A Markov Chain Model for the Decoding Probability of Sparse Network Coding

Pablo Garrido[†], Daniel E. Lucani[$], Ramón Agüero[†]

[†] University of Cantabria

Santander, Spain

{pgarrido,ramon}@tlmat.unican.es

[$] Department of Electronic Systems

Aalborg University, Denmark

del@es.aau.dk

Abstract

Random Linear Network Coding (RLNC) has been shown to offer an efficient communication scheme, leveraging a remarkable robustness against packet losses. However, it suffers from a high computational complexity, and some novel approaches, which follow the same idea, have been recently proposed. One of such solutions is Sparse Network Coding (SNC), where only few packets are combined in each transmission. The amount of data packets to be combined can be set from a density parameter/distribution, which could be eventually adapted. In this work we present an semi-analytical model that captures the performance of SNC on an accurate way. We exploit an absorbing Markov process where the states are defined by the number of useful packets received by the decoder, i.e the decoding matrix rank, and the number of non-zero columns at such matrix. The model is validated by means of a thorough simulation campaign, and the difference between model and simulation is negligible. We also include in the comparison some more general bounds that have been recently used, showing that their accuracy is rather poor. The proposed model would enable a more precise assessment of the behavior of SNC techniques.

Index Terms

Random Codes; Sparse Matrices; Network Coding; Absorbing Markov Chain

I. Introduction

Network Coding (NC) techniques foster a new communication paradigm, where packets are no longer immutable entities and nodes across the network could retransmit, discard or recode them. Among these techniques, Random Linear Network Coding (RLNC) stands as one of the most interesting solutions, since it provides robustness against packet losses. On the other hand, some questions have been raised about the decoding complexity of RLNC.

In order to reduce such complexity, the authors of [1] promoted Sparse Network Coding (SNC) techniques. Afterwards, a Tunable Sparse Network Coding (TSNC) scheme was introduced by Feizi *et al.* in [2], which, in a nutshell, proposed tuning the density of the coded packets, as they are being generated by the source during the transmission. There is a trade-off between the reduction of the computational complexity and the performance degradation induced by the corresponding overhead. Hence, it would be really helpful if the optimum density configuration could be found. However, there was not an appropriate model for sparse coding techniques, and only some approximate bounds have been used. These bounds aimed to be applied in a large number of cases, and focused on a single dimension: the degrees of freedom, i.e the decoding matrix rank, as the unique piece of the relevant information. As will be seen later, their accuracy is quite poor.

In this paper we propose a complete semi-analytical model for SNC techniques. We include a second dimension, the covered packets, i.e the non-zero columns at the decoding matrix. It is based on an Absorbing Markov Chain and it precisely mimics the probability of generating new information when sparse coding schemes are used. To our best knowledge, there is not any similar proposal in the related literature. We shall later see that the accuracy of the proposed model is very high.

The rest of the paper is structured as follows: Section II summarizes the operation principles of NC techniques and recalls some of the bounds that have been used in previous works to estimate the performance of SNC. In Section III we describe the proposed model and we exploit the properties of Absorbing Markov Chains to assess the performance of SNC techniques. Finally, in Section IV we validate our proposal, by means of an extensive simulation campaign. We also compare its performance with that exhibited by some of the bounds that have been previously used in the related literature. We finally conclude the paper in Section V, highlighting its most relevant contributions and advocating some aspects that will be addressed in our future work, by exploiting the proposed model.

II. From Random Linear Network Coding to Tunable Sparse Network Coding

A. *RLNC*

NC techniques were originally proposed by Ahlswede *et al.* in their seminal paper [3], where the *store and forward* approach was questioned; they also proved that the use of a coding scheme yields the maximum multicast capacity. Some subsequent works by Koetter and Medard [4], and Li *et al.* [5] broadened that idea, proposing the use of linear codes, and Ho *et al.* [6] presented a randomized network coding approach that achieved the maximum multicast capacity with high probability, advocating the RLNC scheme. Since those initial works, we have witnessed an increasing interest on potential applications of NC.

Many works have studied the benefits of NC techniques. Katty *et al.* [7] and Chachulski *et al.* [8] were some of the first ones proposing actual protocols, COPE and MAC-independent Opportunistic Routing & Encoding (MORE), respectively. Each of them represents a different NC flavor: *Inter-flow* and *Intra-flow*. In *Inter-flow* NC packets belonging to different information flows are combined. Although this approach has been thoroughly analyzed [7], [9]–[11], it exhibits some drawbacks if applied over realistic scenarios, as was shown in [12], [13].

On the other hand, *Intra-flow* NC techniques are based on the combination of packets belonging to the same flow; among them, RLNC scheme stands out as the most widespread solution, due to its simplicity and good performance. Indeed, it hides losses from the upper layers over point-to-point links [14], [15], reduces signalling overhead over opportunistic networks [8], and leverages efficient transmissions over wireless mesh networks [16]–[18].

There are also other coding solutions: LT [19] and Raptor Codes [20], which share some of the advantages of the RLNC scheme: (i) resiliency to packet losses, (ii) low overhead and (iii) suitability for heterogeneous networks and devices. However, they do not provide (i) on-the-fly coding/decoding [21], (ii) low delay, and (iii) recoding capabilities, which are considered to be some of the most relevant advantages of NC. Opposed to fountain codes, the encoder can generate coded packets as they arrive from the upper layers, and does not need to wait until the whole generation has been already received. The decoder can also start to decode packets as they are received, thus reducing the delay of fountain codes. Moreover, recoding has been shown to offer a more robust behavior over error prone links and for multicast transmissions, compared to legacy routing approaches, where intermediate nodes just store-and-forward the received packets. Some of these benefits are discussed in [16], [17].

On the other hand, the main argument questioning the use of RLNC is their decoding complexity, $O(k^3)$, where k is the number of packets to decode, which is considerably higher than other approaches (for instance, LT or Raptor Codes). The coding throughput, defined as the rate at which coding is carried out, is severely impacted by the coding parameters (Galois Field size, $GF(2^q)$, and generation size, k). In general, greater values of these parameters would lead to lower coding throughput [22], [23]. Furthermore, network overhead, which is mainly due to the transmission of useless packets (linear dependent combinations) and the corresponding protocol header, is also affected by the coding parameters, as Heide *et al.* analyzed in [24], focusing on the field and generation sizes, and the code sparsity.

Several works have focused on reducing the coding and decoding complexity proposing different alternatives. One initial idea to separate the RLNC performance from the amount of transmitted data, was to divide the data into generations [25], chunks [26], segments [27] or groups [28] of k packets. More advanced schemes, advocating the combination of overlapping generations and sparse coding techniques, were proposed in [21], [29], [30]. Another approach, which exploits sparse coding scheme to reduce the complexity, was first proposed in [1]. Afterwards, Heide *et al.* presented a sparse coding scheme with a non-random selection [24], which would allow different recoding mechanisms. Another example was the TSNC scheme, introduced by Feizi *et al.* [2], which fosters a dynamic increase of the coding density as long as the transmission evolves.

Feizi's work was later broadened in [31], where different recoding approaches and more complex tuning functions were proposed. Additionally, the authors of [32] presented a practical implementation of the TSNC approach. They used a lower bound to estimate the impact of the density on the overall overhead, but as will be seen later, such bound was not very accurate. A robust TSNC protocol was proposed in [33] for enhancing the reliability and speed of data gathering in smart grids.

Other alternate approaches to reduce both the complexity and the overhead of RLNC advocate the use of inner and outer codes, such as Fulcrum Codes [34] or BATS Codes [35]. Fulcrum Codes, which are suitable for heterogeneous devices, propose using different field sizes to decode the received packets. On the other hand, BATS Codes reduce the computational complexity by means of an outer code, based on a fountain coding scheme. An additional coding solution using inner and outer codes, as well as sparse techniques was presented in [36], which exploits the *Gamma* distribution for the code design.

B. Tunable Sparse Network Coding

As already mentioned, Tunable Sparse Network Coding was initially proposed by Feizi *et al.* [2], and was later broadened in [31], [32]. The main reasoning was that the complexity of the decoding process for traditional RLNC solutions is considerably higher than in other approaches, for instance LT or Raptor Codes, which exploit sparse coding techniques. LT or Raptor Codes propose a random distribution of the density, where sparse packets (built by the combination of a few original packets) are more likely to be sent.

