On the Delay Distribution of Random Linear Network Coding

M. Nistor*, D. E. Lucani*, T. T. V. Vinhoza*, R. A. Costa† and J. Barros*

*Instituto de Telecomunicações, Faculdade de Engenharia, Universidade do Porto, Rua Dr Roberto Frias s/n, 4200-465, Porto, Portugal
†Instituto de Telecomunicações, Faculdade de Ciências, Universidade do Porto, Rua Campo Alegre 1023/1055, 4169-007, Porto, Portugal
{mnistor, dlucani, vinhoza, rfcosta, jbarros}@fe.up.pt.

***Abstract*—A fundamental understanding of the delay behavior of network coding is key towards its successful application in real-time applications with strict message deadlines. Previous contributions focused mostly on the average decoding delay, which although useful in various scenarios of interest is not sufficient for providing worst-case delay guarantees. To overcome this challenge, we investigate the entire delay distribution of random linear network coding for any field size and arbitrary number of encoded symbols (or generation size). By introducing a Markov chain model we are able to obtain a complete solution for the erasure broadcast channel with two receivers. A comparison with Automatic Repeat reQuest (ARQ) with perfect feedback, round robin scheduling and a class of fountain codes reveals that network coding on $\mathbf{GF}(2^4)$ offers the best delay performance for two receivers. We also conclude that $\mathbf{GF}(2)$ induces a heavy tail in the delay distribution, which implies that network coding based on XOR operations although simple to implement bears a relevant cost in terms of worst-case delay. For the case of three receivers, which is mathematically challenging, we propose a brute-force methodology that gives the delay distribution of network coding for small generations and field size up to $\mathbf{GF}(2^4)$.**

***Index Terms*—network coding, delay, transport protocols, probabilistic analysis**

I. Introduction

After a decade of research the throughput benefits and robustness properties of network coding [3] have been well established for highly dynamic networks. This research effort has resulted in real-life protocols for wireless mesh networks [4] and peer-to-peer content distribution [5], among other applications [6]. Arguably less well understood is the delay behavior of network coding, which is of particular relevance if network coding techniques are to be employed equally successfully in real-time applications such as live streaming or automatic control systems. The design of such systems requires knowledge not just of the average decoding delay, which has already been studied to some extent, but also of the worst-case delay, which can be inferred from the complete delay distribution. Providing such a characterization of network coding delay for various scenarios of interest is the goal of this paper.

Consider the broadcast scenario depicted in Figure 1. A source wants to transmit M information symbols $s_1, s_2, .., s_M$ to a set of receivers R, with $N = |R|$. Each receiver observes the output of an independent erasure channel. To overcome the impairments of the channels while serving all of the receivers simultaneously, the source is allowed to mix the incoming symbols and send out linear combinations following the basic rules of random linear network coding (RLNC) [7]. More specifically, the encoder mixes $s_1, s_2, .., s_M$ and outputs coded symbols of the form $\sum_{j=1}^{M} \alpha_j s_j$. The coding coefficients $\alpha_1, \ldots, \alpha_M$ are independently and randomly selected from $\mathbf{GF}(q)$. Coded symbols are transmitted over independent erasure channels. A symbol is erased with probability ϵ_i on channel i, $\forall i = 1, 2 \ldots N$. After collecting M linearly independent combinations, each decoder is able to reconstruct the encoding matrix by means of Gaussian elimination thus recovering the original information symbols. Feedback is limited to one acknowledgment for the reception of all M symbols. Our main figure of merit is the decoding delay D_i of each receiver R_i, which is defined as the total number of time slots required for R_i to decode the M information symbols. Seeking to characterize the probability distribution of the decoding delay of RLNC for the aforementioned communications scenario, we make the following contributions:

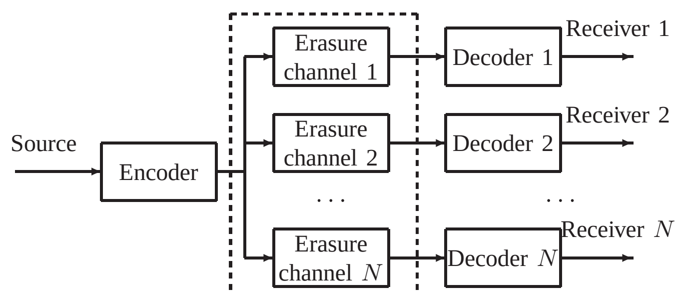


Fig. 1: System model.

- *Fundamental Analysis:* We propose a Markov chain ap-

M. Nistor, D. E. Lucani, T. T. V. Vinhoza and J. Barros ({mnistor, dlucani, vinhoza, jbarros}@fe.up.pt) are with the Instituto de Telecomunicações, Departamento de Engenharia Electrotécnica e de Computadores, Faculdade de Engenharia da Universidade do Porto, Portugal. R. A. Costa (rfcosta@fe.up.pt) is with the Instituto de Telecomunicações, Departmento de Ciência de Computadores, Faculdade de Ciências, Universidade do Porto, Portugal. Parts of this work have been presented at the Information Theory and Applications Workshop (ITA 2010) [1] and at the IEEE Symposium on Network Coding (NetCod 2010) [2]. This work was partially funded by the Fundação Calouste Gulbenkian under grant 103905, by Fundação para a Ciência e Tecnologia (Portuguese Foundation for Science and Technology) under grant SFRH-BD-27273-2006 and by the European Commission under grant FP7–INFSO–ICT–215252 (N-Crave Project).

proach that enables us to derive the aforementioned delay distribution for the case of two receivers and independent erasure channels. The one receiver case follows immediately as a special case.

- *Performance Evaluation and Comparison:* In the case of two receivers, we demonstrate that RLNC outperforms Automatic Repeat reQuest (ARQ) with perfect feedback for a Galois field larger or equal to $\mathbf{GF}(2^2)$. The performance of RLNC is also shown to be superior to that of Luby Transform (LT) codes (a class of fountain codes [8]) and round robin scheduling — irrespective of the field size. A similar analysis for the one receiver case reveals that the delay distribution of RLNC is close to that of ARQ schemes with perfect feedback already for a relatively small field size, e.g., $\mathbf{GF}(2^4)$. Our results also show that opting for network coding over $\mathbf{GF}(2)$, which is convenient for its low computational complexity, bears the cost of a heavy tail in the delay distribution.
- *Brute-Force Analysis:* Since a generalization of the proposed delay analysis becomes difficult already for three receivers, we propose an alternative brute-force methodology. The key idea is to take a large number of fixed erasure patterns and measure for every erasure pattern the delay of all possible encodings or sets of linear combinations of information symbols. Using a computing cluster, we are able to demonstrate that this approach is feasible for small field size and limited number of coded symbols (i.e. small *generation size*), yielding a characterization of the delay distribution under such scenarios.

The remainder of the paper is organized as follows. Section II provides an overview of relevant related work. The analysis of RLNC delay is given in Section III, which explains the Markov chain model and discusses in detail the cases of two receivers and one receiver. Section IV compares the delay performance of RLNC against three reference schemes, namely, ARQ with perfect feedback, round robin scheduling, and a LT code. Section V describes the proposed brute-force methodology for analyzing the delay performance for scenarios with three receivers. The paper concludes with Section VI.

II. Related Work

Network coding first appeared as an information theoretic multicast problem under which the decoding delay is of no importance [3]. The algebraic framework in [9] and the emergence of Random Linear Network Coding (RLNC) [7] led to practical applications in which the nodes in the network generate linear combinations of information symbols using random coefficients in a Galois field, as described in Section I. Typically, these random coefficients are sent in the header of the packet that carries the coded symbols, which enables the receiver to learn the coding matrix and recover the original information by Gaussian elimination [10].

Most results that address network coding delay take into account only the average delay performance. In [11] this metric is computed for a broadcast scenario with multiple receivers and then compared to round robin scheduling. The average delay of network coding is shown to decrease with a rising number of receivers. Likewise, the work described in [12] provides results for the average delay yet includes also the energy and throughput performance, as well as a comparison with standard ARQ schemes. The average delay for a time-division duplexing scheme is provided in [13] in a broadcast network and the case of one receiver as a function of the field size of RLNC is characterized without addressing the actual delay distribution. The contribution in [14] is focused on the average delay performance of systematic network coding with small field sizes, once again in a broadcast network.

When feedback is available, more sophisticated mechanisms can be used to broadcast a data stream to multiple receivers. A typical approach is for the receiver to send an acknowledgement once it decodes a complete set of coded symbols (or generation). The work presented in [15] considers a limited number of information symbols and characterizes the average delay on a line network. It follows that in this special case the delay performance is concentrated around its expectation. Fountain codes, such as LT Codes [8], form yet another class of codes that provide reliable communication and throughput efficiency by acknowledging the successful decoding of the original message block.

