An Analytical Model for Sparse Network Codes: Field Size Considerations

Amir Zarei, Peyman Pahlevani and Daniel E. Lucani, Senior Member, IEEE

***Abstract*—One of the by-products of Sparse Network Coding (SNC) is the ability to perform partial decoding, i.e., decoding some original packets prior to collecting all needed coded packets to decode the entire coded data. Due to this ability, SNC has been recently used as a technique for reducing the Average Decoding Delay (ADD) per packet in real-time multimedia applications. This study focuses on characterizing the ADD per packet for SNC considering the impact of finite field size. We present a Markov Chain model that allows us to determine lower bounds on the mean number of transmissions required to decode a fraction of a generation and the ADD per packet of the generation. We validate our model using simulations and show that the smaller finite fields, e.g., $q = 2^4$, outperform large finite fields, e.g., $q = 2^{32}$, in regard to the ADD per packet and provide a better trade-off between the ADD per packet and the overall number of transmissions to decode a generation.**

***Index Terms*—Random Linear Network Coding, Sparse Network Coding, Average Decoding Delay per packet**

I. Introduction

RANDOM Linear Network Coding (RLNC) was introduced as a random strategy for decentralized Network Coding (NC) which achieves multi-cast capacity asymptotically [1]. From a practical perspective, RLNC considers a sender that divides a block data into n original packets and organizes them in a group called a generation. The sender generates random linear combinations of those n packets, while a receiver(s) collects coded packets until it would be able to decode the whole generation. This occurs as soon as the receiver obtains n linearly independent coded packets. RLNC designed to achieve capacity will have poor partial performance. Even if the number of coded packets is very close to (but not exactly) sufficient, the fraction of original packets that can be decoded will be negligible. This disadvantage can be significantly reduced by operating a sparse implementation of RLNC [2]. Sparse Network Coding (SNC) was first introduced as a mechanism to alleviate the decoding complexity of RLNC by selecting a large fraction of zero-valued coding coefficients in the encoding matrix [3], [4]. In this context, an encoding vector is defined w - $sparse$, if it consists of exactly w coding coefficients which are chosen at random and their value is selected uniformly from the elements of a finite field of size q (F_q), while the remaining $n - w$ entries of the encoding vector are set to zero. According to this definition SNC with $w = n$ is identical to RLNC. This definition is also different than that has provided in [2] that the w selected coefficients must be nonzero.

SNC is also able to perform partial decoding employing a low complexity encoding and decoding algorithm. Let us look at a scenario where a data stream is to be delivered to multiple users over a shared medium. The stream needs to be decoded in real time, or near-real time with a finite amount of buffering. If NC is to be employed for the transmission, the stream would have to be broken into generations of original packets, and each generation would have to be encoded separately. If such a transmission strategy is employed, the coded packets from any given generation will only be transmitted from a finite amount of time, before the encoder moves on the next generation. If the user channel qualities are very different, it is likely that a user may not receive the required number of coded packets to decode the entire of a generation. It is reasonable to assume that many real-time applications, if suitably pre-coded, can be reliably played back from a large fraction of original packets. For example, in conversational video applications, there is a limit of 100 ms in which each original packet must be received and playback within this time. Moreover, packet loss rates between 1 and 10 percent can be tolerated [9]. Unlike RLNC, SNC does not need to wait until the whole generation is decoded and it can exploit the partial recovery of the original packets. When the playback time is approaching, the decoder must decode and playback the recoverable packets and drop packets that cannot be decoded. In this way, the receiver may have the benefits of forward erasure correction considering the delay constraint of the conversational video applications. A key point is that the level of sparsity can have effect on the decoding delay so that too sparse or dense coding schemes may create higher delay.

Both SNC and RLNC are exploiting on-the-fly version of Gauss-Jordan algorithm [7]. In this approach, the newly received vector is forward substituted into the previous received encoding vectors, and subsequently backward substitution is performed to bring the decoding matrix into echelon form. Since SNC's decoding matrix is sparse, it needs less operations to bring echelon form of the matrix leading faster decoding speed and less computational complexity. Such that, the authors in [8] provided an efficient and intuitive algorithm for tuning SNC by reordering rows and columns of a matrix to perform efficient Gaussian elimination in sparse matrices. This algorithm guarantees linear decoding complexity of sparse packets.