On the other hand, the legacy RLNC scheme generates coded packets by randomly combining the original packets of the same generation, p_i (where $i = 1, \ldots, k$). The corresponding coefficients, c_i, are selected from a Galois Field, $GF(2^q)$, and the two required operations (sum and product) are as well defined over $GF(2^q)$:

$$p'_{\text{RLNC}} = \sum_{i=1}^{k} c_i \cdot p_i \tag{1}$$

Using a Sparse Coding Scheme only a set of w randomly selected packets, $W = \{p_{j_1}, p_{j_2}, \cdots, p_{j_w} | p_{j_k} \neq p_{j'_k}, \forall j_k \neq j'_k\}$, from the same generation, are combined to build a coded packet:

$$p'_{\text{TSNC}} = \sum_{i=1}^{w} c_i \cdot p_{j_i} \tag{2}$$

in this case the random coefficients, c_i, are selected from nonzero elements in the Galois Field $GF(2^q)$. The use of highly sparse coded packets (low w) would increase the throughput of the decoding operations [31]. On the other hand, it could also lead to a greater probability of transmitting linear dependent combinations, thus increasing the corresponding network overhead and jeopardizing the performance. In this sense, based on the observation that the probability of generating linear dependent packets is higher as the transmissions evolves, TSNC is proposed to tune the density throughout the transmission. Some works, e.g. [31], [32], have already shown a high reduction on the computational complexity, with a slight increase of the corresponding overhead.

Trullols *et al.* [37] derived, for the RLNC scheme, the exact decoding probability for a successful decoding event after the reception of a number of coded packets. Zhao *et al.* proposed in [10] a simplified modification, proposing a generalization of the original expression. And in [38] the model was broaden to include multiple sources transmitting the same information. All the existing works aiming to analytically characterize sparse coding schemes either lack accuracy or have some limitations. Li *et al.* showed in [39] that the singularity probability, i.e the probability for a random sparse matrix to have a full-rank, can be upper and lower bounded. However, the analysis was limited to rather large finite fields ($q \to \infty$) and results were obtained for very low generation sizes, $k \leq 5$. An extension of this work was presented in [40], where a more generic coefficient distribution was considered. On the other hand, Blomer *et al.* highlight in [41] the complexity of the problem we tackle in this work, establishing upper and lower bounds for the number of linear dependencies within a random sparse matrix. Although they do not explicitly exploit this result for coding purposes, their conclusions are still valid for our sparse coding model.

Regarding the use of such loose bounds for network coding solutions, the authors of [31], [32], [42] exploit a lower bound to find a trade-off between complexity and overhead. Such lower bound establishes that the probability of receiving a new linearly independent packet by the decoder, when it already has r linearly independent packets, is:

$$\text{Prob}_{r^+}^{\text{TSNC}}(k, d) \geq 1 - (1 - d)^{k-r} \tag{3}$$

where k is the generation size and $d = \frac{w}{k}$ is the corresponding coding density.

Following the approach proposed in [32], the encoder decides the most appropriate density in order to send packets with the lowest density, keeping the number of packets transmitted below a defined *budget*. The information available at the encoder is, in most cases, rather limited; in this case, the destination reports, with a dedicated control packet, the number of useful packets that have been already received. Hence, a more precise model for such probability might lead to the proposal of mechanisms to strongly improve the performance of TSNC.

Although it is not the main objective through this paper, it is also worth highlighting other open questions of RLNC and TSNC schemes. First, coding coefficients transmission could cause a considerable overhead. The authors of [24] evaluated how this overhead is affected by different coding parameters. In any case, the corresponding overhead is rather limited for the binary case ($q = 1$) or/and when coding operations are restricted to a generation of size k. For instance, the authors of [33] were able to add a reasonable overhead for 4000 data packets coming from 4000 different sources, leveraging an efficient solution, using the identifier of data packets. Note that the use of sparse coding techniques would as well lead to mode efficient coding vector representations. We could, for instance, represent each non-zero coefficient by an index-scalar pair. On the other hand, various proposals for recoding operations at intermediate nodes have been also published. The authors of [31] analyzed two different solutions (on a theoretical level), while [33] proposes a practical implementation of sparse coding techniques, with recoding at intermediate nodes, which is exploited to gather information over smart grid scenarios.

III. Markov Chain Model

In this Section we introduce a Markov Chain Model that mimics the behavior of SNC techniques. This model is valid from the receiver's perspective, i.e it is based on the status of the decoder, and it is therefore independent on the particular network that connects the source with the destination. We define a set of states (r, c), where r is the current rank of the decoding matrix and c is the number of non-zero columns. Clearly, $r \leq c$. Based on these states, we can establish a discrete Markov Chain, $\mathcal{S}_q(w, k)$. Figure 1 shows an illustrative example for such chain, with $w = 3$ and $k = 10$. As can be seen, $\mathcal{S}$ is an absorbing process, since the state (k, k) is absorbing, and will be eventually reached.

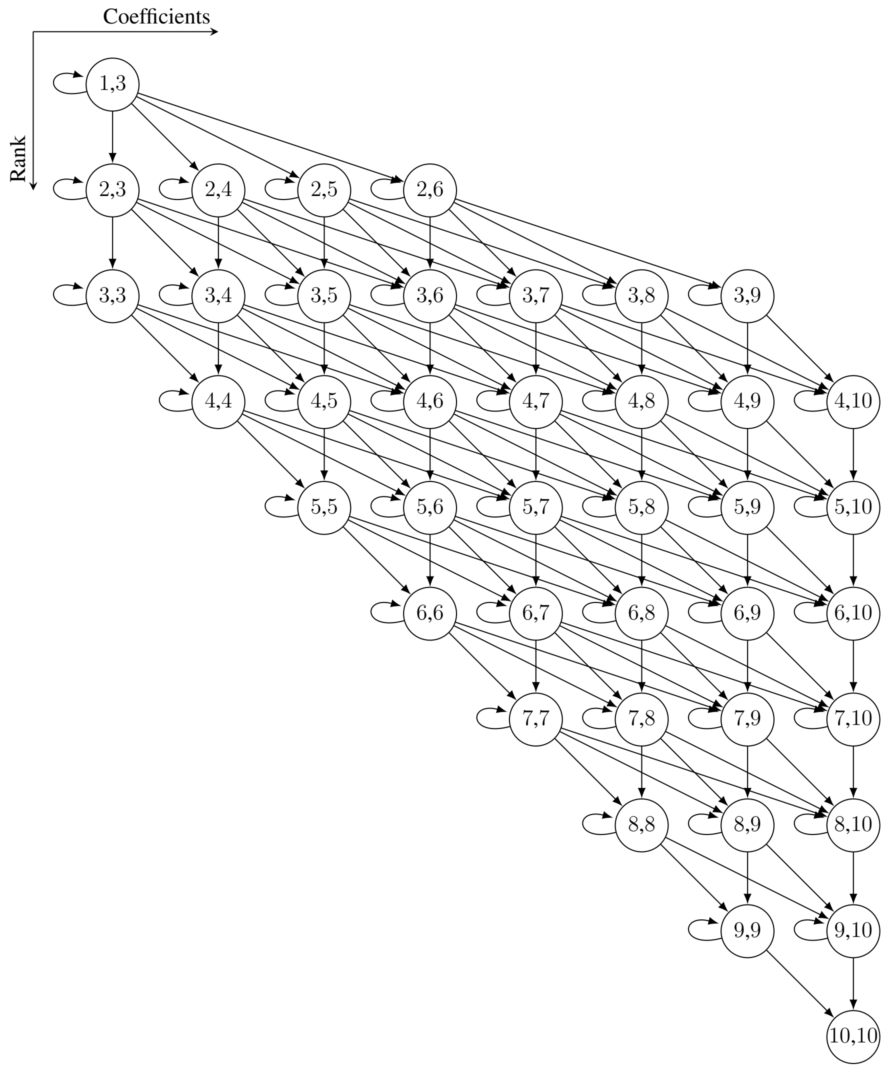


Fig. 1. Markov chain for $w = 3$ and $k = 10$. We assume that $q > 1$, otherwise, the states $(2, 3)$ and $(3, 3)$ would not be feasible.