Naturally, it is possible to explore feedback in a more elaborate way. For a half duplex channel, [16] combines the idea of incremental redundancy with network coding, i.e., using feedback to request additional coded symbols. The work therein proves that there exists an optimal number of coded symbols that can be transmitted before the sender receives an acknowledgment. There, the optimum is defined in terms of the average delay required to complete the transmission of a generation of information symbols. What each receiver feeds back is the number of the degrees of freedom that are still required for decoding the entire generation successfully. An extension to this work is presented in [12], which offers a complete delay and energy characterization of the aforementioned coding scheme. Online network coding mechanisms for random arrivals of information symbols are considered in [17], [18], [19], [20]. These contributions assume that information symbols are combined dynamically using a sliding window mechanism, whereby the destination node acknowledges the degrees of freedom it receives. Unnecessary information symbols are thus dropped from the sender queue. The delay implications of this online network coding mechanism are addressed in [21] and [22].

Since most results in the literature are based on the average delay, the worst case delay performance is still not well understood. Previous work on worst case delay includes [23] which uses deterministic network calculus to describes delay service guarantees for a packet at an intermediate node, however the network model therein does not admit packet losses. By considering erasure channels our work offers results for a more realistic network model. This scenario of a source broadcasting packets to several receivers over erasure channels was analyzed in [24], where it was shown that the minimization of delay for this broadcast scenario is NP-hard. Thus, knowing the complete delay distribution is clearly a step-forward, as it allows us to give upper and lower delay bounds for such a model.

III. Main Result

In this section we show that the delay distribution problem can be cast in a Markov chain model. This approach allows us to obtain the delay distribution for RLNC. More specifically, we provide an exact characterization for two receivers over independent erasure channels. The one-receiver instance is obtained as a special case. The following definitions are useful.

Definition 1: The *knowledge space* (or simply the *knowledge*) K_l of a receiver l at a given time t is defined as the linear span of the linear combinations of symbols $s_1, s_2, .., s_M$ received by l until time t.

Definition 2: We say that a node has k *degrees of freedom (dofs)* if the dimension of its knowledge space is k.

Definition 3: We say that a linear combination is *innovative* to receiver l at time t if it does not belong to K_l.

We also require the following events related to the arrival of a new coded symbol or linear combination.

Definition 4: Let E_K denote the event that occurs when a received linear combination is innovative given the knowledge space K_l of receiver l.

Definition 5: Let E_K^{nz} denote the event that occurs when a received linear combination, with a coding vector that is not all zeros, is *not* innovative with respect to the knowledge space K_l of receiver l.

Definition 6: Let Z denote the event that corresponds to the arrival of a linear combination with an all-zeros coding vector.

Since in our broadcast setting the source is common to all receivers, it is likely that subsets of receivers have the same information at any given time. We formalize this intuition as follows.

Definition 7: We say that a subset of receivers $L \subseteq R_l, L \neq \emptyset$ and $|L| > 1$ share the *common knowledge* C_L at a given time t if $C_L = \cap_{\{l \in L\}} K_l$ at time t.

A. General Case

We consider the scenario depicted in Figure 1, in which a source wants to transmit M symbols to N receivers. The transmission adds a degree of freedom (dof) to the knowledge space of a receiver if the channel does not erase it and the sent linear combination is linearly independent of all previously received linear combinations. We can describe this process by means of a Markov chain model.

A Markov chain model is defined by a set of states and a set of transitions with given probabilities. In our case, a state consists of the numbers of dofs at each receiver, the number of dofs for the common knowledge space of pairs of receivers, the number of dofs for the common knowledge space of groups of 3 receivers, and so forth until we reach the number of dofs for the common knowledge space of all receivers. The dependence between the receivers inherent to the broadcast scenario is captured by these state variables of common knowledge.

Each state is described by a set of elements as shown in Figure 2. By $(i_1, i_2, \ldots, i_N)$ we represent the dofs for each of the N receivers, with $i_l = \dim(K_l)$, where K_l is the knowledge of receiver l, denoted here as R_l. We write

$$\begin{aligned} c_{1,2} &= \dim(C_{1,2}) \\ &\ldots \\ c_{1,2,\ldots,N} &= \dim(C_{1,2,\ldots,N}) \end{aligned}$$

for the common knowledge between each combination of $2, 3, \ldots, N$ receivers. The total number of indexes of a state is given by the expression $\sum_{\gamma=0}^{N-1} \binom{N}{\gamma} = 2^N - 1$, which results in a total number of states of $(M+1)^{2^N-1}$. The first state of the model represents 0 dof for all receivers, naturally the common dofs are also 0, hence the state can be represented as $(0, 0, ..., 0)$.

A transition to other states depends on the previous state, on the set of receivers for which the coded symbol was correctly received and the subset of nodes that obtain an innovative linear combination. In every time slot a transition occurs. The maximum dofs at each receiver is reached when M linear independent information symbols are received. When all nodes receive M dofs, they all share the same knowledge. Thus, there exists an absorbing state, which is $(M, M, ..., M)$. A state β of a Markov chain is called absorbing if it has a transition probability $p_{\beta,\beta} = 1$, which implies that the process never changes state once it reaches state β.

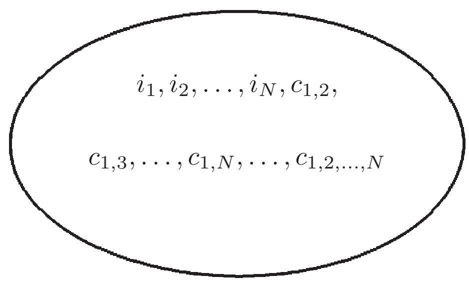


Fig. 2: General state for the N-receiver case.

Transition Probability Matrix: As the state space is finite, we can represent the transition probability distribution by the transition matrix $\mathbf{T}$, whose (u, v) element gives the probability of going from state u to state v. Since Markov chain is stationary, the transition matrix $\mathbf{T}$ does not change with time.

From the $(M+1)^{2^N-1}$ states we discard those never entered by the process. We call these states invalid states and focus on the valid ones. We denote by A the number of valid states. Notice that $A \leq (M+1)^{2^N-1}$. For the lower bound, the number of valid states should be greater than $(M+1)^N$, which corresponds to the total number of N-tuples representing the individual knowledges of the receivers for M transmitted information symbols. Hence, A is bounded by two exponential (on N) growth terms:

$$(M+1)^N \leq A \leq (M+1)^{2^N-1}. \tag{1}$$

Lemma 1: For the $N = 2$ receiver case, the number of valid states is the solution of a difference equation. For $M \geq 2$,

$A(M,N)$ is given by:

$$A(M,2) = 10 + \frac{47}{6}(M-2) + 2(M-2)^2 + \frac{1}{6}(M-2)^3. \quad (2)$$

The proof can be found in Appendix.

The general expression for the transition probability matrix of size $A \times A$ is given in (3).

$$\mathbf{T} = \begin{bmatrix} p_{1,1}(1) & p_{1,2}(1) & ... & p_{1,A}(1) \\ \vdots & ... & \vdots & \vdots \\ p_{A,1}(1) & p_{A,2}(1)... & p_{A,A-1}(1) & p_{A,A}(1) \end{bmatrix}, \quad (3)$$

where $p_{j,a}(1)$ denotes the probability of arriving at state a after one step when the chain starts in state j. Here, state 1 corresponds to $(0,0,\dots,0)$ and state A corresponds to $(M,M,,\dots,M)$. As state A is an absorbing state, we have that $p_{A,A}(1) = 1$ and $p_{A,a}(1) = 0$, $\forall a = 1,2,...,A-1$.

A k-step transition probability matrix can be computed as $\mathbf{T}^k$, i.e., the k-th power of the transition matrix. This k-step transition probability matrix represents the probability of arriving at each of the states in k transitions (time slots). The expression for matrix $\mathbf{T}^k$ is given by (4).

$$\mathbf{T}^k = \begin{bmatrix} p_{1,1}(k) & p_{1,2}(k) & ... & p_{1,A}(k) \\ \vdots & ... & \vdots & \vdots \\ 0 & ... & 0 & 1 \end{bmatrix}, \quad (4)$$

where $p_{j,a}(k)$ denotes the probability of reaching state a in k steps after starting from state j. We are particularly interested in the case of $p_{1,A}(k)$ because it describes the probability that the information symbols are successfully decoded after k time slots.

