Gamma Codes: A Low-Overhead Linear-Complexity Network Coding Solution

Kaveh Mahdaviani, Masoud Ardakani, Hossein Bagheri, Chintha Tellambura
Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. T6G 2V4
Email: {mahdaviani,ardakani,tellambura}@ece.ualberta.ca, bagheri2@ualberta.ca

***Abstract*—We introduce a family of sparse random linear network codes with outer-code. Due to the bold role of the incomplete gamma function in their design, we call these codes *"Gamma codes"*. We show that Gamma codes outperform all the existing linear-complexity network coding solutions in terms of reception overhead, while keeping the encoding and decoding complexity linear in the block length.**

I. Introduction and Background

Network coding for multicast turned into a promising research field soon after the basic concept of network coding was introduced in [1]. The idea of linear network coding, where the transmitted packets are formed as linear combinations of information packets, was introduced in [2]. This idea was further extended using an algebraic approach to find the coefficients of the linear combination in [3]. Later, using random coefficients, it was shown that random linear network coding for multicast is sufficient to achieve zero reception overhead[1] with arbitrarily close to zero failure probability when the code alphabet q is large enough [4], [5].

In brief, encoding of random linear network codes is done by forming random linear combinations of data packets at the source and every other node of the network. For block length of K information packets, each containing d symbols, the complexity of encoding is therefore of $\mathcal{O}(Kd)$ operations in $GF(q)$ per coded packet. Each receiver receives enough linear combinations to form a full rank linear equation system, and performs the decoding by solving it. The complexity of decoding in this case is of order $\mathcal{O}(K^2 + Kd)$ per information packet, which is impractical for applications with moderate to large block length. We will refer to this coding scheme as dense random linear network coding.

One of the most important steps in reducing the complexity of network coding was the idea of fragmentation of data at the source to distinct generations [6]. Restricting the random linear combinations to be formed only within each generation, the final linear equation system will be sparse and locally solvable inside each generation. However, to resolve the problem of *rare blocks*, and *block reconciliation*, a significant number of control messages needed to be exchanged [7], [8]. Consequently, [9] proposed random scheduling for generations to avoid control traffic. Also known as sparse random linear network coding (SRLNC), this idea reduces the complexity of encoding per coded packet and the complexity of decoding per information packet respectively to $\mathcal{O}(gd)$, and $\mathcal{O}(g^2+gd)$, where g is the number of packets in each generation. For small d this complexity is feasible, making SRLNC an attractive approach in practice. On the other hand, the reception overhead in this scheme, affected by the *curse of coupon collector* phenomena, will not vanish even for very large number of information packets or alphabet size [10]. For sufficiently large number of generations the reception overhead will grow with K as $\mathcal{O}(\log(K))$, hence raising a trade-off between complexity and reception overhead.

In independent attempts to reduce the logarithmically growing reception overhead in the computationally efficient SRLNC, [11] and [12] proposed overlapped generations (i.e., some generations share common packets), so that generations could help each other to decode faster. In [10] a new *overlapped* SRLNC scheme, called the *Random Annex code* was introduced and shown to outperform all the previous schemes.

Another idea for avoiding the logarithmically growing overhead of SRLNC suggested in [9] is to use an outer-code. In fact, [9] shows that using an outer-code can reduce the overhead to a constant, independent of K. However, in [9], the outer-code is considered as a separate block, which will come to participate in the decoding after the recovery of a $1-\delta$ fraction of the generations, for some small predefined δ. This outer-code is then responsible for the recovery of the remaining δ fraction of the generations. This design also ignores the received packets pertaining to the remaining δ fraction of generations.

It is easy to show that receiving enough packets to recover a $1-\delta$ fraction of the generations for some small δ, without the help of outer-code raises a significant probability of receiving more than g packets for some generations. This results in linearly dependent packets and hence a significant reception overhead which will not vanish even as the block length tends to infinity.

A. Main Idea

Considering overlapped SRLNC, it is clear that the overlap between different generations can be viewed as a simple repetition outer-code. Not limiting ourselves to such a simple outer-code, in this work we study the design of SRLNC with outer-code in a more general way.

The idea of using an outer-code for SRLNC has been introduced previously in [9]. However, the design we propose

[1] In this work we define the reception overhead as the number of received packets required for successful decoding divided by the number of information packets, minus one.

978-1-4673-1892-1/12/$31.00 © 2012 IEEE

in this work has major differences compared to [9] as follows; i) Instead of waiting for the SRLNC to recover a large fraction of generations, in our design the outer-code comes to play as soon as the first generation is recovered. This results in a joint decoding scheme instead of the separate decoding used in [9]. ii) Unlike [9], the outer-code in a Gamma code does not ignore the packets received in non-full rank generations. Indeed, instead of using classic erasure correcting outer-codes such as right regular LDPC codes [13], [14] used in [9], we use a fixed-rate version of the Raptor code [15]. Our selection is motivated by Raptor code's natural capability to start participation in recovering some erased bits even when the fraction of known bits are much smaller than the code rate.