SNC's partial performance was first analyzed in [2] using an Absorbing Markov Chain. The model was based on a new suitable concept, so that the authors focused on decoding delay of each original packet to get Average on Decoding Delay of all original packets of a generation, denoted by ADD per packet. The provided model characterized the overall number of transmissions and the ADD per packet to recover a generation considering large finite field sizes(e.g., $q \geq 2^{16}$). The assumption of large size of q is a potential limitation, especially for the practical implementations of SNC. To investigate this limitation, we have simulated a scenario of one sender and one receiver, where the

A. Zarei is with Institute for Advanced Studies in Basic Sciences (IASBS), Zanjan, Iran, and a visiting researcher at Department of Engineering at Aarhus university (e-mail: az.hldv@gmail.com). P. Pahlevani is with Wireless Communication Laboratory, IASBS, Zanjan, Iran. D. E. Lucani is with Department of Engineering at Aarhus university

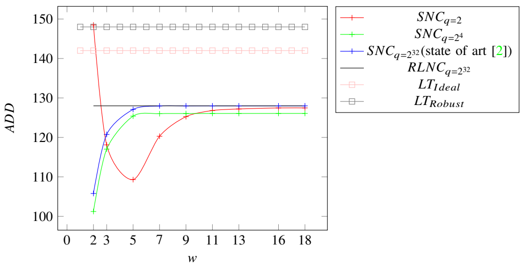


Fig. 1: The ADD per packet to decode a generation of 128 original packets for three different systems: SNC, RLNC and LT.

SNC sender uses a fixed w to deliver a generation of 128 original packets to the receiver over a lossless channel. In such scenario, we scrutinize the impact of the finite field size on the ADD per packet by Fig. 1. In other words, Fig. 1 illustrates the impact of different q on the ADD per packet for some values of w, and shows that the SNC system establishes a better performance of $q = 2$ for $w \in (3, 9]$, as well $q = 2^4$ for $w \leq 3$ and $w > 9$. Fig. 1 has also considered the RLNC performances for $q = 2^{32}$, and LT code [6] based on two degree distributions: Ideal and Robust soliton. Fig. 1 asserts the improved ADD per packet of SNC and RLNC than the provided LT code. We can finally conclude that modeling the impact of q on the ADD is key to characterize a SNC system. This paper focuses on generalizing the preliminary work in [2] for the case of general finite field sizes and makes the following contributions:

Proposing the analytical model: The model is based on an Absorbing Markov Chain, where the states employ the notion of covering original packets, i.e., the nonzero columns at the decoding matrix. This enables us to nominate the states with the ability to recover a fraction of a generation. Next, we present lower bounds on the number of transmissions to recover a fraction of a generation (recovering minimum x from n original packets) and the imposed ADD per packet to recover a generation.

Confirming the accuracy of the model: Our analysis shows that the difference of the Markov model and the simulation results are insignificant for a wide range of generation sizes, packet loss rates, finite field sizes, and sparsity levels. More precisely, the deviations with reference to mean square of the gap between the simulation results and the provided lower bounds for the overall number of transmissions to decode a fraction of a generation and ADD per packet of the generation are only 6 % and 7 %, respectively. Moreover, as a benchmark, we compare our results with RLNC and LT code.

Trade-off and Operating Regions for Finite Field and Sparsity: Although using $q = 2^{32}$ leads to a better performance for the overall number of transmissions, $q = 2^4$ outperforms $q = 2^{32}$ in terms of ADD per packet. Furthermore, the proposed model enables us to opt the suitable sparsity and field size to establish the trade-off between ADD per packet and the overall number of transmissions required to decode a generation.

The rest of the paper is structured as follows. In Section II and III, we describe the system model and analyze the impact of the finite fields size, respectively. Then, Section IV validates the models by means of a broad simulation campaign and discusses the obtained results. Finally, Section V concludes the paper.

II. System Model

We consider a SNC system of one sender transmitting coded packets to one receiver. For simplicity, we first suppose that the channel between the sender and the receiver is lossless, then extend our model over a channel with the probability of packet loss equal to ϵ. A time step is defined as the duration of transmitting one coded packet, implying that n coded packets can be received at the receiver in n time steps in a lossless channel. We say that the generation of $\mathbf{S}$ consists of n original packets $\mathbf{S} = \{\mathbf{s}_i\}_{i=1}^{n}$. At time step j, the sender generates a *w-sparse* coded packet $\mathbf{u}_j$, which is constructed as $\mathbf{u}_j = \sum_{i=1}^{w} g_{f(i),j} \times s_{f(i)}$, where $f(i)$ is selected randomly from the set $\{1, ..., n\}$. The coefficients $g_{f(i),j}$ are chosen randomly from the elements of F_q, in an identical and independent pattern conforming to the following probability

$$p(g_{i,j} = v) = \frac{1}{q}, \quad \forall v \in \{0, 1, 2, ..., 1 - q\}. \tag{1}$$