Decoding Delay: Our goal is to derive a probability distribution for the decoding delay, i.e., determining $P(D=k)$ for M symbols, where $k \geq M$ represents the number of time slots needed to decode the information. Let us formalize our definition of decoding delay and its link to the Markov chain model.

A decoding delay D of k time slots indicates that exactly k time slots are required for all receivers to decode the information, i.e., to transition to state $(M,...,M)$ of the Markov chain for the first time. The probability of $P(D \leq k) = p_{1,A}(k)$, which is the probability of arriving in state $(M,...,M)$ of the Markov chain given that the system started at state $(0,...,0)$. Thus, the probability of decoding in exactly k time slots is given by $P(D=k) = P(D \leq k) - P(D \leq k-1)$.

Computational Complexity: For large N, the model requires high computational complexity, because the number of valid states A increases exponentially with the number of receivers. The impact is evident when computing the transition probability matrix and managing the operations with matrices. We want to find $P(D \leq k)$, which requires us to multiply a $A \times A$ matrix up to $(k-1)$ times. For instance, $N=3$ requires already $O(M^7)$ states, which means a state has 7 elements, 3 for the knowledge of receivers and 4 for the common knowledge. This is still feasible for the two-receiver case, as we discuss in Section III-B.

Particular cases: Two special cases are observed when the channels assign specific values for the erasure probability and a third one emerges when the receivers are arranged in a different way:

- Perfect channels: It is the case when no erasures occur. A state is represented only by the knowledge of one receiver and the Markov chain has $(M+1)$ states.
- Noisy channels: This situation corresponds to channels with high erasure probabilities, whose delay can be approximated by independent random variables. The cumulative distribution function of the N random variables is then the product of the cumulative distribution function of delay for each one of the N receivers. Hence, the Markov chain reduces to the case of one receiver and has $(M+1)$ states.
- Degraded channels: This case refers to the situation where the channel of receiver i is a degraded version of the channel of $i-1$, $\forall i = 2,3,...N$. In this case, the maximum decoding delay is always the delay of receiver N. The Markov chain reduces again to the case of only one receiver and has $(M+1)$ states.

In these particular situations, we can observe that our Markov chain model is represented by one receiver. The total number of states equals $M+1$. For that reason, the model can be easily extended to accommodate an arbitrary number of receivers.

B. The case of two receivers

For this case we denote by K_1 the knowledge of the first receiver, by K_2 the knowledge of the second receiver and by $C = K_1 \cap K_2$ the common knowledge of both receivers. In this case, each state is described by 3 elements (i_1, i_2, c), with $i_1 = \dim(K_1)$, $i_2 = \dim(K_2)$ and $c = \dim(C)$. The first state corresponds to $(0,0,0)$. The elements (i_1, i_2, c) evolve in each slot and the final state is defined by (M,M,M). The following theorem states the possible transition probabilities for the two receiver case. Let d_1 and d_2 denote the dimensions of the *non* common knowledge of R_1 and R_2, respectively. This means that $d_a = \dim(K_a \backslash C)$ and $i_a = c + d_a$, where $i_a = \dim(K_a)$, $c = \dim(C)$ and $a \in \{1,2\}$.

Theorem 1: In the Markov model RLNC over $\mathbf{GF}(q)$ with two receivers, there exist at most 7 states to which state (i_1, i_2, c) can transit to with non-zero probability. The transition probabilities are given by (5), where

- $P(E^{nz}_{K_a} \cup Z) = (q^{-M+c+d_a})$,
- $P(E_{K_a} \cap E_{K_a \cup K_b}) = 1 - q^{-M+c+d_a+d_b}$,
- $P(E_{K_a} \cap E_{K_b} \cap E_{K_a \cup K_b}) = 1 - q^{-M+c+d_a+d_b}$,
- $P(E_{K_a} \cap E^{nz}_{K_b}) = \frac{(1-q^{-M+c+d_a})(q^{-M+c+d_b} - q^{-M+c})}{(1-q^{-M+c})}$,
- $P(E^{nz}_{K_a} \cap E^{nz}_{K_b} \cup Z) = (q^{-M+c+d_b}) - \left[\frac{(1-q^{-M+c+d_a})(q^{-M+c+d_b} - q^{-M+c})}{1-q^{-M+c}}\right]$,
- $P(E_{K_a} \cap E_{K_b} \cap E^{nz}_{K_a \cup K_b}) = \left[\frac{(1-q^{-M+c+d_a})(1-q^{-M+c+d_b})}{(1-q^{-M+c})} - (1 - q^{-M+c+d_a+d_b})\right]$,

with $a, b \in \{1,2\}$ and $a \neq b$.

Proof: The first part of the proof is combinatorial in nature and relies on considering all possible events, namely (a) independent channels suffering erasures, and (b) the coded symbol adding a dof, (c) the coded symbol not adding a dof, or (d) a coding vector of all zeros with respect to the vector

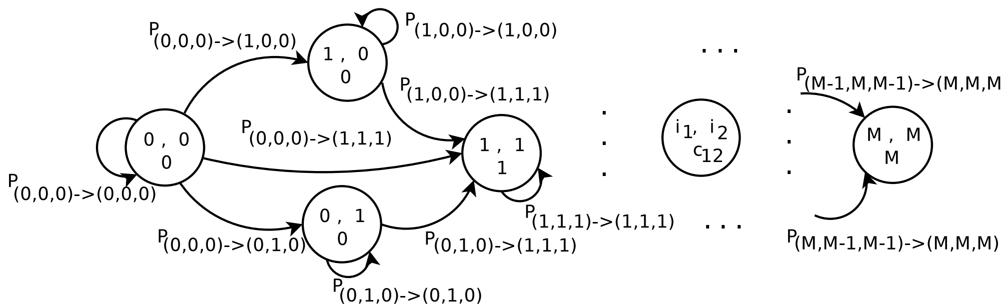


Fig. 3: Markov Chain for two-receiver case.

$$P_{(i_1,i_2,c)\to(i'_1,i'_2,c')}= \tag{5}$$
$$\begin{cases}\epsilon_1\epsilon_2+\epsilon_1(1-\epsilon_2)P(E^{nz}_{K_2}\cup Z)+\epsilon_2(1-\epsilon_1)P(E^{nz}_{K_1}\cup Z)+ & \\ \quad +(1-\epsilon_1)(1-\epsilon_2)P(E^{nz}_{K_1}\cap E^{nz}_{K_2}\cup Z), & \text{if 1) } i'_1=i_1, i'_2=i_2, c'=c \\ (1-\epsilon_1)(1-\epsilon_2)P(E_{K_1}\cap E_{K_2}\cap E_{K_1\cup K_2}), & \text{if 2) } i'_1=i_1+1, i'_2=i_2+1, c'=c+1 \\ (1-\epsilon_1)(1-\epsilon_2)P(E_{K_2}\cap E^{nz}_{K_1})+\epsilon_1(1-\epsilon_2)P(E_{K_1}\cap E_{K_2}\cap E^{nz}_{K_1\cup K_2}), & \text{if 3) } i'_1=i_1, i'_2=i_2+1, c'=c+1 \\ \epsilon_1(1-\epsilon_2)P(E_{K_2}\cap E_{K_1\cup K_2}), & \text{if 4) } i'_1=i_1, i'_2=i_2+1, c'=c \\ (1-\epsilon_2)(1-\epsilon_1)P(E_{K_1}\cap E^{nz}_{K_2})+\epsilon_2(1-\epsilon_1)P(E_{K_1}\cap E_{K_2}\cap E^{nz}_{K_1\cup K_2}), & \text{if 5) } i'_1=i_1+1, i'_2=i_2, c'=c+1 \\ \epsilon_2(1-\epsilon_1)P(E_{K_1}\cap E_{K_1\cup K_2}), & \text{if 6) } i'_1=i_1+1, i'_2=i_2, c'=c \\ (1-\epsilon_1)(1-\epsilon_2)P(E_{K_1}\cap E_{K_2}\cap E^{nz}_{K_1\cup K_2}), & \text{if 7) } i'_1=i_1+1, i'_2=i_2+1, c'=c+2. \end{cases}$$

spaces $K_1, K_2, K_1 \cup K_2$. We also observe the fact that several combinations generate the same transition and that at every time slot the source can provide at most one dof to each receiver.