Furthermore, it is shown that the optimal rate for the outer-code in Gamma codes is significantly lower than that of the design proposed in [9], yet it results in a much smaller reception overhead.

In Section IV we compare the performance of our proposed design with the Random Annex code [10], as the best known SRLNC with overlapping generations, as well as the SRLNC with outer-code design of [9]. This comparison shows that Gamma codes significantly reduce the reception overhead compared to all the existing linear-complexity designs.

The rest of this paper is organized as follows. The next section is devoted to describing the encoding and decoding structure of the proposed code. We also discuss the network model in the same section. In Section III, we provide a discussion on the selection of parameters of the outer-code used in the structure of Gamma codes. Finite length numerical comparisons with the existing counterparts are provided in Section IV, and finally Section V concludes the results.

II. Network Model and Coding Scheme

A. Network Model

In this work, following the convention in [10], [11], we consider the transmission of a file from a source to a destination over a unicast link. Due to packet loss, random processing times in the intermediate nodes, and also diverse routings in a dynamic network structure, the link is supposed to introduce an unknown erasure rate and a variable delay. As a result, using random linear network coding at the intermediate nodes, a receiver would receive a random subset of linear combinations of the transmitted packets. A specific coding scheme can be used to set constraints on the combinations such as limiting the combinations to be formed inside each generation [6], or to establish more dependencies among the received packets using either outer-codes [9], or overlapping generations [10]–[12].

B. Encoding

The process of encoding at the source in Gamma codes consists of two steps. For a K-packet long file, the first step is to use a linear outer-code of rate R to encode the message into N *"outer-coded"* packets. The source then partitions the N outer-coded packets into $n = \lceil \frac{N}{g} \rceil$ generations, where $\lceil x \rceil$ is the smallest integer larger or equal to x. For convenience, we assume that N is a multiple of g.

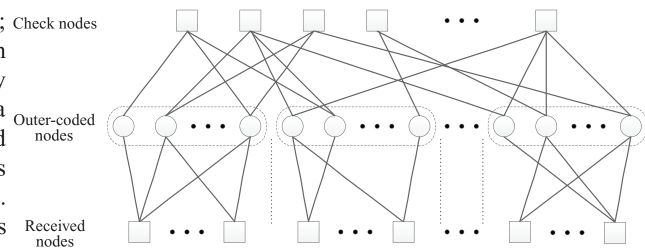

Fig. 1. The decoding graph for a Gamma code with check nodes, outer-coded nodes, and received nodes corresponding to outer-code's check equations, outer-coded packets, and received packets respectively. Each group of outer-coded nodes separated in the figure by a dashed box represents a generation.

The second step consists of iteratively forming output packets to be transmitted through the network. For each output packet, at the source we first select a generation $j \in \{1, \cdots, n\}$, uniformly at random with replacement. We also select an element $\alpha \in (GF(q))^g$ uniformly at random. Consequently we form the output packet as the linear combination of the g outer-coded packets of the selected generation j with coefficient vector α. Along with each output packet, the index of the selected generation and the coefficient vector α are also transmitted with the packet.

Coding at the intermediate nodes follows the convention of SRLNC as in [9]–[11]. Thus, the complexity of encoding per output packet at the source, and at any intermediate node is $\mathcal{O}(gd + \bar{d}(1-R)/R)$, and $\mathcal{O}(gd)$, respectively, where $\bar{d}$ is the average degree of outer-code's check nodes. This constant complexity per output packet translates to an overall linear complexity of encoding in terms of the block length K.

C. Decoding

The decoding process starts as soon as the receiver receives enough packets from one of the generations to form a full rank linear equation system for that generation. We refer to such a generation as a *"full rank generation"*. This generation is then decoded by Gaussian elimination. At this point, an iterative decoding scheme starts which operates on the decoding graph of the code shown in Fig. 1.

Each iteration in this process consists of two steps. At the first step each node corresponding to a recently recovered outer-coded packet from the recently recovered generation(s) will be removed from the decoding graph. Removing these nodes reduces the degree of the outer-code check nodes. Any check node reduced to degree one recovers a new outer-coded packet. This step is equivalent to one iteration of the *edge deletion decoding* [16] on the outer-code.

In the second step, the newly recovered outer-coded packets of step one are removed from the linear equation systems of non-recovered generations. Since the coefficient vectors of the output equations are dense, the reduced equation systems will preserve their rank with high probability, especially for large enough alphabet size q. Since the unknowns are reduced and the rank is preserved, there is a possibility that new generations can be recovered through Gaussian elimination.