Let $\mathbf{u}_1, ..., \mathbf{u}_m$ denote coded packets that the receiver has collected to decode the $\mathbf{S}$ ($m \geq n$). Also, let M represents the $m \times n$ decoding matrix constructed at the receiver. The relationship between $\mathbf{u}_1, ..., \mathbf{u}_m$ and the original packets $\mathbf{s}_1, ..., \mathbf{s}_n$ is

$$\begin{bmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_m \end{bmatrix} = \mathbf{M} \begin{bmatrix} \mathbf{s}_1 \\ \vdots \\ \mathbf{s}_n \end{bmatrix}$$

where

$$\mathbf{M} = \begin{bmatrix} g_{1,1} & g_{1,2} & g_{1,3} & \cdots & g_{1,n} \\ g_{2,1} & g_{2,2} & g_{2,3} & \cdots & g_{2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_{m,1} & g_{m,2} & g_{m,3} & \cdots & g_{m,n} \end{bmatrix}$$

In order to record the coefficients of performed linear combinations, a network coding header is appended to each produced original packet by the sender as

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_n \end{bmatrix} = \left[\begin{array}{ccc|c} 1 & \dots & 0 & \mathbf{s}_1 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 1 & \mathbf{s}_n \end{array}\right] = \begin{bmatrix} I_n \mathbf{S} \end{bmatrix}$$

Decoding can be done via Gauss-Jordan elimination:

$$Y = \mathbf{MX} = \begin{bmatrix} \mathbf{M} & \mathbf{MS} \end{bmatrix} \Longrightarrow RREF(Y) = \begin{bmatrix} I_n \mathbf{S} \end{bmatrix},$$

where $RREF(Y)$ is Y matrix in Reduced Row Echelon Form.

We define the decoding delay of original packet s_i as the overall number of time steps until the original packet s_i is decoded, denoted by d_i. By using this definition, the ADD per packet for the generation of $\mathbf{S}$ can be calculated as

$$ADD = \frac{\sum_{i=1}^{n} d_i}{n}. \tag{2}$$

III. Proposed Model

Eq. (1) obviously shows that by reducing the field size q, the probability of choosing a zero-coefficient as an element of F_q is increased. Therefore, for small field sizes, the probability of choosing the zero-coefficient as part of the w original packets is higher. For example, if $q = 2$, the coding coefficients are chosen from the set $\{0, 1\}$, and this probability $p(g_{i,j} = 0)$ is equal to 0.5. This value has a significant impact on the probability of generating a linear dependent coded packet. Because of this, the smaller the q, the less valid the provided model in [2] is. To design a complete model incorporating the effect of field size, it is enough to consider the probability of choosing the zero-coefficient.

The system is modeled through an Absorbing Markov model with states (c,t), where t is the current number of coded packets collected by the receiver, and c is the number of covering original packets by t. Every new received coded packet causes the state to change from (c,t) to $(c+i,t+1)$. As an example, the transition probability $p^{w,n}_{c,t}(i)$ between states (c,t) and $(c+i,t+1)$, when the sender sets $w = 2$ to transmit coded packets, is defined in the following way:
The new received coded packet covers zero, one or two new original packets. For $i = 0$, where any new original packet is not covered, the total probability obtains from three probabilities. First, the probability that the combined original packets already covered by $t-1$ collected coded packets. Second, one new original packet is selected by the sender but its coefficient is equal to zero. Third, two new original packets are selected but their coefficients are equal to zero.

$$p^{2,n}_{c,t}(0) = \left(\frac{\binom{c}{2}}{\binom{n}{2}}\right) + \left(\frac{\binom{c}{1}\times\binom{n-c}{1}}{\binom{n}{2}}\times(\frac{1}{q})^1\right) + \left(\frac{\binom{n-c}{2}}{\binom{n}{2}}\times(\frac{1}{q})^2\right).$$

For $i = 1$, where a new original packet is covered, the total probability obtains from two probabilities. First, the probability that one new original packet is selected and its coefficient is not equal to zero. Second, two new original packets are selected but the coefficient of one of them is not equal to zero.

$$p^{2,n}_{c,t}(1) = \left(\frac{\binom{c}{1}\times\binom{n-c}{1}}{\binom{n}{2}}\times(1-\frac{1}{q})^1\right) +$$
$$\left(\frac{\binom{n-c}{2}}{\binom{n}{2}}\times(1-\frac{1}{q})^1\times(\frac{1}{q})^1\times\binom{2}{1}\right).$$

Finally, for $i = 2$, where two new original packets are covered, the total probability obtains when two new original packets are covered and their coefficients are not equal to zero.