Let $f = \dim(K_1 \cup K_2) \leq M$. Note that $f = i_1 + i_2 - c = d_1 + d_2 + c$ or equivalently, $c = i_1 + i_2 - f$. Let us also define $f' = \dim(K_1 \cup K_2 \cup v)$, where v is the incoming coded symbol. Note that the transitions to i'_a based on i_a are straight-forward, i.e. either $i'_a = i_a$, because of an erasure or the fact that no additional dof is provided to K_a, or $i'_a = i_a + 1$, if there is no erasure and the coded symbol is innovative to K_a. When an incoming coded symbol v is innovative to $K_1 \cup K_2$ this implies that $f' = f + 1$. Conversely, if it does not provide a dof or the coding vector is all-zero then $f' = f$. Using this knowledge, we can determine the values of c' based on the transition to i'_1, i'_2 and f'. Thus, there are 3 possible values for c', namely, $c, c+1, c+2$. If both receivers maintain the same dofs after the transition, clearly $c' = c$ because $f' = f$, $i'_1 = i_1$ and $i'_2 = i_2$. When an incoming coded symbol adds a dof to $K_1 \cup K_2$, i.e., $f' = f + 1$, then we have two possibilities: (i) $c' = c$ corresponding to the case in which only one receiver, say a, gets a new dof, because $i'_a = i_a + 1$, $i'_b = i_b$, and $b \neq a$, or (ii) $c' = c+1$ in case both receivers get a new dof, because $i'_1 = i_1 + 1$ and $i'_2 = i_2 + 1$.

When the incoming coded symbol is not innovative to $K_1 \cup K_2$, i.e., $f' = f$, then (i) $c' = c+1$ if only one receiver, say a, gets a new dof because $i'_a = i_a + 1$, $i'_b = i_b$, and $b \neq a$ or (ii) $c' = c+2$ if both receivers get a new dof, because $i'_1 = i_1 + 1$ and $i'_2 = i_2 + 1$.

Thus, a state has at most 7 transition states with non-zero transition probability, including the case of self-transition. The transition probabilities match the previously described events, combining the effect of erasures and innovativeness of the coded symbols. This concludes the first part of the proof.

Let us now prove the expressions for the probabilities of the different events for a coded symbol in terms of the knowledge at the receivers. The probability that a coded symbol is innovative with respect to the knowledge space K_a of receiver a is given by

$$P(E_{K_a}) = P(E_{K_a} \cap E_C) = 1 - q^{-M+c+d_a}. \tag{6}$$

The event that a coded symbol is not innovative or that the coding vector is all-zero is the negation of an innovative coded symbol arriving at the receiver. Thus, we get

$$P(E^{nz}_{K_a} \cup Z) = q^{-M+c+d_a}. \tag{7}$$

The probability of a coded symbol not adding a dof to one receiver's knowledge space while it is already a part of the other receiver's knowledge space is given by

$$\begin{aligned} P(E_{K_a} \cap E^{nz}_{K_b}) &= P(E_{K_a} \cap E^{nz}_{K_b} \cap E_C) \\ &= P(E_{K_a}|E^{nz}_{K_b} \cap E_C)P(E^{nz}_{K_b}|E_C)P(E_C) \\ &= P(E_{K_a}|E_C)P(E_C)P(E^{nz}_{K_b}|E_C) \\ &= \frac{1-q^{-M+c+d_a}}{1-q^{-M+c}}(1-q^{-M+c})P(E^{nz}_{K_b}|E_C). \end{aligned}$$

Given that

$$P(E^{nz}_{K_b}|E_C) = 1 - P(E_{K_b}|E_C) = 1 - \frac{1-q^{-M+c+d_b}}{1-q^{-M+c}},$$

then

$$P(E_{K_a} \cap E^{nz}_{K_b}) = \frac{(1-q^{-M+c+d_a})(q^{-M+c+d_b} - q^{-M+c})}{(1-q^{-M+c})}. \tag{8}$$

The probability that a coded symbol is not innovative for both receivers or that the coding vector is all-zero is given by the expression

$$\begin{aligned}
&P(E^{nz}_{K_a} \cap E^{nz}_{K_b} \cup Z)\\
=\ & 1 - [P(E_{K_a} \cap E_{K_b}) + P(E_{K_a} \cap E^{nz}_{K_b}) + P(E_{K_b} \cap E^{nz}_{K_a})]\\
=\ & P(E_{K_b}) - P(E^{nz}_{K_b} \cap E_{K_a}).
\end{aligned}$$

Taking into consideration (7) and (8), the final result is given by

$$\begin{aligned}
&P(E^{nz}_{K_a} \cap E^{nz}_{K_b} \cup Z)\\
&= \left(q^{-M+c+d_b}\right) - \left[\frac{(1-q^{-M+c+d_a})(q^{-M+c+d_b} - q^{-M+c})}{1-q^{-M+c}}\right]. \quad (9)
\end{aligned}$$

The probability that a coded symbol is innovative for both receivers while not adding a dof to $K_a \cup K_b$ is given by the expression $P(E_{K_a} \cap E_{K_b} \cap E_{K_a \cup K_b}) = 1 - q^{-M+c+d_a+d_b}$. For the case of a coded symbol that adds a dof to the knowledge space of both receivers but does not add a dof to $K_a \cup K_b$ we can write

$$\begin{aligned}
&P(E_{K_a} \cap E_{K_b} \cap E^{nz}_{K_a \cup K_b})\\
&= P(E_{K_a} \cap E_{K_b} \cap E_C) - P(E_{K_a} \cap E_{K_b} \cap E_{K_a \cup K_b})\\
&\quad P(E_{K_a} | E_{K_b} \cap E_C) P(E_{K_b} | E_C) P(E_C) -\\
&\quad - P(E_{K_a} \cap E_{K_b} \cap E_{K_a \cup K_b})\\
&= P(E_{K_a} | E_C) P(E_{K_b} \cap E_C) - P(E_{K_a} \cap E_{K_b} \cap E_{K_a \cup K_b}).
\end{aligned}$$

The probability of a coded symbol being innovative to K_a, given that is also innovative to the common knowledge of both receivers is

$$P(E_{K_a} | E_C) = \frac{P(E_{K_a} \cap E_C)}{P(E_C)} = \frac{1 - q^{-M+c+d_a}}{1 - q^{-M+c}}. \quad (10)$$

The probability $P(E_{K_a} \cap E_{K_b} \cap E^{nz}_{K_a \cup K_b})$ is obtained by combining (6) and (10).

$$\begin{aligned}
&P(E_{K_a} \cap E_{K_b} \cap E^{nz}_{K_a \cup K_b}) \quad (11)\\
&= \left[\frac{(1-q^{-M+c+d_a})(1-q^{-M+c+d_b})}{(1-q^{-M+c})} - (1-q^{-M+c+d_a+d_b})\right].
\end{aligned}$$

The probability that a coded symbol is innovative with respect both to K_a and to $K_a \cup K_b$ is equivalent to the probability that is innovative with respect to K_a, K_b and $K_a \cup K_b$ yielding

$$\begin{aligned}
&P(E_{K_a} \cap E_{K_a \cup K_b})\\
&= P(E_{K_a} \cap E_{K_b} \cap E_{K_a \cup K_b}) = (1 - q^{-M+c+d_a+d_b}). \quad (12)
\end{aligned}$$

This concludes the proof. ■

Thus, we can compute the probability distribution of the decoding delay (4). But first let us provide some intuition on the transition probabilities for each state (i_1, i_2, c) to state (i'_1, i'_2, c') given by the following 7 cases:

1) $i'_1 = i_1$, $i'_2 = i_2$, $c' = c$: this case includes the events that (i) both channels induce erasures, (ii) one channel induces erasure and the receiver corresponding to the other channel gets a coded symbol that does not increase the dofs of its knowledge or was encoded with all zero coefficients, or (iii) the coded symbol is not innovative for both receivers or was encoded with all zero coefficients.
2) $i'_1 = i_1 + 1$, $i'_2 = i_2 + 1$, $c' = c + 1$: both receivers get innovative coded symbols and the transmitted coded symbol is innovative with respect to $K_1 \cup K_2$.
3) $i'_1 = i_1$, $i'_2 = i_2 + 1$, $c' = c + 1$: this case considers the events that (i) both receivers get the coded symbol and it is innovative for R_2 but not to R_1, (ii) only R_2 receives the coded symbol and it is innovative for R_2 but does not add a dof to K_1, or (iii) only R_2 receives the coded symbol and the coded symbol adds a dof to both K_1 and K_2 but is not innovative for $K_1 \cup K_2$.
4) $i'_1 = i_1$, $i'_2 = i_2 + 1$, $c' = c$: R_2 gets a new coded symbol that is innovative for K_1, K_2 and $K_1 \cup K_2$, and the channel associated to R_1 suffers an erasure.
5) $i'_1 = i_1 + 1$, $i'_2 = i_2$, $c' = c + 1$: this case is symmetric to 3).
6) $i'_1 = i_1 + 1$, $i'_2 = i_2$, $c' = c$: this case is symmetric to 4).
7) $i'_1 = i_1 + 1$, $i'_2 = i_2 + 1$, $c' = c + 2$: both R_1 and R_2 receive the coded symbol. The coded symbol is innovative to both receivers but is not innovative for $K_1 \cup K_2$.