Theorem 1. *(Transition Probabilities)*

The transition probability between states (r, c) and $(r+i, c+j)$, $p_{r,c}(i,j)$ are as follows:

$$\vartheta_q(r,c,w)\prod_{t=0}^{w-1}\frac{c-t}{k-t} \qquad \textit{if}\begin{cases} i=0 \\ j=0 \end{cases} \tag{4.1}$$

$$[1-\vartheta_q(r,c,w)]\prod_{t=0}^{w-1}\frac{c-t}{k-t} \qquad \textit{if}\begin{cases} i=1 \\ j=0 \end{cases} \tag{4.2}$$

$$\binom{w}{j}\frac{\prod\limits_{t=0}^{w-j-1}(c-t)\prod\limits_{t=c}^{c+j-1}(k-t)}{\prod\limits_{t=0}^{w-1}(k-t)} \qquad \textit{if}\begin{cases} i=1 \\ j=1\dots w \end{cases} \tag{4.3}$$

$$0 \qquad \textit{otherwise} \tag{4.4}$$

where $\vartheta_q(r,c,w)$ is the probability for a randomly generated sparse vector, with w non-zero elements, to be linearly dependent with the already received r vectors, from the c-dimensional space, i.e no new coefficients are used. Hence, c captures the coefficient distribution length for the already received packets. A Galois Field $GF(2^q)$ is assumed.

Proof: We use simple combinatorial mathematics to derive the transition probabilities of the absorbing Markov chain that was previously discussed.

For any (r, c) state, if the received packet includes any novel coefficient, the rank is always increased. The number of combinations that would change the state from (r, c) to $(r+1, c+j), \quad j > 0$ are:

$$\mathcal{C}_{(r,c)}(r+1, c+j) = \mathbb{C}^{c}_{w-j}\cdot\mathbb{C}^{k-c}_{j} = \binom{c}{w-j}\cdot\binom{k-c}{j} \tag{5}$$

where $\mathbb{C}^n_t$ is the combination of n elements taken t at a time without repetition.

In addition, the overall number of possible vectors are $\mathbb{C}_w^k$. Hence, the corresponding probability is:

$$p_{r,c}(1, j > 0) = \frac{\binom{c}{w-j} \cdot \binom{k-c}{j}}{\binom{k}{w}} = \frac{\frac{c!}{(c-w+j)!(w-j)!} \cdot \frac{(k-c)!}{(k-c-j)!j!}}{\frac{k!}{(k-w)!w!}} =$$

$$= \binom{w}{j} \cdot \frac{\prod_{t=0}^{w-j-1} (c-t) \cdot \prod_{t=c}^{c+j-1} (k-t)}{\prod_{t=0}^{w-1} (k-t)} \quad (6)$$

On the other hand, if the new packet does not include any new coefficient, the corresponding vector could be either linearly dependent or independent, and this is established by ϑ. The combinations that do not increase the number of already received coefficients is $\mathbb{C}_w^c$. Hence, the probability of staying at the current state, (r, c) is:

$$p_{r,c}(0,0) = \vartheta_q(r,c,w)\frac{\binom{c}{w}}{\binom{k}{w}} = \vartheta_q(r,c,w)\frac{\frac{c!}{(c-w)!w!}}{\frac{k!}{(k-w)!w!}} = \vartheta_q(r,c,w)\prod_{t=0}^{w-1}\frac{c-t}{k-t} \quad (7)$$

While the probability of going to $(r+1, c)$ can be calculated as follows:

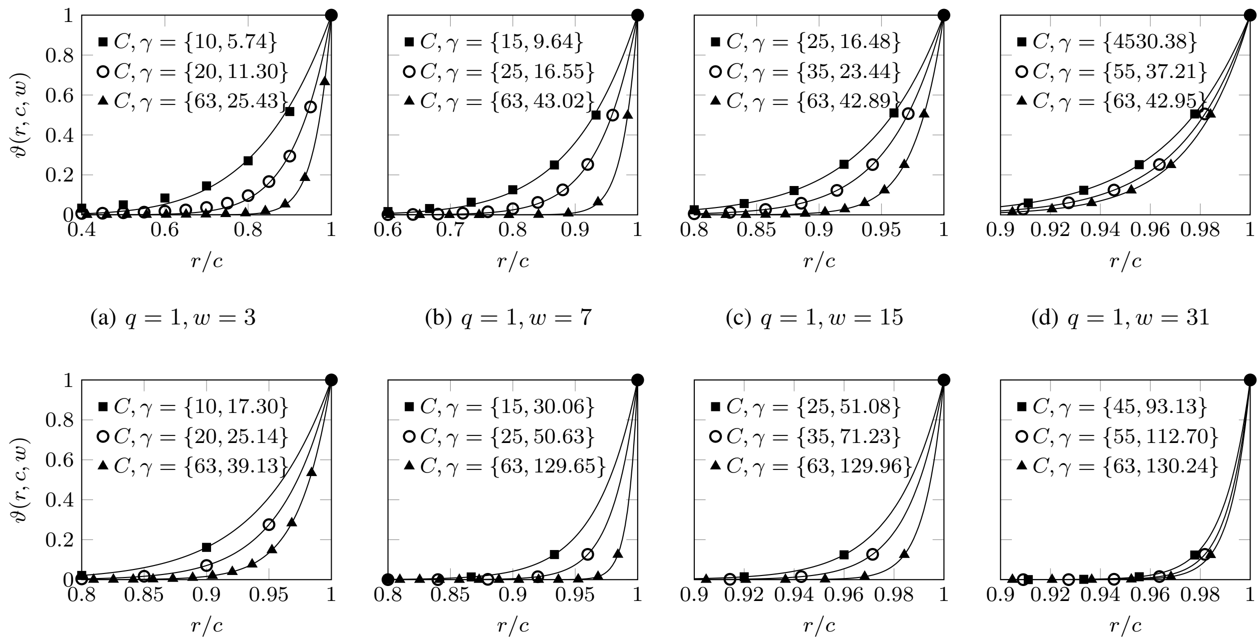
$$p_{r,c}(1,0) = [1 - \vartheta_q(r,c,w)]\frac{\binom{c}{w}}{\binom{k}{w}} = [1 - \vartheta_q(r,c,w)]\prod_{t=0}^{w-1}\frac{c-t}{k-t} \quad (8)$$

Whenever a packet is received the rank can only increase in one single unit, and thus, $p_{r,c}(i,j) = 0, \quad i > 1$. Likewise, the number of novel coefficients per packet cannot be larger than w, so $p_{r,c}(i,j) = 0, \quad j > w$.

■

A. *Empirical modeling of* $\vartheta_q(r,c,w)$

To the best of our knowledge there is not a closed expression for $\vartheta_q(r,c,w)$, so we conducted a Montecarlo analysis to empirically obtain it. We start by synthetically reaching every state of the corresponding *Markov* chain, and then we generated a new vector, enforcing that the w