$$p^{2,n}_{c,t}(2) = \left(\frac{\binom{n-c}{2}}{\binom{n}{2}}\times(1-\frac{1}{q})^2\right).$$

By using Eq. (3), the transition probabilities can be easily derived to the case of general w, in the following way:

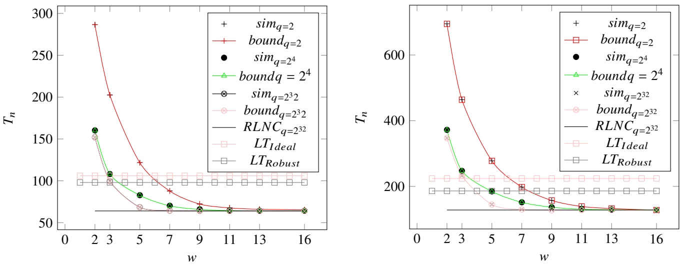
$$p^{w,n}_{c,t}(i) = 0 \qquad \text{if } (c,t) \in \mathbb{A},$$

$$p^{w,n}_{c,t}(i) = \sum_{j=i}^{b_2} \frac{\binom{c}{w-j}\times\binom{n-c}{j}}{\binom{n}{w}}\times(1-\frac{1}{q})^i\times(\frac{1}{q})^{j-i}\times\binom{j}{i}$$
$$\text{if } (c,t) \notin \mathbb{A} \text{ and } i \in B_{n,c,w}. \quad (3)$$

where $\mathbb{A}$ consists of the states with possibility of decoding at least x out of n original packets, i.e., absorbing states, and $B_{n,c,w} = [b_1, b_1+1, ..., b_2]$,

$$B_{n,c,w} = \begin{cases} [w-c, w] & \text{if } w \in (c, n-c], \\ [0, n-c] & \text{if } w \in (n-c, c], \\ [0, w] & \text{if } w \le c \text{ and } w \le n-c \\ [w-c, n-c] & \text{if } w > c \text{ and } w > n-c. \end{cases}$$

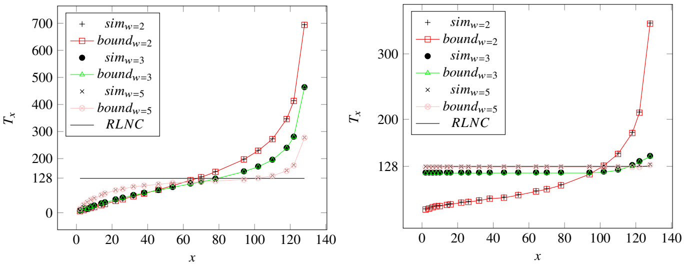
A coded packet is constituted by w original packets and the coefficients can be zero or nonzero. Thus, the states with $(c < w, t)$ are visible. However, the states with $(c > min(t \times w, n), t)$ are not visible because every transmission covers maximum w new original packets. Therefore, we consider the probability of visiting these states equals to zero.



(a) $n = 64$

(b) $n = 128$

Fig. 2: The average number of transmissions needed to decode a generation for different n,w and q.



(a) $q = 2$

(b) $q = 2^{32}$

Fig. 3: The average number of transmissions needed to decode a fraction of a generation for $n = 128$, and $q = 2$ and $q = 2^{32}$.

Finally, the occurrence probability of the state (c, t), denoted by $pr_{c,t}$, can be derived as follows:

$$pr_{c,t} = \begin{cases} 1 & \text{if } t = 0 \text{ or } 1, \\ \sum_{i=0}^{w} p^{w,n}_{c-i,t}(i) \times pr_{c-i,t-1} & \text{otherwise.} \end{cases} \quad (4)$$

A. Lower bound on the overall number of transmissions

In order to derive a lower bound on the average number of transmissions required to decode x from n original packets, we first restrict our focus on the Markov model 's parameters. More precisely, *Lemma 1* in [2] determines that the $\mathbb{A}$ includes potential states with the chance of decoding x from n original packets. Note that, the occurrence of these states dose not guarantee decoding x original packets. Thus, to provide a reasonable approximation, we calculate a multiplication of the occurrence probability of each absorbing state by its t parameter, then, sum these values.

Lemma 1 ([2]). For recovering minimum x from n original packets, merely states with $(c \ge x, t \ge (x + \lceil\frac{c-x}{w}\rceil)$ are absorbing.