Note that in case 7) the common part increases by 2 in a single time slot. This happens if the knowledge of both receivers is increased, i.e., the coded symbol is innovative to both receivers, but it is already part of the common knowledge space. For instance, when $M = 2$ we can have the following situation. Before receiving a new coded symbol, R_1 has $K_1 = \{s_1\}$ and R_2 has $K_2 = \{s_2\}$, therefor $K_1 \cap K_2 = \emptyset$. This is represented by state $(1, 1, 0)$, i.e., each receiver has one dof and they share no common knowledge. In the next time slot, a new coded symbol is transmitted that is a linear combination of s_1 and s_2. The knowledge of both receivers increases by one once this coded symbol is received, while the common part is increased by 2 since $\dim(K_1 \cap K_2) = \dim(s_1, s_2) = 2$. This state is represented by $(2, 2, 2)$.

C. The case of one receiver

Suppose now that the source wants to send M information symbols to a receiver. A similar model Markov model, focusing on average decoding delay, is studied in [13]. The transitions from i_1 to state i'_1 are given by the following two cases:

- $i'_1 = i_1$: the coded symbol suffers an erasure or the coded symbol is received but it does not add a new dof to the knowledge of the receiver.
- $i'_1 = i_1 + 1$: a coded symbol is received and the receiver gets a new dof.

The associated transition probabilities are given by

$$P_{(i_1)\to(i'_1)} = \begin{cases} \epsilon_1 + (1-\epsilon_1) P(E^{nz}_{K_1} \cup Z), & \text{if } i'_1 = i_1 \\ (1-\epsilon_1) P(E_{K_1} \cup Z), & \text{if } i'_1 = i_1 + 1 \end{cases}$$

$$= \begin{cases} \epsilon_1 + (1-\epsilon_1) q^{-M+i_1}, & \text{if } i'_1 = i_1 \\ (1-\epsilon_1)(1 - q^{-M+i_1}), & \text{if } i'_1 = i_1 + 1. \end{cases}$$

As in the general case, the probability $P(D \leq k)$ is given by the element $p_{1,A}(k)$ from $\mathbf{T^k}$, where $A = M + 1$. The $(M+1)$-th state is associated with state $i_1 = M$ and the first state is associated to state $i_1 = 0$.

IV. Insights and Practical Implications

Having derived analytical expressions for the delay distribution of RLNC, we are now ready to discuss some of the insights they offer. To this end, we compare the decoding delay of RLNC with three well established transmission schemes:

- **ARQ**: The sender will transmit every symbol, whose reception is not acknowledged by each receiver. This scheme is known to achieve optimal throughput and minimum delay over the erasure channel with one receiver (see [19]).
- **LT codes**: This class of rateless codes is well known to ensure reliability and optimum throughput over erasure channels (see [25]). Feedback is only used for acknowledging the successful reception of all information symbols. The encoder performs combinations of information symbols using an appropriate degree distribution in order to minimize the number of redundant code symbols. Usually, the degree is chosen from a robust soliton distribution, which requires two additional parameters to be set, namely a constant, $const < 1$, and a bound δ on the decoding failure probability [26].
- **Round robin**: This mechanism is well known to yield optimum throughput in broadcast scenarios [11] [27], where the receivers experience the same probability of erasure. It is a scheduling scheme with minimalistic feedback, where the source sends the information symbols in round-robin fashion until the receivers announce the reception of all M information symbols.

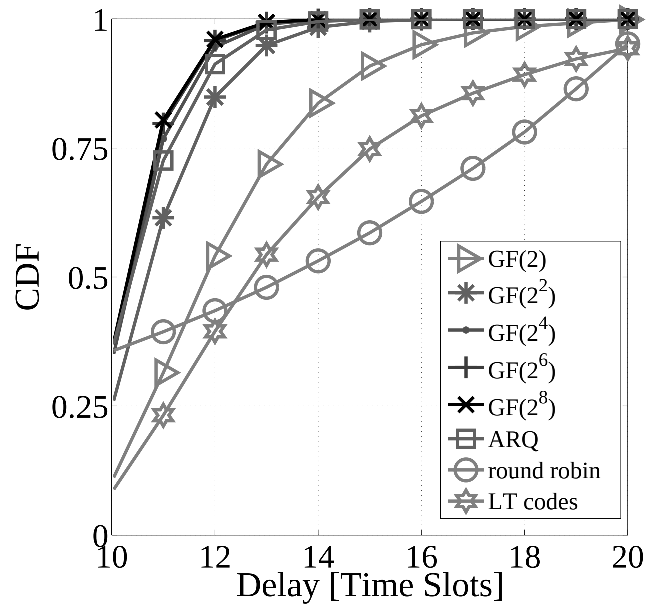
The numerical results for these three techniques were obtained through simulation. For the ARQ scheme, the source sends the same information symbol until it receives ACKs of successful reception from all the receivers. In the case of the LT codes we use the robust soliton distribution, with $const = 0.01$ and $\delta = 0.7$. As our analysis focuses on small values of M, the performance of LT codes is not sensitive to the values of $const$ and δ. The round robin experiences the same minimalistic feedback as our model by only acknowledging the correctly received information symbols at all the receivers. Using these reference systems as a benchmark for the delay performance of RLNC, we now discuss the two receiver case in detail.

A. Two receivers

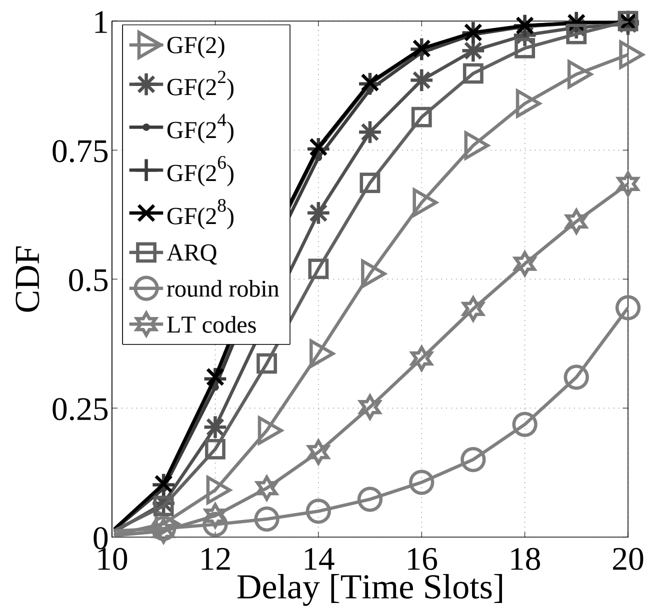
We characterize the delay distribution of RLNC by means of the transition probability matrix defined in (4). The analysis is carried out for different field sizes, erasure probabilities and number of information symbols.

Field size: In order to study the effect of the field size, we assume that the channels have identical erasure probability. The number of transmitted information symbols is fixed to 10 and we vary the field size from $\mathbf{GF}(2)$ to $\mathbf{GF}(2^8)$. A comparison with ARQ, round robin scheduling and LT codes is also included. For the case of a channel with small erasure probability (equal to 0.05), the findings are shown in Figure 4(a), where it is possible to see that RLNC in $\mathbf{GF}(2^4)$ outperforms ARQ, and RLNC over $\mathbf{GF}(2)$ outperforms LT codes.

For a scenario with higher erasure probability, Figure 4(b) shows that RLNC outperforms ARQ when done in fields as small as $\mathbf{GF}(2^2)$. LT codes and the scheduling scheme are outperformed by RLNC.



(a) $\epsilon = 0.05$



(b) $\epsilon = 0.2$

Fig. 4: The effect of the field size in the two-receiver case. Here we have $M = 10$, $\epsilon = 0.05$ (a), $\epsilon = 0.2$ (b) and various field sizes.

For field sizes larger than $\mathbf{GF}(2^4)$, the delay performance is similar, as evidenced by both figures. Thus, in the following we take $\mathbf{GF}(2^4)$ as a representative for higher fields. We conclude from here that for $\mathbf{GF}(2^4)$ and a broadcast scenario with two receivers, RLNC outperforms all the other transmission schemes. In particular, $\mathbf{GF}(2)$ performs better than LT codes and round robin. Moreover, as the erasure probability increases, the minimum field size under which RLNC outperforms ARQ becomes smaller.

