouftah, “A survey on cross-layer quality of service approaches in WSNs for delay and reliability-aware applications,” IEEE Communications Surveys & Tutorials, vol. 18, no. 1, pp. 525-552, 1st quarter 2014.

[6] A. H. Mohajerzadeh, M. H. Yaghmaee, and V. Fakoor, “Total data collection algorithm based on estimation model for wireless sensor network,” *Wireless Personal Communications*, vol. 81, no. 2, pp. 745-778, 2015.

[7] T. C. Aysal and K. E. Barner, “Constrained decentralized estimation over noisy channels for sensor networks,” *IEEE Transactions on Signal Processing*, vol. 56, no. 4, pp. 1398-1410, April 2008.

[8] Y. Huang and Y. Hua, “On energy for progressive and consensus estimation in multihop sensor networks,” *IEEE Transactions on Signal Processing*, vol. 59, no. 8, pp. 3863-3875, Aug. 2011.

[9] W. Lam and A. Reibman, “Quantizer design for decentralized systems with communication constraints,” *IEEE Trans. Communications,* vol. 41, pp. 1602-1605, 1993.

[10] Z.-Q. Luo, “Universal decentralized estimation in a bandwidth constrained sensor network,” *IEEE Trans. Information Theory*, vol. 51, no. 6, pp. 2210-2219, June 2005.

[11] F. Belabed and R. Bouallegue, “A comparative analysis of machine learning classification approaches for fountain data estimation in wireless sensor networks,” *in IEEE 2019 15th Intern. Wireless Communications & Mobile Computing Conference (IWCMC)*, 2019, pp. 1251-1254.

[12] A. Argyriou and O. Alay, “Distributed estimation in wireless sensor networks with an interference canceling fusion center,” *IEEE Trans. Wireless Communications*, vol. 15, no. 3, pp. 2205-2214, March 2015.

[13] W. Sun, X. Yuan, J. Wang, Q. Li, L. Chen, and D. Mu, “End-to-end data delivery reliability model for estimating and optimizing the link quality of industrial WSNs,” *IEEE Trans. Automation Science and Engineering*, vol. 15, no. 3, pp. 1127-1137, Aug. 2017.

[14] K. Miranda and V. Ramos, “Improving data aggregation in wireless sensor networks with time series estimation,” *IEEE Latin America Transactions*, vol. 14, no. 5, pp. 2425-2432, May 2016.

[15] X. Gong, X. Wang, J. Guo, A. Wang, D. Xu, N. An, X. Chen, D. Fang, and X. Zheng, “Dedv: A data collection method for mobile sink based on dynamic estimation of data value in WSN,” *in IEEE 2016 Intern. Conf. Networking and Network Applications (NaNA)*, 2016, pp. 77-83.

[16] R. Yarinezhad, and A. Sarabi, “MLCA: A Multi-Level Clustering Algorithm for Routing in Wireless Sensor Networks,” *Journal of Communication Engineering* vol. 8, no. 2, pp. 249-265, Summer-Autumn 2019.

[17] S. Vahabi, M. Lahabi, M. Eslaminejad, and S. E. Dashti, “Geographic and Clustering Routing for Energy Saving in Wireless Sensor Network with Pair of Node Groups,” *Journal of Communication Engineering* vol. 8, no.1, pp. 146-157, Winter-Spring 2019.

[18] S.M. Mirrezaei, “Towards systematic Luby transform codes: optimisation design over binary erasure channel,” *Electronics Letters*, vol. 56, no. 11, pp. 550-553, May 2020.

[19] S.M. Mirrezaei, K. Faez, and S. Yousefi, “Towards Fountain Codes. Part II: Belief Propagation Decoding” *Wireless personal communications,* vol. 77, no. 2, pp. 1563-1584, 2014.

[20] A. Apavatjrut, C. Goursaud, K. Ja_res-Runser, C. Comaniciu, and J.-M. Gorce, “Toward increasing packet diversity for relaying LT fountain codes in wireless sensor networks,” *IEEE Communications Letters*, vol. 15, no. 1, pp. 52-54, Jan. 2010.

[21] M. Luby, “Lt codes,” *in the 43rd Annual IEEE Symposium on Foundations of Computer Science*, 2002, *Proceedings IEEE*, 2002, pp. 271-280.

[22] M. Zhang, W. Lei, and X. Xie, “Combined degree distribution: A simple method to design the degree distribution of fountain codes,” *in 2013 IEEE Third Intern. Conf. on Information Science and Technology (ICIST)*, pp. 1089-1092.

[23] K.-K. Yen, Y.-C. Liao, C.-L. Chen, and H.-C. Chang, "Modified robust soliton distribution (MRSD) with improved ripple size for LT codes," *IEEE Communications Letters*, vol. 17, no. 5, pp. 976-979, May 2013.

[24] G. T. Chen, L. Cao, F. Zhao, H.-f. Zheng, and M. Pan, "Analysis of robust soliton distribution for LT code*," in 2012 IEEE 11th International Conference on Signal Processing*, vol. 2, pp. 1546-1549.

[25] D. J. MacKay and D. J. Mac Kay, Information theory, inference and learning algorithms. Cambridge university press, 2003.

[26] H. Chen, "Performance-energy tradeoffs for decentralized estimation in a multihop sensor network," IEEE Sensors Journal, vol. 10, no. 8, pp. 1304-1310, 2010.

[27] S. Wesolowski, N. Klco, R. Furnstahl, D. Phillips, and A. Thapaliya, "Bayesian parameter estimation for effective field theories," *Journal of Physics G: Nuclear and Particle Physics*, vol. 43, no. 7, p. 074001, 2016.

[28] M. Cai, N. W. Schuck, J. W. Pillow, and Y. Niv, "A bayesian method for reducing bias in neural representational similarity analysis," *in Advances in Neural Information Processing Systems*, 2016, pp. 4951-4959.

[29] A. R. Solow, "A simple bayesian method of inferring extinction: comment," *Ecology*, vol. 97, no. 3, pp. 796-798, 2016.

[30] N. J. Gordon, D. J. Salmond, and A. F. Smith, "Novel approach to nonlinear/nongaussian bayesian state estimation," *IEE proceedings F (radar and signal processing),* vol. 140, no. 2, pp. 107-113, April 1993,

[31] L. Suo, G. Zhang, J. Lv, and X. Tian, "Performance analysis for finite length LT codes via classical probability evaluation," *IEEE Communications Letters*, vol. 21, no. 9, pp. 1957-1960, Sept. 2017.

[32] L. Enciso, P. Quezada, J. Fernandez, B. Figueroa, and V. Espinoza, "Analysis of performance of the routing protocols ad hoc using random waypoint mobility model applied to an urban environment," *12th Intern. Conf. on Web Information Systems and Technologie, WEBIST* 2016, pp. 208-213.

[33] A. Pramanik, B. Choudhury, T. S. Choudhury, W. Arif, and J. Mehedi, "Behavioral study of random waypoint mobility model based energy aware manet," *in 2016 IEEE 3rd Intern. Conf. on Signal Processing and Integrated Networks (SPIN),* 2016, pp. 624-629.

[34] F. Belabed and R. Boouallegue, "Clustering approach using node mobility in wireless sensor networks," *in 2017 IEEE 13th Intern. Wireless Communications and Mobile Computing Conf. (IWCMC)*, 2017, pp. 987-992.