The same as previous model [2], we can now provide a lower bound based on the states $(c \ge x, t \ge (x + \lceil\frac{c-x}{w}\rceil)) \in \mathbb{A}$, in the following way:

$$T_x = \sum_{\forall(c,t)\in\mathbb{A}} \lceil pr_{c,t} \times t \rceil, \quad (5)$$

where t is the collected number of coded packets for covering c original packets, and T_x signifies the average number of transmissions in order to decode minimum x from n original packets. In addition, to recover the whole generation in Eq. (5), let $x = n$. In this case T_n shows the average number of transmissions needed to recover the whole generation.

B. Lower bound on ADD per packet

T_x and d_i are an approximation bound on the decoding minimum x original packets and the required number of transmissions until the original packet s_i is recovered, respectively. To derive a lower bound on the ADD per packet, we say for decoding the x-th packet of the x original packets, collecting T_x coded packets is needed. Also, it suffices collecting maximum T_x coded packets to recover the remaining original packets. Therefore, T_i can be investigated as a lower bound on the decoding delay of d_i. By substituting T_i with d_i in Eq. (2), a lower bound can be gained on the ADD per packet of the generation, in the following way:

$$\widetilde{ADD} = \frac{\sum_{i=1}^{n} T_i}{n} \leq ADD, \tag{6}$$

C. Complexity of the Markov model

In order to derive the time complexity of the Markov model, we need to count the overall number of states after t transmissions. To this end, we take advantage of *Lemma 2*.

Lemma 2. The overall number of vertices, denoted by $f(t, w, n)$, in T of depth t is

$$f(t, w, n) = \begin{cases} \frac{t}{2} \times (2 + (w \times (t-1))) & \text{if } t \leq \lfloor \frac{n}{w} \rfloor, \\ \frac{\lfloor \frac{n}{w} \rfloor}{2} \times (2 + (w \times (\lfloor \frac{n}{w} \rfloor - 1))) & \\ +(t - \lfloor \frac{n}{w} \rfloor) \times (n - w + 1) & \text{if } t > \lfloor \frac{n}{w} \rfloor. \end{cases} \tag{7}$$

proof : Our Discrete-time absorbing Markov chain can be corresponded to a directed rooted tree T, where the root is a state of $(c = w, t = 1)$ and t shows the level of the tree (we assume that the root is on level 1). We can now count the overall number of vertices in the T of depth t (or the total number of states after t transmissions). One key observation is that for covering n original packets, at least $\lceil \frac{n}{w} \rceil$ received coded packets are required. Based on the Markov model for $t \leq \lfloor \frac{n}{w} \rfloor$, by receiving a $w-sparse$ coded packet, w new original packets would be covered, then the overall number of vertices at the depth t is equal to the number of vertices at the depth $t-1$ plus w. Therefore, we can claim that the number of vertices on each the tree's level follows an arithmetic progression of common difference w, initial term of 1, and general formula of $a_m = (m-1)w + 1$, so the addition of all the arithmetic progression's terms can be derived by $f(\lfloor \frac{n}{w} \rfloor, w, n) = \frac{\lfloor \frac{n}{w} \rfloor}{2} \times (2 + (w \times (\lfloor \frac{n}{w} \rfloor - 1)))$. On the other hand, after $\lceil \frac{n}{w} \rceil$ transmissions by receiving a coded packet, the number of states would not be increased than the previous level because all the original packets have been already covered, then the number of vertices on each level is a constant of $a_{\lfloor \frac{n}{w} \rfloor} = (n - (w-1))$. Finally, the overall number of vertices for $t > \lfloor \frac{n}{w} \rfloor$ is derived by summing two parts such that, for $t \leq \lfloor \frac{n}{w} \rfloor$, the tree has $f(\lfloor \frac{n}{w} \rfloor, w, n)$ vertices, and for $t > \lfloor \frac{n}{w} \rfloor$, $((t - \lfloor \frac{n}{w} \rfloor) \times (n - w + 1))$ vertices. Thus, the total number of vertices of the tree is $f(\lfloor \frac{n}{w} \rfloor, w, n) + ((t - \lfloor \frac{n}{w} \rfloor) \times (n - w + 1))$.

According to the Markov model, the number of absorbing states is infinite, thus in Eq. (7), $t \longrightarrow \infty$. However, in order to implement the model, the number of absorbing states is bounded by a constrain. Such that we only regard the absorbing states that $pr_{c,t} > \alpha$, where $\alpha = 10^{-5}$. Based on our experiment, for each n after $n + (n/10)$ transmissions $pr_{c,t} < \alpha$, if (c, t) is absorbing state. We can now obtain the complexity of the Markov based on *Theorem 1*.