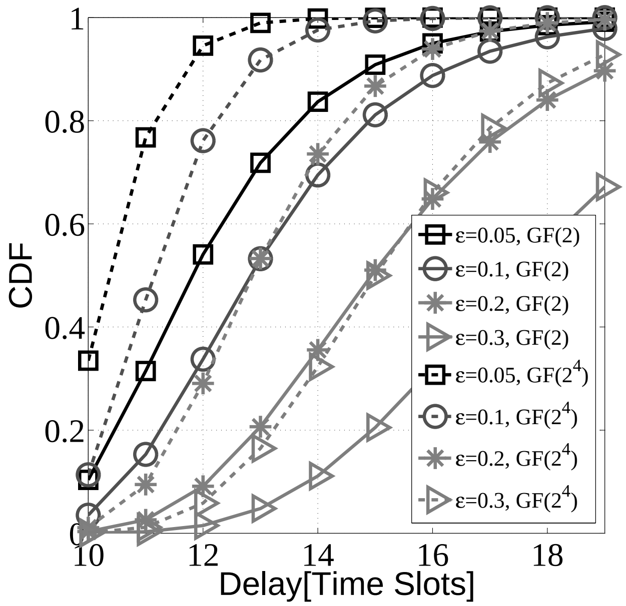
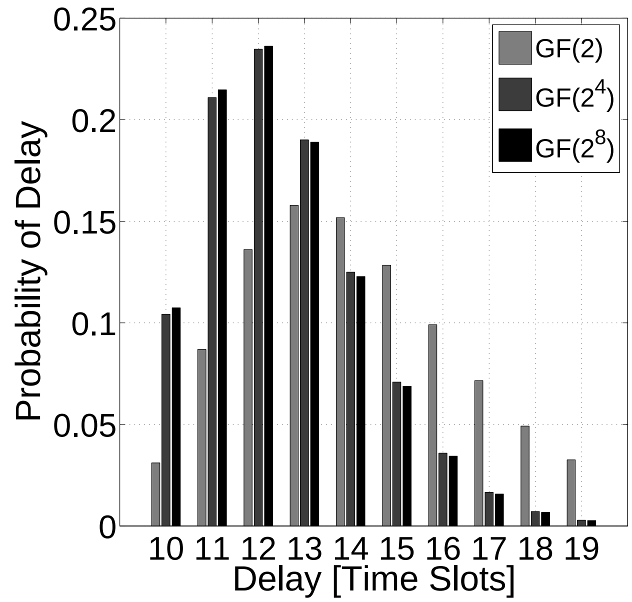


Fig. 5: The effect of erasure probability in the two-receiver case with $M = 10$, fixed field sizes and various erasure probabilities.

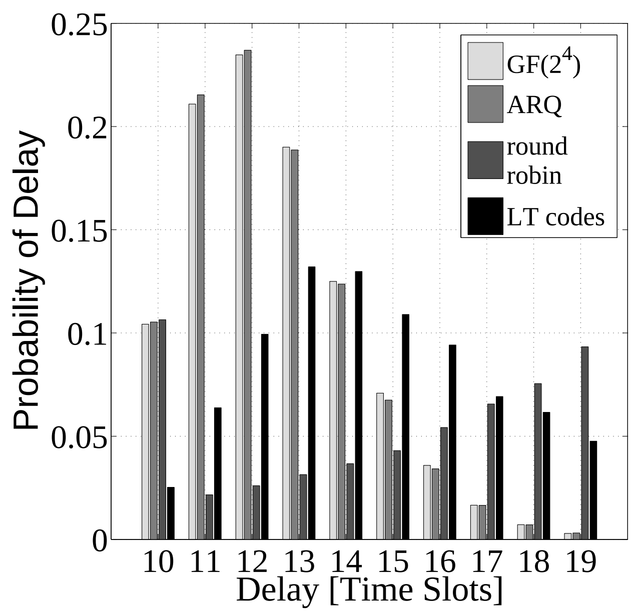
Erasure probability: The effect of erasure probability is illustrated in Figure 5. We fix the number of information symbols ($M = 10$) and the field size ($\mathbf{GF}(2)$ and $\mathbf{GF}(2^4)$), and vary the erasure probability ($\epsilon = 0.05$, $\epsilon = 0.1$, $\epsilon = 0.2$ and $\epsilon = 0.3$). The channels have identical erasure probabilities. We present only $\mathbf{GF}(2)$ and $\mathbf{GF}(2^4)$, because for $\mathbf{GF}(q \geq 2^4)$ the results are similar to $\mathbf{GF}(2^4)$. In this case, by increasing the erasure probability, the average delay increases with the same proportion for $\mathbf{GF}(2)$ and for $\mathbf{GF}(2^4)$. For instance, if we consider the case of $\mathbf{GF}(2)$, the average delay for $\epsilon = 0.05$ is 12.68 and for $\epsilon = 0.3$ we get 18.40, which implies an increase of 45%. The same analysis for $\mathbf{GF}(2^4)$ yields 10.96 and 15.77, respectively corresponding to an increase of 44%. Hence, for two receivers, the result shows that the effect of erasures and field size are completely separable.

B. One-receiver

In order to study the effect of field size in delay when using RLNC, we fix the number of information symbols, $M = 10$, and erasure probability, $\epsilon = 0.2$, while varying the field size. From our illustrations in Figure 6(a) we can see that $\mathbf{GF}(2)$ induces heavy tails. The average delay for $\mathbf{GF}(2)$ is 14.31 time-slots and for $\mathbf{GF}(2^4)$ and $\mathbf{GF}(2^8)$ is 13.5 time-slots. We compare other transmission schemes with $\mathbf{GF}(2^4)$, as shown in Figure 6(b). The results for ARQ and $\mathbf{GF}(2^4)$ are similar. Round robin scheduling is outperformed by $\mathbf{GF}(2)$. LT codes provide the worst performance for this scenario. Both the round robin mechanism and the LT codes induce heavier tails than those observed for small field sizes. RLNC over $\mathbf{GF}(2)$ thus offers superior worst-case delay guarantees than the two competing schemes.



(a) RLNC with various field sizes.



(b) $\mathbf{GF}(2^4)$, ARQ, round robin and LT codes

Fig. 6: The effect of the field size in the one-receiver case with $M = 10$, $\epsilon = 0.2$.

C. Number of information symbols

We now analyze the effect of the number of information symbols to be encoded for various field sizes. The probability that a receiver obtains M linearly independent combinations from M received coded symbols is given by: $P = \prod_{j=0}^{M-1}(1 - q^{j-M})$ [11], where q is the field size. By changing variables and calling $u = j - M$ we have: $P = \prod_{u=-M}^{-1}(1 - q^u)$. Note that for a given field size q, if u is below a certain value, q^u is negligible when compared to 1. Hence, when we increase the number of information symbols, i.e. the value of M, the probability of having M linearly independent combinations from M coded information symbols remains the same. For $\mathbf{GF}(q \geq 2^2)$, we may safely neglect the effect of the number of information symbols in the probability of obtaining M linearly independent combinations from M coded symbols. For the decoding delay in noisy channels, the erasure probability plays the central role on the choice of a suitable M in order to ensure, for example, that all M information symbols are delivered on time when the application has strict deadlines.

Algorithm 1 Number of Time Slots for an Experiment

Data: erasure probability (ϵ), number of information symbols (M), number of receivers (N)
Result: number of time slots (t)
$Sum \leftarrow 0$
$TotalSum \leftarrow 0$
$t \leftarrow M - 1$
while $TotalSum \leq 0.99$ **do**
  $t \leftarrow t + 1$
  $Sum \leftarrow Sum + \binom{t-1}{M-1} \cdot (1-\epsilon)^M \cdot \epsilon^{t-M}$
  $TotalSum \leftarrow Sum^N$
end while

V. Brute-Force Analysis

The next natural step would be to extend the delay analysis of Section III to the case of $N > 2$ receivers. Given the mathematical difficulty of characterizing the required transition probabilities and hence obtain the delay distribution of network coding for more than two receivers, we propose an alternative brute-force approach.