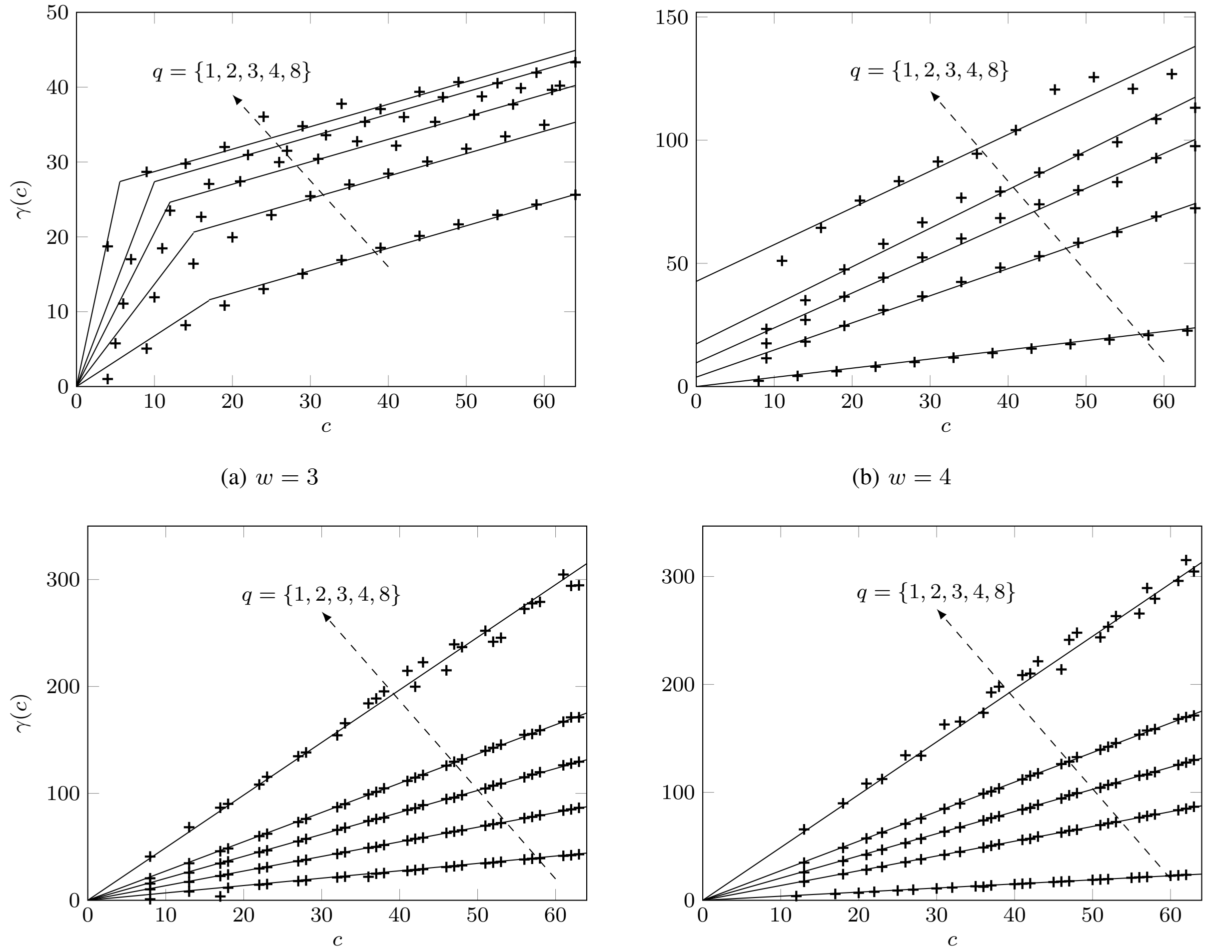
(a) $q = 1, w = 3$ (b) $q = 1, w = 7$ (c) $q = 1, w = 15$ (d) $q = 1, w = 31$

(e) $q = 3, w = 3$ (f) $q = 3, w = 7$ (g) $q = 3, w = 15$ (h) $q = 3, w = 31$

Fig. 2. $\vartheta(r, c)$ Vs. $\frac{r}{c}$. Markers correspond to the values obtained with the Montecarlo analysis, while the solid lines are the fitting curves $\left(\frac{r}{c}\right)^{\gamma}$.

components were only selected from the c already received coefficients. Afterwards we calculated the rank of the corresponding matrix to see whether it had increased; if that was the case the vector was linearly independent. We estimated the corresponding probability by counting the number of successes over a total of 100000 independent experiments. Note that ϑ is the probability for a generated vector to be linearly dependent and thus $1 - \vartheta$ corresponds to the probability for a generated vector to be linearly independent, in both cases knowing that the number of received coefficients remains the same, i.e. no novel coefficients are used.

Figure 2 shows how ϑ varies against the ratio $\frac{r}{c}$ for various combinations of q, w and c. In all cases we can use a function $\widetilde{\vartheta} = \left(\frac{r}{c}\right)^{\gamma}$ to approximate the observed behavior. Figure 2 also shows such fitting functions with a solid line, yielding a rather accurate approximation. This fitting is valid for all the possible combinations of $w \leq \frac{k}{2}$ and q, and for every c value (from 1



(a) $w = 3$ (b) $w = 4$

(c) $w_{\text{odd}} > 3$ (d) $w_{\text{even}} > 4$

Fig. 3. Fitting of $\gamma_q(c)$ function. Markers correspond to the values obtained with the Montecarlo analysis, while the solid lines are the fitting curves.

to k).

Figure 3 shows the evolution of γ against c for different w and q configurations. As can be seen on the lower figures, for $w > 3$, there is a clear linear relationship between γ and c, and the slope of the corresponding line only depends on the values of w and q. On the other hand, for $w = 3$ and $w = 4$, a different behavior was observed (see upper figures). In order not to increase the complexity of the model, we have approximated the behavior of $w = 3$ with two different

TABLE I

m AND c_0 FOR DIFFERENT q VALUES

<table>
<tr><td>q</td><td>1</td><td>2</td><td>3</td><td>4</td><td>8</td></tr>
<tr><td>m_{odd}</td><td>0.676</td><td rowspan="2">1.367</td><td rowspan="2">2.055</td><td rowspan="2">2.738</td><td rowspan="2">4.891</td></tr>
<tr><td>m_{even}</td><td>0.337</td></tr>
<tr><td>c_0</td><td>17</td><td>15</td><td>12</td><td>10</td><td>6</td></tr>
<tr><td>m_4</td><td>0.337</td><td>1.101</td><td>1.417</td><td>1.565</td><td>1.491</td></tr>
<tr><td>b_4</td><td>0</td><td>3.817</td><td>9.627</td><td>17.298</td><td>42.634</td></tr>
</table>

lines, since it can be seen that, for lager c's, the slope of the corresponding function does not depend on q, being all of them parallel. As will be seen later, the results are rather accurate, despite there is a non-negligible difference between the observed values and the corresponding fitting.

With all of the above into account, we can use the following functions to estimate the value of γ:

$$\begin{cases} \gamma = m_{\text{odd}} \cdot c & c < c_0 \\ \gamma = 0.3\left[c - c_0\left(1 - m_{\text{odd}}\right)\right] & c \geq c_0 \end{cases} \qquad w = 3 \tag{9}$$

$$\gamma = m_4 \cdot c + b_4 \qquad w = 4 \tag{10}$$

$$\gamma = m \cdot c \qquad 4 < w < k/2 \tag{11}$$

where the value of the slope m depends on q, and c_0 is the point where the slope of γ changes for $w = 3$. The corresponding values of m and c_0 for the different q's are given in Table I

B. Fundamental Matrix

Once we have established all the transition probabilities, we can build the fundamental matrix for the Absorbing Markov Chain. According to [43], the canonical form, for t and r transient and absorbing states, respectively, of such matrix, can be defined as follows:

$$P = \begin{bmatrix} I_{r\times r} & 0 \\ R_{t\times r} & Q_{t\times t} \end{bmatrix} \tag{12}$$

Since there is only one absorbing state (k, k), I is an identity matrix of one single element and R is a column vector with the transition probabilities of all the remaining states to (k, k). Finally Q is a matrix with the transition probabilities between the transient states. We will assume that the first row/column of this matrix correspond to the initial state, i.e. $(1, w)$.

Theorem 2. *Average number of transitions (Theorem 3.2.4 in [43]). The average number of transitions before being absorbed, when starting in a transient state i, is the i^{th} element of the column vector*

$$M = (I - Q)^{-1}\,\Gamma \tag{13}$$

where I is an identity matrix with the same dimension as Q, and Γ is an all-one column vector. Furthermore, the matrix $N = (I - Q)^{-1}$ is called the fundamental matrix *for P.*

Corollary 3. *Average number of transmissions. The average number of transitions defines the average number of transmissions, since in our case, a transition always corresponds to a packet transmission, and the first element of M would correspond to the number of transitions that are required to hit the (k, k) state from the initial one $(1, w)$.*

Theorem 4. *Probability of being in state j after T transitions (Theorem 3.1.1 in [43]). The entry $p_{ij}^{(T)}$ of P^T is the probability of being in state j after T transitions, provided that the chain was started in i*

Corollary 5. *Probability of succesfully decode a generation. Since the chain always starts from the initial state $1 \to (1, w)$ and there is only one absorbing state, $t + r \to (k, k)$, the probability* *of successfully decode a generation after #TX transmissions is:* $\xi(\#TX) = p_{1,\boldsymbol{t+r}}^{(\#TX)}$

Another parameter of interest would be the probability of increasing the rank of the corresponding decoding matrix with every transmission. This would allow establishing dynamic tuning schemes for the coding density, since lower densities would yield lower coding/decoding times, but they might as well lead to a higher number of transmissions.