Theorem 1. The time complexity of the Markov chain is $O(n^2)$. *proof:* Based on *Lemma 2*, the overall number of vertices for $t = n + (n/10)$ can be calculated by summing two parts such that, for $t \leq \lfloor \frac{n}{w} \rfloor$, the number of vertices and its complexity are $\frac{\lfloor \frac{n}{w} \rfloor}{2} \times (2 + (w \times (\lfloor \frac{n}{w} \rfloor - 1)))$ and $O(\frac{n^2}{w})$, respectively. For $t > \lfloor \frac{n}{w} \rfloor$, the number of vertices and its complexity are $(((n + n/10) - \lfloor \frac{n}{w} \rfloor) \times (n - w + 1))$ and $O(n^2)$. Therefore, we can conclude the total complexity is $O(n^2)$. □

Based on *Theorem 1*, we have the following corollaries to present the complexity of the provided lower bounds.

Corollary 1: The complexity of Eq. (5) is $O(n^2)$. *proof:* To derive T_n based on Eq. (5) and *Lemma 1* we have to run the Markov model for $t \in [n, .., n + (n/10)]$. However, it is clear that by running $pr_{n,n+(n/10)}$ the values of $pr_{n,n}, \ldots, pr_{n,n+(n/10)-1}$ would also be obtained. Then, the time complexity of Eq. (5) for calculating $pr_{n,n+(n/10)}$ and summing a multiplication of the occurrence probability of absorbing states by their t parameters are $O(n^2)$ and $O(n)$, respectively, totally $O(n^2)$.

Corollary 2: The complexity of Eq. (6) is $O(n^3)$. *proof:* To calculate Eq. (6) we need to run Eq. (5) for gaining T_i, where $i \in [1, 2, .., n]$, thus the time complexity of Eq. (6) is $O(n \times n^2)$.

D. Impact of packet loss

Until now, the channel has considered lossless. The model can be smoothly expanded for a loss channel. To this purpose, we only need to amend the transition probabilities in the following way:

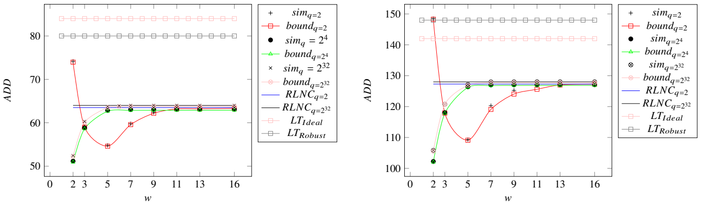
$$\widetilde{p_{c,t}^{w,n}(i)} = p_{c,t}^{w,n}(i) \times (1 - \epsilon), \tag{8}$$

where ϵ is the loss rate of the wireless channel.

IV. SIMULATION AND MODEL VALIDATION

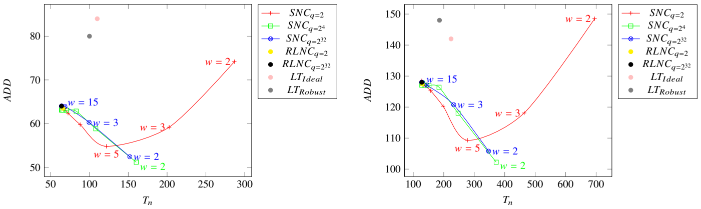
This section evaluates the validity of the proposed Markov model using a broad simulation campaign. SNC operations are handled by running Kodo library in C++ to perform encoding/decoding processes. This allows us to analyze key parameters affecting the SNC technique; sparsity level, generation size, and the finite field size. We have carried 50000 independent runs for data tuple (n, w, q), and report the expected of these experiments. As a benchmark, we compare our model with the performance of RLNC and the designed degree distributions for LT codes such as Ideal and Robust. The deviation of two plots is computed through the mean square of its vectors. For examples if the vectors x and y have n entries, the deviation of x and y equals to $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\frac{x_i - y_i}{x_i})^2}$.

Fig. 2 (a) and (b) depicts the average number of transmissions required to decode a generation via the simulation (sim) and theoretical (bound) results for the different q, w and n. We can first highlight that the difference of the two results is almost negligible. A deviation of $\sim$ 6%, for moderate values of w (w = 2 and 3), shows the validity of the proposed model. Previous studies [2], [5] showed that the use of sparse codes leads to an increase in the overall number of transmissions since high sparsity (small w) rises the likelihood of producing linear dependent coded packets, consequently rising the transmission overhead. Fig. 2 also shows the impact of the field size on the overall number of transmissions such that we can conclude that the smaller the size of q, the higher the transmission overhead



(a) $n = 64$ **(b)** $n = 128$

Fig. 4: Average decoding delay for different n,w and q.