A. Methodology

We start by fixing the number of receivers N, the number of information symbols to be combined M, the field size q and the erasure probability ϵ of each channel. Based on the values of these parameters, erasure patterns are generated for testing purposes. Their length t must be sufficient to allow for all receivers to recover all of the information symbols with high probability. It is useful to represent each erasure pattern as a matrix of the form $\underline{ep} = [\underline{ep}_1 | \underline{ep}_2 | \ldots | \underline{ep}_N]^T$, where $\underline{ep}_i$ is a binary vector of length t. Each element of the vector indicates if a coded symbol transmitted at the corresponding time slot is erased or arrives correctly at receiver i.

In a second step, we generate all possible linear combinations of the M information symbols that can be generated by the encoder for t time slots. For each erasure pattern, we try out all possible encodings, which are erased at the given positions and decoded by Gaussian Elimination. The delay is measured by counting the number of slots until all information symbols are decoded by all receivers. To reduce the computational effort, we adapt the number of time slots to the parameter values of each experiment, such that at least M coded symbols are received with probability larger than 99%. The minimum number of time slots required is computed using Algorithm 1.

For each erasure probability we run a number of experiments (one for each erasure pattern) and compute the mean and standard deviation of the decoding delay. The length of the erasure pattern is determined using Algorithm 1. To ensure the statistical significance of our results, we must run a sufficient number of experiments depending on the value of the erasure probability. Let H be the number of erasures in a specific erasure pattern. We start by setting $H = 0$ and compute the probability of occurrence for this erasure pattern. The procedure is repeated by incrementing H at each step until the total probability of the thus obtained erasure pattern is higher than 0.9. This can be validated using the binomial distribution to compute the total probability of the erasure patterns with up to H erasures, specifically $P_o = \sum_{h=0}^{H} \binom{N \cdot t}{N \cdot t - h} \cdot (1-\epsilon)^{N \cdot t - h} \cdot \epsilon^h$. The value of H for which this probability exceeds 0.9 is denoted as H_{max}. It follows that the adequate number of erasure patterns for experimentation is given by $\Lambda = \sum_{h=0}^{H_{max}} \binom{N \cdot t}{N \cdot t - h}$.

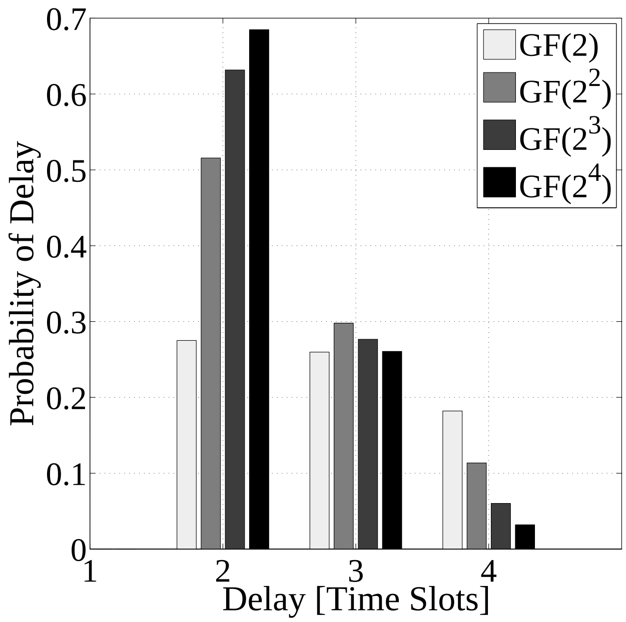
B. Experiments

The proposed approach is obviously demanding from a computational point of view. In our experiments, we used a cluster of computers to obtain results for field sizes up to $\mathbf{GF}(2^4)$ and a limited number of receivers. The cluster distributes the computing effort among several independent cores. Since the number of encodings is very large, we assign independent threads for the same experiment. The number of information symbols to be coded was kept deliberately small in our examples ($M = 2$), however larger values of M are possible depending on the computational power of the cluster and the available time. Recall that the delay is measured in each experiment for every set of encodings and every receiver. The delay histogram is then obtained by counting the number of coded symbols that yields a particular delay value. Coded symbols under which the receivers are unable to decode all the information symbols are also accounted for. The probability of decoding failure is given by

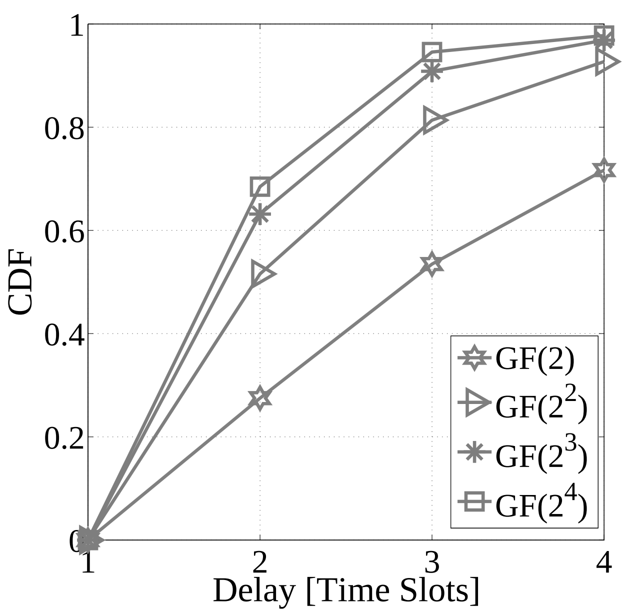
$$\mathcal{P}(\overline{D}) = \frac{1}{P_o} \sum_{\lambda=1}^{\Lambda} \mathcal{P}(\overline{D} | \Lambda_\lambda) \cdot \mathcal{P}(\Lambda_\lambda), \tag{13}$$

with $\mathcal{P}(\Lambda_\lambda) = (1-\epsilon)^{N \cdot t - h} \cdot \epsilon^h$, $\forall h = 0, 1 \ldots H$ and $P(\overline{D} | \Lambda_\lambda) = (1 - \frac{G_\lambda}{G})$, where G_λ is the number of encodings that guarantees successful decoding under the erasure pattern Λ_λ and $G = \max_\lambda G_\lambda$.

Comparing with other existing methods, the brute-force is the most well suited for our metric of interest. Whereas for a given set of input parameters the brute-force methodology generates all possible linear combinations of information symbols, the Monte Carlo method chooses a smaller number of linear combinations at random. The accuracy of analysis for Monte Carlo simulation depends on the choice of the random function we choose to produce the linear combinations and



(a) Probability of Delay



(b) CDF

Fig. 7: Brute-Force Analysis for $N = 3$, $M = 2$, $\epsilon = 0.05$.

on the repetition of experiments. Consequently, even if it requires more resources, the brute-force method should be addressed for a more precise analysis. As an option for heavy computational situations, Monte Carlo method may be used instead.

As an example, Figure 7 shows the results for the case of $N = 3$ receivers, $M = 2$ and $\epsilon = 0.05$ and various field sizes $\mathbf{GF}(2)$ up to $\mathbf{GF}(2^4)$. The number of experiments used in the study is 79.

The results provided in Figure 7 were obtained for erasure patterns that represent 97.5% of all possible occurrences. Since we are trying all possible encodings, we obtain the cumulative distribution function (CDF) for the delay, shown in Figure 7(b). Both histogram and cumulative function reveal that once again $\mathbf{GF}(2)$ induces a heavy tail and the performance of $\mathbf{GF}(2^3)$ is close to that of $\mathbf{GF}(2^4)$. Using equation (13), we can compute the probability of decoding failure for all field sizes. It varies from 0.032 to 0.097.

The brute-force approach can be used also to validate the analytical method proposed in Section III. For the case of $N = 2$, $M = 2$, $\epsilon = 0.05$ and $\mathbf{GF}(2^4)$ we must to run 37 experiments. Figure 8 shows that the analytical method and the brute-force approach yield the same delay distribution.

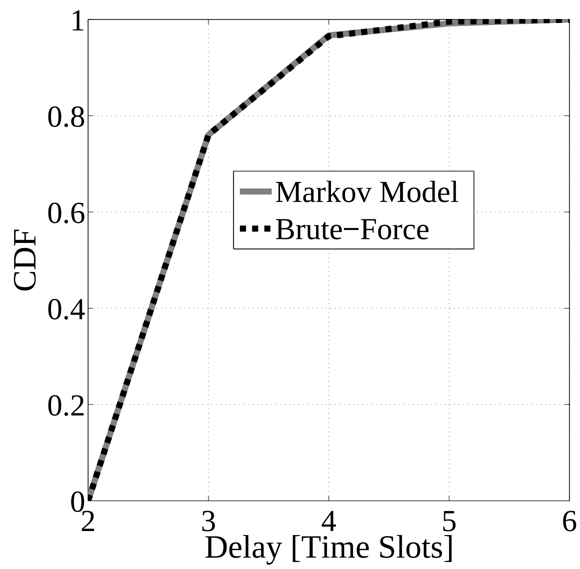


Fig. 8: Brute-Force vs. Analytic Model for $N = 2$, $M = 2$, $\epsilon = 0.05$, $\mathbf{GF}(2^4)$.