Theorem 6. *Transient Probabilities (Theorem 3.5.7 in [43]). The probability of visiting a state* j*, when starting a transient state* i*, is the* (i, j) *entry of the transient probabilities matrix* H*:*

$$H = (N - I) \cdot N_d^{-1} \tag{14}$$

where N_d *is a diagonal matrix with the same diagonal of* N*.*

Corollary 7. *Probability of receiving a linearly independent packet. We define a set of states* $\mathbf{s}(r)$ *as all the states from the chain where the rank equals* r*,* $\mathbf{s}(r) = \{(i, j) \in \mathbf{S} \, | i = r\}$ *. Hence, the probability of increasing the rank of the matrix when* r *independent packets have been already received can be calculated as follows:*

$$\delta(r) = \sum_{\forall j \, |(r,j) \in \mathbf{S}} H(1, n_j) \cdot (1 - p_{r,j}(0, 0)) \tag{15}$$

where n_j *is the index corresponding to the* (r, j) *state.*

C. Impact of errors

So far we have assumed an ideal wireless channel between the transmitter and the receiver. However the model can be easily broadened so as to consider packet-erasure links. For that we just need to modify the corresponding transition probabilities as follows:

$$\widetilde{p_{r,c}}(i,j) = \begin{cases} p_{r,c}(i,j)\,(1-\alpha) & (i,j) \neq (0,0) \\ p_{r,c}(i,j)\,(1-\alpha) + \alpha & (i,j) = (0,0) \end{cases} \tag{16}$$

where α is the frame error rate of the wireless link.

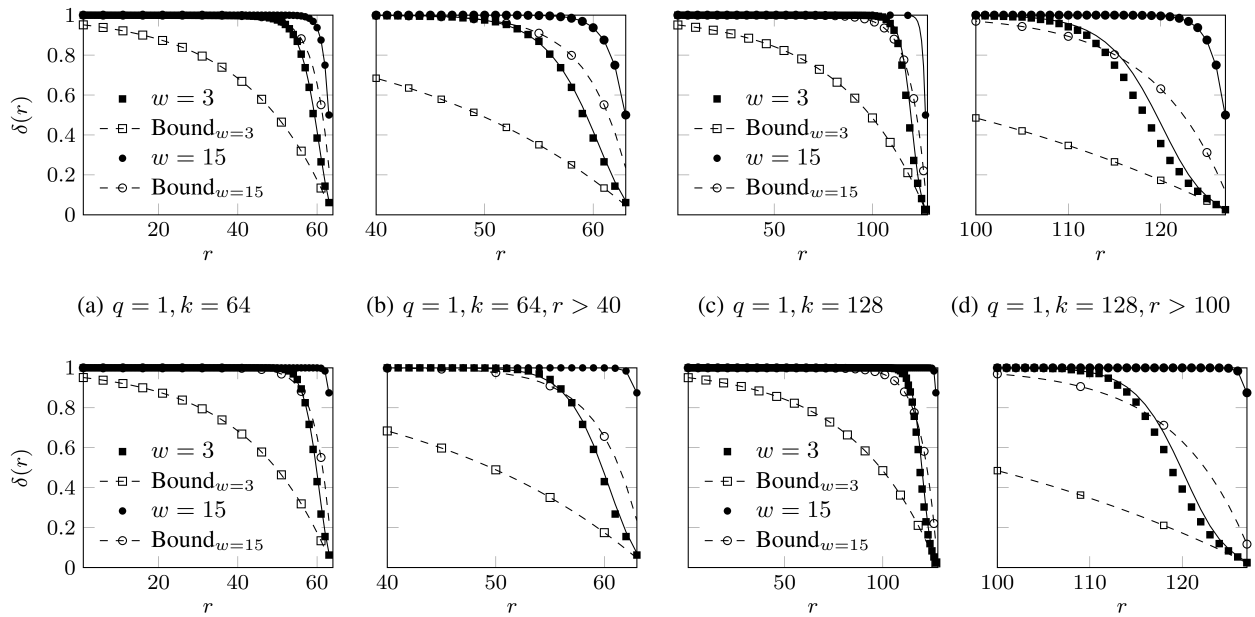
We can easily see that $\widetilde{Q} = (1-\alpha)\,Q + \alpha I$, and hence:

$$\widetilde{M} = \left(I - \widetilde{Q}\right)^{-1}\Gamma = \left[I - ((1-\alpha)\,Q + \alpha I)\right]^{-1} = \frac{(I-Q)^{-1}}{1-\alpha} \tag{17}$$

IV. Simulation and Model Validation

In this Section we assess the validity of the proposed model and all the results that were previously discussed by means of an extensive simulation campaign. We use the M4RIE library [44] to transmit (using TSNC) 10000 different generations, in order to ensure statistical tightness of the corresponding results.

Figure 4 shows the probability of receiving a linearly independent packet against the current rank at the receiver, (15). As can be seen, the probability is close to 1 until the rank is considerable large. Afterwards, it decreases quite sharply, especially for low values of q and w. We can also see the small difference between the simulation results (solid line) and the proposed model (markers). The model shows a good accuracy, being the mean squared error in the worst case ($k = 128$, $q = 3$) $3.14 \cdot 10^{-4}$. On the other hand, the lower bound that was discussed in Section II, which has been used in various works until now, yields a much lower probability, exhibiting very little accuracy. Since the corresponding operations that need to be performed are much faster if the coding density (w) is low, this result is really interesting, since it shows that there is not any disadvantage in using a low w until the rank at the receiver is quite high; at that moment increasing w would probably improve the corresponding performance. It is worth noting that from now on, all results have been obtained for the binary case, $GF(2)$.



(a) $q = 1, k = 64$ (b) $q = 1, k = 64, r > 40$ (c) $q = 1, k = 128$ (d) $q = 1, k = 128, r > 100$

(e) $q = 3, k = 64$ (f) $q = 3, k = 64, r > 40$ (g) $q = 3, k = 128$ (h) $q = 3, k = 128, r > 100$

Fig. 4. Probability of increasing rank Vs. current rank at the decoder for different field sizes ($q = 1, 3$) and generation sizes ($k = 64, 128$). Markers show the values obtained with the proposed model and the solid lines correspond to the simulation results

Another interesting result would be the average number of transmissions that would be required to correctly receive a generation. Table II collects such metric for different configurations, in which we have modified the value of both w and k. The theoretical result (*Model*) is obtained using (13). The simulation result (*Simul*) corresponds to the average of 10000 independent runs per configuration. We can first highlight that the difference between the two values is almost negligible (the maximum relative error is less than 0.8%), proving the validity of the proposed model. Note that for larger densities ($\approx 1/2$, i.e $w = k/2$) the average number of additional transmissions is ≈ 1.6, for all k; this result matches the value that would have been obtained for the traditional *RLNC* approach, as it is proved in Appendix A. However, for lower densities, i.e. $w = 3$, the number of required transmissions increases considerably, especially for larger k.