(a) $n = 64$ **(b)** $n = 128$

Fig. 5: The ADD and the T_n for different values of w.

is. This result due to the fact that, the probability of choosing a zero-coefficient as an element of F_q is increased, especially for smaller w which the overhead is more noticeable. Fig. 2 finally demonstrates the improved performance of RLNC than Ideal and Robust Soliton degree distributions for LT codes in terms of the total number of transmissions. This is because more ability of RLNC to produce linear independent coded packets, and because RLNC takes advantage of Gaussian elimination algorithm for its decoding process. We can observe that although the LT code can present a better the total number of transmission than SNC in some configurations, SNC provides a lower number of transmissions depending on the size of w and F_q, particularly in larger w since by generating denser coded packets the SNC performance goes toward RLNC.

Fig. 3 (a) and (b) illustrates the average number of transmissions required to decode minimum x out of 128 original packets for $q = 2$ and $q = 2^{32}$. The difference of the simulation results and the obtained lower bound is very small (a deviation of ∼ 7%). When SNC is used, by decoding a fraction of the original packets, the next transmissions can be a combination of decoded original packets. These transmitted packets do not rise the independent linear combination received at the receiver side, they only rise the transmission overhead, especially for smaller w. Therefore, Fig. 3 shows that the highest number of transmissions to recover a generation occurs for $w = 2$ to all the sizes of q. Furthermore, Fig. 3 depicts the impact of the finite field size on partial decoding for the different w and q. It is clear that, for smaller w, partial decoding starts earlier than other w. For the case of $q = 2$ and $w = 2$, 3 and 5, the receiver starts partial decoding as soon as it collects 6.96, 8.74 and 18.85 coded packets, respectively. These numbers increase to 63.39, 118.54 and 128, for $q = 2^{32}$ and $w =$ 2, 3 and 5, respectively.

Fig. 4 (a) and (b) collects the imposed ADD per packet to recover a generation for the different settings, in which we have modified the values of n, q and w. We can first remark that the difference of the simulation and analytical results is almost negligible.. Also, the impact of the sparsity level on the SNC's ADD per packet is significant. Such that for small w ($w = 2$ and 3) and the size of moderate generations ($n = 64$ and 128), SNC provides 13 % and 12 % improvement in compared with RLNC for $q = 2^4$ and $q = 2^{32}$, respectively. Fig. 4 also shows a different behavior for $q = 2$ and $w < 5$. Such that the high probability of choosing the zero-coefficient, along with the small size of the w lead to rise the transmission overhead, consequently an increase in ADD per packet. Moreover, Fig. 4 indicates that for $w = 2$, $q = 2^4$ and $q = 2^{32}$ have a better ADD per packet than $q = 2$. However, for $w \in [3, 11]$, $q = 2$ has the better performance than other finite field sizes. Although using $q = 2^{32}$ leads to better efficiency in terms of the overall number of transmissions, $q = 2^4$ provides 4% and 3% improvement compared to $q = 2^{32}$ for ADD per packet to decode the generations with sizes 64 and 128, respectively. The illustrated results in Fig. 4 finally show an improvement in SNC and RLNC than the LT code in terms of ADD per packet. This is because NC schemes take advantage of more efficient decoding algorithm in order to perform partial decoding.

Fig. 5 (a) and (b) depicts a trade-off curve between the imposed ADD per packet and the overall number of transmissions for SNC, RLNC and LT systems. When we use a SNC scheme includes moderate generation sizes and $w \in [3, 9]$, the optimal choices in terms of the ADD per packet and the overall number of transmissions are $q = 2$ and $q = 2^{32}$, respectively. Also, $q = 2^4$ establishes a trade-off between $q = 2$ and $q = 2^{32}$ for ADD and t_n. However, for $w \leq 3$ and $w > 11$, $q = 2^4$ has a lower ADD per packet. Above all, we can conclude that NC approaches provide a more suitable performance than the provided LT code in with regard to the total number of transmissions and the ADD per packet.

V. Conclusion

This paper has addressed the impact of the field size on two SNC's metrics, i) the average number of transmissions required to decode a fraction of a generation, and ii) the imposed ADD per packet to decode a generation. We have derived the mostly tight lower bounds on the two mentioned metrics. Our results show that SNC's ADD per packet can be significantly reduced through selecting the appropriate w and q.

VI. Acknowledgment

The author would like to thank Esmaeil Delfaraz Pahlevanlo for constructive criticism of the manuscript. This work was supported in part by the AUFF Starting Grant Project AUFF-2017-FLS-7-1, the TuneSCode project granted by the Danish Council for Independent Research (Grant No. DFF 1335-00125), and the SCALE-IoT project granted by the Danish Council for Independent Research (Grant No. DFF 7026-00042B).