VI. Conclusions

We considered the delay behavior of network coding which is key to the design of any system subject to strict message deadlines. By determining the delay distribution of RLNC for the case of two receivers, we were able to identify which parameter settings can meet specific worst-case guarantees. The benefits of RLNC in the broadcast scenario of interest were further highlighted by comparing its delay distribution to that of three other transmission schemes. Perhaps the most important insights are that network coding for two receivers over $\mathbf{GF}(2^4)$ outperforms ARQ and that $\mathbf{GF}(2)$, although simple to implement, induces a heavy tail in the delay distribution.

Our analysis was given for the case of one-hop communication, but the insights we obtained can be translated to more complex networks. In fact, the sender can be either the original source at the edge of the network or an intermediate node somewhere in the core. Likewise, the receivers can be either intermediate nodes in the network or the final destination at the edge of the network. On the other hand, the information symbols as defined in our problem statement could in fact be coded symbols if the sender acts as an intermediate node.

What does not change is the fact that computing the delay distribution of network coding is a high-dimensional and computationally demanding problem. Since we are interested in guidelines for system design, the proposed methods based on a Markov chain model and a brute-force approach could be used in combination to seek meaningful heuristics and bounds for the design. In particular, a hybrid search would be able to select the most relevant erasure patterns while capitalizing on the Markov chain to take the impact of the field size into account. Devising such strategies is part of our ongoing work.

APPENDIX

The proof of Lemma 1 is given in the following.

Proof: The number of valid states is the solution of a difference equation when $N = 2$ and $M \geq 2$, and we denote it by $A(M, 2)$. By considering the number of valid states for $M = 2, 3, 4, 5, 6$ information symbols, we noticed that they follow a difference equation of the form $A(M+5, 2) = A(M+4, 2) - 4A(M+3, 2) + 6A(M+2, 2) - 4A(M+1, 2) + A(M, 2)$. The characteristic polynomial for this difference equation is $z^4 - 4z^3 + 6z^2 - 4z + 1 = 0$, which has only one root, $z = 1$, of multiplicity 4. Therefore, the linear recurrence equation with constant coefficients takes the form

$$A(\theta, 2) = c_1 + c_2\theta + c_3\theta^2 + c_4\theta^3, \tag{14}$$

with $\theta \in 0...3$. From (14) we find $c_1 = 10$, $c_2 = \frac{47}{6}$, $c_3 = 2$ and $c_4 = \frac{1}{6}$ after applying the initial conditions, e.g., the number of valid states for $M = 2, 3, 4, 5, 6$. By substituting the coefficients and $\theta = M - 2$ in expression (14) we get $A(M, 2) = 10 + \frac{47}{6}(M-2) + 2(M-2)^2 + \frac{1}{6}(M-2)^3$, which is the same with (2).

This concludes the proof. ■

REFERENCES

[1] M. Nistor, J. Barros, F. Vieira, T. T. V. Vinhoza, and J. Widmer, “Network coding delay: A brute-force methodology,” in *Proc. Information Theory and Applications Workshop (ITA 2010)*, San Diego, CA, February 2010.

[2] M. Nistor, R. A. Costa, T. T. V. Vinhoza, and J. Barros, “Non-asymptotic analysis of network coding delay,” in *Proc. IEEE International Symposium on Network Coding (NetCod 2010)*, Toronto, Canada, June 2010.

[3] R. Ahlswede, N. Cai, S.-Y.R. Li, and R.W. Yeung, “Network information flow,” *IEEE Trans. Inform. Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

[4] S. Katti, H. Rahul, W. Huss, D. Katabi, M.Médard, and J.Crowcroft, “XORs in the air: Practical wireless network coding,” in *ACM Sigcomm*, Pisa, Italy, September 2006.

[5] C. Gkantsidis and P. Rodriguez, “Network coding for large scale content distribution,” in *Proc. IEEE Infocom*, Calcutta, India, December 2005.

[6] C. Fragouli, J.-Y. Le Boudec, and J. Widmer, “Network coding: An instant primer,” *ACM Computer Communication Review*, vol. 36, pp. 63–68, Jan. 2006.

[7] T. Ho, M. Médard, J. Shi, M. Effros, and D.R. Karger, “On randomized network coding,” *Proc. of 41st Annual Allerton Conference on Communication, Control, and Computing*, vol. 1, pp. 1–10, October 2003.

[8] M. Luby, “LT codes,” in *Proc. 43rd Ann. Symp. on Fundations of Computer Science*, Vancouver, Canada, November 2002.

[9] R. Koetter and M. Médard, “An algebraic approach to network coding,” *IEEE/ACM Trans. Networking*, vol. 11, no. 5, pp. 782–795, October 2003.

[10] P. A. Chou, T. Wu, and K. Jain, “Practical network coding,” in *Proc. 41st Allerton Conf. Communication, Control and Computing*, Monticello, IL, USA, October 2003.

[11] A. Eryilmaz, A. Ozdaglar, and M. Médard, “On delay performance gains from network coding,” *Proc. 40th annual conference on information sciences and systems, CISS 2006*, pp. 864–870, March 2006.

[12] D. E. Lucani, M. Médard, and M. Stojanovic, “On coding for delay - new approaches based on network coding in networks with large latency,” *Proc. Information Theory and Applications Workshop (ITA 2009)*, February 2009.

[13] D. E. Lucani, M. Stojanovic, and M. Médard, “Random linear network coding for time division duplexing: Field size considerations,” in *Proc. GLOBECOM*, Hawaii, USA, December 2009.

[14] M. Pedersen, J. Heide, F. H. P. Fitzeka, and T. Larsen, “Network coding for mobile devices - systematic binary random rateless codes,” in *Proc. ICC 09 - Workshop on Cooperative Mobile Networks*, Dresden, Germany, June 2009.

[15] T. K. Dikaliotis, A. G. Dimakis, T. Ho, and M. Effros, “On the delay of network coding over line networks,” in *Proc. IEEE ISIT*, Seoul, Korea, June-July 2009.

[16] D. E. Lucani, M. Stojanovic, and M. Médard, “Random linear network coding for time division duplexing: When to stop talking and start listening,” in *IEEE Infocom*, Rio de Janeiro, Brazil, April 2009.

[17] C. Fragouli, D. Lun, M. Médard, and P. Pakzad, “On feedback for network coding,” in *Proc. 41st annual conference on information sciences and systems, CISS 2007*, Baltimore, MD, USA, March 2007.

[18] L. Keller, E. Drinea, and C. Fragouli, “Online broadcasting with network coding,” in *4th Workshop on Network Coding, Theory, and Applications, NetCod 2008*, Hong Kong, China, January 2008.

[19] J. K. Sundararajan, D. Shah, and M. Médard, “ARQ for network coding,” in *IEEE ISIT 2008*, Toronto, Canada, July 2008.

[20] J. K. Sundararajan, D. Shah, and M. Médard, “Online network coding for optimal throughput and delay–the two-receiver case,” *ISITA*, vol. 1, pp. 1–4, December 2008.

[21] R.A. Costa, D. Munaretto, J. Widmer, and J. Barros, “Informed network coding for minimum decoding delay,” in *Proc. of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS 08)*, Atlanta, Georgia, September-October 2008.

[22] J. Barros, R. A. Costa, D. Munaretto, and J. Widmer, “Effective delay control in online network coding,” in *Proc. IEEE Infocom*, Rio de Janeiro, Brazil, April 2009.

[23] A. Mahmino, J. Lacan, and C. Fraboul, “Guaranteed packet delays with network coding,” in *SECON Workshops ’08*, San Francisco, CA, June 2008.

[24] E. Drinea, C. Fragouli, and L. Keller, “Delay with network coding and feedback,” in *ISIT*, Seoul, Korea, June-July 2009.

[25] A. Shokrollahi, “Raptor codes,” *IEEE/ACM Trans. Networking*, vol. 14, pp. 2551–2567, March 2006.

[26] D. J. C. Mackay, *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press, 2003.

[27] L. Tassiulas, R. Tassiulas, and A. Ephremides, “Dynamic server allocation to parallel queues with randomly varying connectivity,” *IEEE Trans. Inform. Theory*, vol. 39, no. 2, pp. 466–478, March 1993.