In order to complement the average values collected in Table II, Figure 5 shows the probability

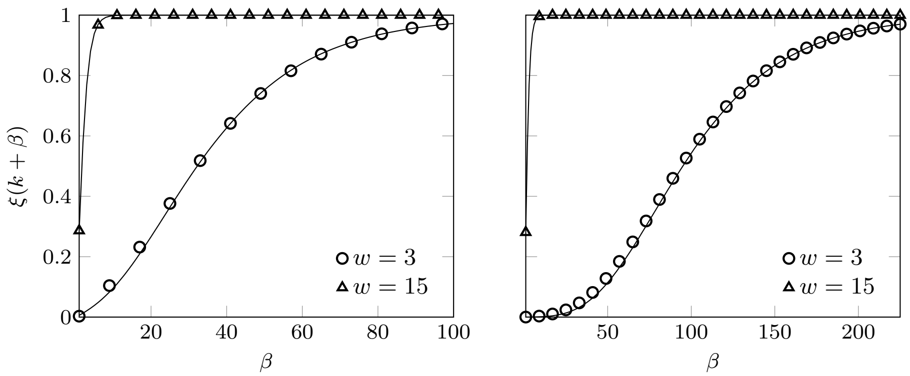
TABLE II

AVERAGE NUMBER OF TRANSMISSIONS REQUIRED TO SUCCESSFULLY DECODE A GENERATION FOR DIFFERENT SNC CONFIGURATIONS

		$w = 3$	$w = 7$	$w = 15$	$w = 31$	$w = 64$
$k = 32$	Model	43.83	33.62	33.58	-	-
	Simul	44.17	33.64	33.60	-	-
$k = 64$	Model	100.34	65.92	65.62	65.62	-
	Simul	101.49	65.91	65.61	65.60	-
$k = 128$	Model	230.36	131.22	129.85	129.63	129.62
	Simul	231.89	131.19	129.84	129.62	129.61

that the successful decoding of the whole generation happened after receiving β additional packets (i.e $k + \beta$). Such probability is defined in Corollary 5, and the simulation results are obtained after the independent transmission of 10000 generations. Again, the difference between the results obtained by means of simulation and those using the model is negligible, being the mean squared error in the worst case ($w = 3$ and $k = 128$) rather low, $9.35 \cdot 10^{-5}$. Note that lower densities would yield a worse performance, since the probability of successfully decoding a complete generation is very small when the number of additional transmissions is low. However, when w increases, such probability equals almost 1 for just 5 additional transmissions.

Up to now, we have argued that the use of low coding densities would be beneficial since they would yield shorter decoding times. In order to assess this, Figure 6a shows the number of operations that have been performed at the decoder, from the first reception, until the packet is successfully decoded. On the other hand, Figure 6b shows the average number of coded packets sent by the encoder. We have used the KODO library, which logs the number of operations that are executed at the receiver. The complexity of an individual operation is alike for any w value, so the results shown in the figure provide a rather precise idea of the complexity of the decoding process. We plot the number of accumulated operations and number of total



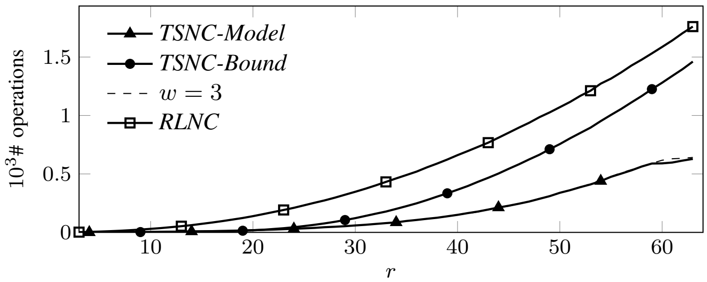
(a) $k = 64$

(b) $k = 128$

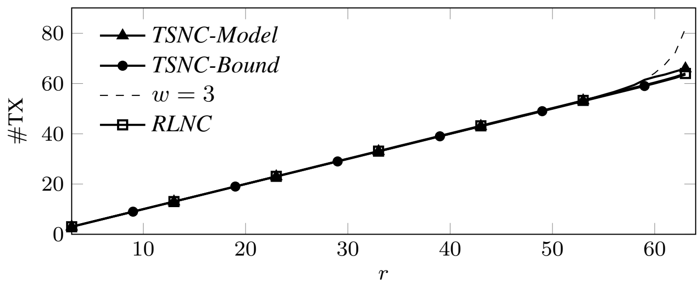
Fig. 5. Probability of a successful decoding event after β extra transmissions, $q = 1$. Markers show the values obtained with the proposed model and the line corresponds to the simulation results

transmissions versus the current rank at the receiver. The figures show the average value after 10000 independent runs. When the *TSNC* approach is used we assume that the encoder has perfect knowledge of the decoder state, and it changes the density when the expected number of transmissions to increase the rank on one unit at the decoder ($\delta^{-1}(r)$) is higher than 1.1. By using the proposed model we leverage a remarkable reduction on the complexity. In particular, the number of required operations is almost the same that those seen for a fixed SNC approach and $w = 3$. However, TSNC is able to keep the number of transmissions, which is remarkably higher for the fixed SNC solution. We can thus conclude that a dynamic tuning of the coding density would certainly yield relevant improvements. We could keep the density at a low value until such probability starts to decrease, and then shift to a higher w.

After validating the proposed model over an ideal channel, we broadened the assessment to include error-prone scenarios. For that we introduce a certain loss probability, α, over the link between the source and the receiver, so that some packets might get lost during their transmission. Figure 7 shows the average number of transmissions that are required to successfully decode a generation ($k = 64$) as a function of α. As can be seen, there is again a very tight match between



(a) Number of operations performed by the decoder



(b) Number of transmissions sent by the encoder

Fig. 6. Number of operations performed by the decoder and transmissions sent by the encoder Vs. the number of r linearly independent packets already received at the decoder, $k = 64, q = 1$

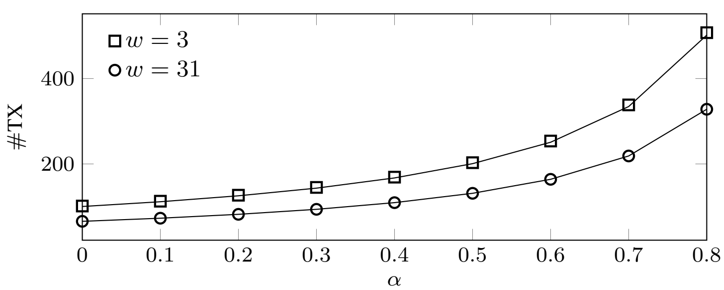


Fig. 7. Average number of transmissions required to decode a generation Vs. Packet Loss Rate (α), $k = 64, q = 1$. Markers show the values obtained with the proposed model and the solid lines correspond to the simulation results

the simulation results and the values obtained with the proposed model, the relative error is less than 0.8%. The impact of greater α values is more relevant for lower densities, since the number of required transmissions would increase quite strongly.

The previous results are broadened in Figure 8, which shows the impact of packet erasure

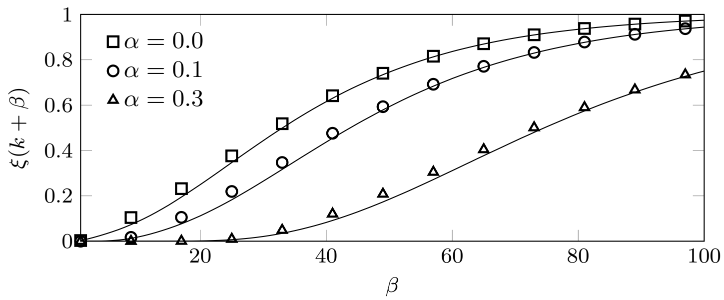


Fig. 8. Probability of a successful decoding event after β extra transmissions, $k = 64$ and $q = 1$. Markers show the values obtained with the proposed model and the solid lines correspond to the simulation results

channels over the probability of being able to decode a generation after $\#\text{TX} = k + \beta$ transmissions. We have used a low density, $w = 3$, since it showed a worse behavior in terms of the additional transmissions that are required. The impact of the packet-erasure links is again clearly seen, since α sharply decreases when gets higher. On the other hand, we can see that there is a slight difference between the simulation results and the values obtained with the model. As was already discussed, the fitting for $w = 3$ was less accurate than the ones for larger densities; in any case, the differences are rather small, and the results clearly show that the proposed model can also be used over error-prone links. In this case the mean squared error for the worst case ($\alpha = 0.3$) is approximately $3.9 \cdot 10^{-4}$.