References

[1] T. Ho, M. Medard, R. Koetter, D.R. Karger, M. Effros, J. Shi and B. Leong, “A Random Linear Network Coding Approach to Multicast,” *IEEE Trans, on Info. Theory,*vol. 52, pp. 4413 - 4430, 2006.

[2] A. Zarei, P. Pahlevani and M. Davoodi, “On the Partial Decoding Delay of Sparse Network Coding,” *IEEE Comm. Letters* , vol. 22, pp. 1668 - 1671, May 2018.

[3] D. Silva, W. Zeng and F. R. Kschischang, “Sparse network coding with overlapping classes,” *Workshop on Network Coding, Theory, and Applications,* July 2009.

[4] J. Heide, M. V. Pedersen, F. H. P. Fitzek and M. Medard, “On Code Parameters and Coding Vector Representation for Practical RLNC,” *Conf. on Comm. (ICC), 2011 IEEE Int.,* 2011.

[5] G. Pablo, L. E. Daniel and A. Ramon, “A Markov Chain Model for the Decoding Probability of Sparse Network Coding,” *IEEE Trans. on Comm.*, vol. 65, pp. 1675 - 1685, April 2017.

[6] M. Luby, “LT codes,” *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science,*, 2002.

[7] J Heide, MV Pedersen, FHP Fitzek, “Decoding algorithms for random linear network codes”,*Int. Conf. on Research in Networking*, 129-136

[8] S. Feizi, D. E. Lucani, M. Medard, “Tunable Sparse Network Coding”,*In Int. Zurich Sem. on Com. (IZS)*, pp. 107-110, Switzerland, 2012.

[9] J. Kurose and K. Ross, “Computer Networking: A Top-Down Approach (7th Edition)”,*Published by Pearson*, April, 2016.

Amir Zarei received his BSc in Software Engineering from Ilam University, Ilam, Iran (2014), and his MSc in computer science at Institute for Advanced Studies in Basic Sciences (IASBS) in Zanjan, Iran (2018). He also was a visiting researcher at Department of Engineering - Network Computing, Communications and Storage at Aarhus University, Denmark. His research interests include Wireless Communication, Network coding, Approximation and Randomized Algorithms.

Peyman Pahlevani is an Assistant Professor in the department of Computer Science and Information Technology of Institute for Advanced Studies in Basic Science (IASBS), Iran. He received his PhD in Wireless Communication from Aalborg University, Denmark in 2014. Prof. Pahlevani was a visiting researcher at the Computer Science Department of University of California, Los Angeles (UCLA). Moreover, He has been collaborated with different institutes and universities such as MIT, and Porto universities. Beside his academic backgrounds, he extended his skills to solve practical challenges in the area of video communication in his previous carrier, as a Researcher Engineer at AIRTAME company. His research interests are in the area of wireless communication, network coding and its applications, vehicular communications, cooperative networking, and video streaming over WiFi links. Dr. Pahlevani has also served as TPC member for international conferences and as reviewer for high impact journals, such as IEEE Transactions on Vehicular Technology.

Daniel E. Lucani is an Associate Professor at the Department of Engineering at Aarhus University since April 2017 as well as the founder, CEO and lead scientist of the start-up company Chocolate Cloud ApS since June 2014. Dr. Lucani is an IEEE Senior Member, the recipient of the IEEE ComSoc Outstanding Young Researcher Award for the EMEA region in 2015, and the recipient of the Danish Free Research Foundation Sapere Aude Starting Grant. Dr. Lucani received the B.S. summa cum laude (2005) and M.S. with Honors (2006) degrees in electronics engineering from Universidad Simon Bolivar, Caracas, Venezuela, and the Ph.D. degree in electrical engineering (2010) from the Massachusetts Institute of Technology (MIT). Dr. Lucani has published more than 170 scientific papers in international journals and top-ranked international conferences as well as 8 patents and patent applications. His research interests include communications and networks, network coding, information theory, coding theory, distributed storage and computation, and their applications to cloud computing technologies necessary to enable IoT, BigData and 5G applications and services. He was the General Co-Chair of the 2014 International Symposium on Network Coding (NetCod2014), General Chair of European Wireless 2019, and was an Associate Editor of EURASIP Journal on Wireless Communications and Networking (2017-2019). Dr. Lucani was an Associate Professor at Aalborg University (2012-2017), an Assistant Professor at the University of Porto (2010-2012).