V. Conclusions and future work

In this paper we have presented the first complete model to mimic the behavior of sparse coding techniques. It is based on an Absorbing Markov Chain, where the states are defined as the combination of the rank and the non-zero columns of the corresponding decoding matrix. After finding the transition probabilities, we have exploited the properties of Absorbing Markov Chains to derive some key performance metrics: average number of transmissions, probability of successfully decoding a generation after $\#\text{TX}$ transmissions, and probability of receiving a linearly independent packet. We have also seen that it can be broadened to consider error-prone links. The model has been validated by means an extensive simulation campaign, yielding an almost perfect match, and clearly outperforming some of the bounds that have been used in previous works.

This model can be easily exploited to select the most appropriate density in sparse coding techniques. In particular, in a TSNC solution, the encoder decides the optimum density value as the transmission evolves, for instance taking into account the trade-off between useless transmissions (linearly dependent combinations) and the reduction on computational complexity. Hence, the proposed model could provide a better insight to establish optimum configurations of TSNC solutions. We will also look at the recoding feature of intermediate nodes. In particular, we will seek solutions that can keep the density of the received packets at a given level. The proposed model might as well help the intermediate nodes to decide when to start recoding packets and the best configuration to do so.

APPENDIX A
AVERAGE NUMBER OF TRANSMISSIONS FOR RLNC

Lucani *et al.* demonstrated in [45] that the average number of transmissions that are required to successfully decode a complete generation when RLNC is used, could be estimated as the sum of the generation size (k) and a constant that only depends on the finite field, $GF(Q = 2^q)$, but not on the generation size. In this Appendix we introduce a novel theorem, which extends such result, by establishing the exact value of such constant.

Theorem 8. *Average Number of Transmissions for RLNC.*

The average number of packets that need to be transmitted to decode a complete generation when RLNC is used can be calculated as follows:

$$\overline{\#TX} = \sum_{i=0}^{k-1} \frac{1}{Prob_{r^+}^{RLNC}(i)} \approx k + \beta \tag{18}$$

where $Prob_{r^+}^{RLNC}(i)$ is the probability that a received packet increases the rank of the decoding matrix when the traditional RLNC scheme is used, which was obtained by Trullols et al. *in [37], $Prob_{r^+}^{RLNC}(i) = 1 - \frac{Q^i}{Q^k}$, and β is a constant that only depends on the finite field used.*

Proof: We start with the initial expression, given by the summation in (18), and we apply the Euler-Maclaurin formula, as can be seen in (19), where $\mathcal{B}_{2t}$ are the Bernoulli numbers and $f^{(n)}(x)$ is the n^{th} derivative of $f(x)$. We assume that the third term of the summation ($t = 3$) is much smaller than the previous ones and we thus neglect the subsequent terms, considering only the initial function $f(x)$ and its first and third derivatives, which are shown in (20).

The result of the integral is given by (21). We also take the following assumptions: $f(0) \approx 1,\ f^{'}(0) \approx 0,\ f^{'''}(0) \approx 0$, which are indeed sensible, given that $k \gg 1$. Finally, (22) shows the sought result, the average number of transmissions, which can be represented as the sum of the generation size, k, and a constant, β, that depends on the particular Galois Field.

■

As can be seen, if we substitute Q with 2, $\overline{\#\text{TX}}$ approximately yields $k + 1.6$, matching the value empirically observed, or computed by the evaluation of the series, the first expression in (18), which could be cumbersome for large k.

On the other hand, the behavior of SNC would be quite similar to the one exhibited by RLNC when the density, defined as $d = \frac{w}{k}$, is close to $d = 1 - \frac{1}{2^q}$. This is due to the fact that in RLNC all coefficients are used for every coded packet, but some of them might be zero. For $GF(2)$, for instance, there would be, in average, $\frac{k}{2}$ zero coefficients per coded packet, i.e. a density of $d \approx \frac{1}{2}$. Table III shows the values of β for various q, obtained by (22), comparing them with the number of transmissions that were obtained by averaging 1000 independent experiments for different configurations of the TSNC. As can be seen, the simulation-based values match almost perfectly the analytical ones, assessing the validity of the Theorem's main result.

$$\overline{\#\mathrm{TX}} = 1 + \sum_{i=1}^{k-1} \frac{1}{1-\frac{Q^i}{Q^k}} = 1 + \int_{i=0}^{k-1} \frac{1}{1-\frac{Q^i}{Q^k}} di + \frac{f(k-1)-f(0)}{2} + \\ + \sum_{t=1}^{\infty} \frac{\mathcal{B}_{2t}}{(2t)!} \left[f^{(2t-1)}(k-1) - f^{(2t-1)}(0) \right] \quad (19)$$

$$f(x) = \frac{1}{1-\frac{Q^x}{Q^k}} \quad f'(x) = \frac{Q^{x-k}\log Q}{\left(Q^{x-k}-1\right)^2} \quad f'''(x) = \frac{Q^{x+k}(\log Q)^3(Q^{2k}+4Q^{x+k}+Q^{2x})}{(Q^x-Q^k)^4} \quad (20)$$

$$\int_{i=0}^{k-1} \frac{1}{1-\frac{Q^x}{Q^k}} dx = \left[x - \frac{1}{\log Q} \log\left(1 - \frac{e^{x \log Q}}{Q^k}\right)\right]_0^{k-1} = k - 1 - \frac{\log\left(1-\frac{1}{Q}\right)}{\log Q} \quad (21)$$

$$\overline{\#\mathrm{TX}} \approx k - \frac{\log\left(1-\frac{1}{Q}\right)}{\log Q} + \frac{1}{2}\left(\frac{1}{Q-1}\right) + \frac{1}{12}\frac{Q\log Q}{\left(Q-1\right)^2} \quad (22)$$

TABLE III

AVERAGE NUMBER OF TRANSMISSIONS ($k+\beta$) FOR SPARSE CODING TECHNIQUES WHEN $d = 1 - \frac{1}{2^q}$

	$q=1$	$q=2$	$q=3$	$q=4$	$q=8$
β - Eq. (22)	1.60	0.41	0.15	0.07	0.00
$k=16$	17.67	16.42	16.16	16.10	16.01
$k=32$	33.63	32.43	32.16	32.08	32.00
$k=64$	65.61	64.41	64.17	64.07	64.01
$k=128$	128.62	128.41	128.16	128.07	128.00
$k=255$	256.60	255.42	255.16	255.08	255.00

ACKNOWLEDGEMENTS

This work has been supported by the Spanish Government (Ministerio de Economía y Competitividad, Fondo Europeo de Desarrollo Regional, FEDER) by means of the projects **COSAIF**, *"Connectivity as a Service: Access for the Internet of the Future"* (TEC2012-38754-C02-01), and **ADVICE** (TEC2015-71329-C2-1-R). This work was also financed in part by the **TuneSCode** project (No. DFF 1335-00125) granted by the Danish Council for Independent Research.

REFERENCES

[1] M. Wang and B. Li, "How practical is network coding," in *Proc. of the 14th IEEE International Workshop on Quality of Service*, June 2006, pp. 274–278.

[2] S. Feizi, D. E. Lucani, and M. Médard, "Tunable sparse network coding," in *Proc. of the Int. Zurich Seminar on Comm*, 2012, pp. 107–110.

[3] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, Jul 2000.

[4] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking.*, vol. 11, no. 5, pp. 782–795, Oct. 2003.

[5] S.-Y. Li, R. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, Feb 2003.

[6] T. Ho, R. Koetter, M. Medard, D. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *Proc of the IEEE International Symposium on Information Theory*, 2003, p. 442.

[7] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the air: Practical wireless network coding," *IEEE/ACM Transactions on Networking*, vol. 16, no. 3, pp. 497–510, Jun. 2008.

[8] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," in *Proc. of the ACM conference on Applications, techonologies, architectures and protocols for computer communications*, vol. 37, no. 4. ACM, Aug. 2007, pp. 169–180.

[9] F. Zhao and M. Médard, "On analyzing and improving COPE performance," in *Proc. of the Information Theory and Applications Workshop (ITA)*, Jan 2010, pp. 1–6.

[10] F. Zhao, M. Médard, M. Hundeboll, J. Ledet-Pedersen, S. A. Rein, and F. H. P. Fitzek, "Comparison of analytical and measured performance results on network coding in IEEE 802.11 ad-hoc networks," in *Proc. of the International Symposium on Network Coding (NetCod)*, June 2012, pp. 43–48.

[11] D. Traskov, N. Ratnakar, D. S. Lun, R. Koetter, and M. Medard, “Network coding for multiple unicasts: An approach based on linear optimization,” in *Proc. of the IEEE International Symposium on Information Theory (ISIT)*, 2006, pp. 1758–1762.

[12] J. Cloud, L. Zeger, and M. Médard, “Effects of MAC approaches on non-monotonic saturation with COPE - a simple case study,” in *Proc. of the Military Communications Conference (MILCOM)*, Nov 2011, pp. 747–753.

[13] D. Gomez, S. Hassayoun, A. Herrero, R. Aguero, and D. Ros, “Impact of network coding on TCP performance in wireless mesh networks,” in *Proc. of the 23rd IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, Sept 2012, pp. 777–782.

[14] J. Sundararajan, D. Shah, M. Medard, M. Mitzenmacher, and J. Barros, “Network coding meets TCP,” in *Proc. of the IEEE International Conference on Computer Communications (INFOCOM)*, April 2009, pp. 280–288.

[15] D. Gómez, E. Rodríguez, R. Agüero, and L. Muñoz, “Reliable communications over lossy wireless channels by means of the combination of UDP and random linear coding,” in *Proc. of the IEEE Symposium on Computers and Communications (ISCC)*, June 2014, pp. 1–6.

[16] P. Pahlevani, D. E. Lucani, M. V. Pedersen, and F. H. P. Fitzek, “PlayNCool: Opportunistic network coding for local optimization of routing in wireless mesh networks,” in *Proc. of the IEEE Globecom Workshops (GC Wkshps)*, Dec 2013, pp. 812–817.

[17] D. Gomez, P. Garrido, E. Rodriguez, R. Aguero, and L. Munoz, “Enhanced opportunistic random linear source/network coding with cross-layer techniques over wireless mesh networks,” in *Proc. of the IFIP Wireless Days Conference (WD)*, Nov 2014, pp. 1–4.

[18] S. Pandi, F. Fitzek, J. Pihl, M. V. Pedersen, and D. Lucani, “Sending policies in dynamic wireless mesh using network coding,” in *Proc. of the 21st European Wireless Conference*, May 2015, pp. 1–7.

[19] M. Luby, “LT codes,” in *Proc. of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, 2002, pp. 271–280.

[20] A. Shokrollahi, “Raptor codes,” *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551–2567, June 2006.

[21] C. Sorensen, D. Lucani, F. Fitzek, and M. Medard, “On-the-fly overlapping of sparse generations: A tunable sparse network coding perspective,” in *Proc of the 80th IEEE Vehicular Technology Conference (VTC Fall)*, Sept 2014, pp. 1–5.

[22] H. Shojania and B. Li, “Parallelized progressive network coding with hardware acceleration,” in *Proc. of the 15th IEEE International Workshop on Quality of Service*, June 2007, pp. 47–55.

[23] C. W. Sørensen, A. Paramanathan, J. Guerrero, M. V. Pedersen, D. E. Lucani, and F. Fitzek, “Leaner and meaner: Network coding in SIMD enabled commercial devices,” in *Proc. of the IEEE Wireless Communications and Networking Conference (WCNC)*, 2016.

[24] J. Heide, M. Pedersen, F. Fitzek, and M. Medard, “On code parameters and coding vector representation for practical RLNC,” in *Proc. of the IEEE International Conference on Communications (ICC)*, June 2011, pp. 1–5.

[25] P. A. Chou, Y. Wu, and K. Jain, “Practical network coding,” October 2003.

[26] P. Maymounkov, N. J. Harvey, and D. S. Lun, "Methods for efficient network coding," in *Proc. of the 44th Annual Allerton Conference on Communication, Control, and Computing*, 2006, pp. 482–491.

[27] M. Wang and B. Li, "R2: Random push with random network coding in live peer-to-peer streaming," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 9, pp. 1655–1666, December 2007.

[28] C. Gkantsidis, J. Miller, and P. Rodriguez, "Comprehensive view of a live network coding P2P system," in *Proc. of the 6th ACM Conference on Internet Measurement (IMC)*. New York, NY, USA: ACM, 2006, pp. 177–188.

[29] B. Tang, S. Yang, B. Ye, Y. Yin, and S. Lu, "Expander chunked codes," *CoRR*, vol. abs/1307.5664, 2013. [Online]. Available: http://arxiv.org/abs/1307.5664

[30] S. Yang and B. Tang, "From LDPC to chunked network codes," in *Proc. of the IEEE Information Theory Workshop (ITW)*, Nov 2014, pp. 406–410.

[31] S. Feizi, D. E. Lucani, C. W. Sorensen, A. Makhdoumi, and M. Medard, "Tunable sparse network coding for multicast networks," in *Proc. of the International Symposium on Network Coding (NetCod)*, June 2014, pp. 1–6.

[32] C. W. Sorensen, A. S. Badr, J. A. Cabrera, D. E. Lucani, J. Heide, and F. Fitzek, "A practical view on tunable sparse network coding," in *Proc. of the 21st European Wireless Conference*, May 2015, pp. 1–6.

[33] R. Prior, D. E. Lucani, Y. Phulpin, M. Nistor, and J. Barros, "Network coding protocols for smart grid communications," *IEEE Transactions on Smart Grid*, vol. 5, no. 3, pp. 1523–1531, May 2014.

[34] D. E. Lucani, M. V. Pedersen, J. Heide, and F. Fitzek, "Fulcrum network codes: A code for fluid allocation of complexity," *CoRR*, vol. abs/1404.6620, 2014. [Online]. Available: http://arxiv.org/abs/1404.6620

[35] S. Yang and R. W. Yeung, "Batched sparse codes," *IEEE Transactions on Information Theory*, vol. 60, no. 9, pp. 5322–5346, Sept 2014.

[36] K. Mahdaviani, M. Ardakani, H. Bagheri, and C. Tellambura, "Gamma codes: A low-overhead linear-complexity network coding solution." in *Proc. of the International Symposium on Network Coding (NetCod)*, 2012, pp. 125–130.

[37] O. Trullols-Cruces, J. Barcelo-Ordinas, and M. Fiore, "Exact decoding probability under random linear network coding," *IEEE Communications Letters*, vol. 15, no. 1, pp. 67–69, January 2011.

[38] P. Garrido, D. Gomez, R. Aguero, and L. Munoz, "Performance of random linear coding over multiple error-prone wireless links," *IEEE Communications Letters*, vol. 19, no. 6, pp. 1033–1036, June 2015.

[39] X. Li, W. H. Mow, and F. L. Tsang, "Singularity probability analysis for sparse random linear network coding," in *Proc. of the IEEE International Conference on Communications (ICC)*, June 2011, pp. 1–5.

[40] ——, "Rank distribution analysis for sparse random linear network coding," in *Proc. of the International Symposium on Network Coding (NetCod)*, July 2011, pp. 1–6.

[41] J. Blomer, R. Karp, and E. Welzl, "The rank of sparse random matrices over finite fields," *Random Structures and algorithms*, vol. 10, no. 4, pp. 407–420, 1997.

[42] A. Tassi, I. Chatzigeorgiou, and D. E. Lucani, "Analysis and optimization of sparse random linear network coding for reliable multicast services," *IEEE Transactions on Communications*, vol. 64, no. 1, pp. 285–299, Jan 2016.

[43] J. G. Kemeny and J. L. Snell, *Finite markov chains*. van Nostrand Princeton, NJ, 1960, vol. 356.

[44] M. R. Albrecht, "The M4RIE library for dense linear algebra over small fields with even characteristic," *CoRR*, vol. abs/1111.6900, 2011. [Online]. Available: http://arxiv.org/abs/1111.6900

[45] D. E. Lucani, M. Médard, and M. Stojanovic, "Random linear network coding for time-division duplexing: Field size considerations," *Proc. of the IEEE Global Telecommunications Conference (GLOBECOM)*, 2009.