These two steps will be repeated until either all input packets are recovered, or the decoder is stuck and no new packet could be recovered. In the latter case, we receive more packets from the network to be able to resume the decoding. The complexity of the decoding per information packet is hence $\mathcal{O}(g^2+gd+\bar{d}(1-R)/R)$. Again, this constant complexity per information packet translates to an overall decoding complexity linear in terms of the block length K.

III. Outer-Code Details

As mentioned before, the idea of using an outer-code to reduce the overhead of SRLNC was first suggested in [9]. However, the design proposed in [9] is based on recovering a $1-\delta$ fraction of generations without the help of the outer-code where a good choice for δ is usually very small as shown in our simulation results. Moreover, this design refuses to use the information transmission capacity of received packets pertaining to the non-full rank generations in the remaining δ fraction. The selection of δ is based on the desired overhead where a smaller desired overhead requires a smaller δ [9]. In addition, the selection of the outer-code in [9] is based on the assumption that the outer-code will come to play its role after the recovery of $1-\delta$ fraction of generations. As a result the conventional erasure correcting codes such as right-regular LDPC codes [13], [14] are selected. These codes are very good for accomplishing the decoding when $1-\delta$ fraction of the block is recovered, but they have a very poor contribution before that point. Consequently, joint decoding of the outer-code and SRLNC is not likely to improve the performance of the solution proposed in [9]. This fact is further investigated in our simulation results.

Obviously recovering a close-to-one fraction of generations without the help of the outer-code requires receiving at least g packets in each of $1-\delta$ fraction of generations. As packets are assumed to be received independently and uniformly at random, this causes a significant number of generations receiving much more than g packets. Since any generation has at most g independent packets, then we will have many linearly dependent received packets, to which we refer as *"excess packets"*. This results in a significant overhead. Indeed we remark without proof that in order to recover a $1-\delta$ fraction of generations with SRLNC without the help of the outer-code, the overhead will converge to

$$\frac{(g-1)!-\Gamma_g^{-1}(\delta)}{g!\delta}-1$$

as the number of generations n tends to infinity. In the above, $\Gamma_a^{-1}(y)$ denotes the inverse of the incomplete Gamma function $\Gamma(a,x)$ with respect to x, where

$$\Gamma(a,x)=(a-1)!\sum_{i=0}^{a-1}\frac{e^{-x}x^i}{i!}$$

for any integer $a>0$, and real number x.

In this section we propose a different approach to the outer-code design. The main goal in this approach is to keep the number of excess received packets as low as possible, and simultaneously, use almost all the non-excess received packets in non-full rank generations as opposed to ignoring them in the decoding.

A. Outer-Code Rate Selection

According to the above discussion, to avoid the reception overhead it is crucial to keep the probability of receiving linearly dependent packets close enough to zero. Moreover selecting the code alphabet size q large enough, it can be shown that any randomly selected set of m received packets pertaining to a single generation will be linearly independent with probability arbitrarily close to one as long as $m \leq g$ [10]. The following discussion provides the background needed for selecting the appropriate rate for the outer-code. Assuming that we have stopped the packet reception process at some arbitrary time, we refer to the total number of received packets divided by the number of generations n, as the *"normalized"* number of received packets.

We refer to the average normalized number of received coded packets required for having the first full rank generation with high probability, as the latency of the first full rank generation and denote it by L_1.

Theorem 1: The latency of the first full rank generation is

$$L_1=\Gamma_g^{-1}((g-1)!(1-\frac{1}{n})) \tag{1}$$

For the proof please refer to the appendix.

Moreover, it can be shown that the variance of this latency tends to zero as $n\to\infty$ [17], [18]. For large enough q, it is obvious that before receiving enough packets to form the first full rank generation we will have no excess packets. However, as soon as a fraction $x_0>0$ of the generations are full rank, any new received packet will be an excess packet with probability x_0. As a result of Theorem 1, we have the following corollary.

Corollary 1: For large enough number of generations n, the maximum number of packets that the receiver can receive to keep the number of excess packets arbitrarily close to zero is

$$M=n\Gamma_g^{-1}((g-1)!(1-\frac{1}{n})). \tag{2}$$

Moreover, we can always choose a large enough alphabet size q in order to guarantee the linear independence of all of the received packets with high probability.

As a result, to keep the average number of excess packets at the minimum, and gain the highest transmission rate at the same time, the receiver needs stop receiving packets after receiving M packets defined by (2). Hence, the best selection for the outer-code rate is given by

$$R=\frac{\Gamma_g^{-1}((g-1)!(1-\frac{1}{n}))}{g}. \tag{3}$$

B. Outer-Code Generating Polynomial

According to the structure of our proposed decoder described in section II-C, the decoder is not able to decode any packet before receiving enough packets to form the first full rank generation. Hence, setting the outer-code rate based on (3), in order to keep the number of excess packets at the minimum, we need an appropriate outer-code to be able to accomplish the decoding when the normalized number of received packets is not much larger than

$$\max\left\{\Gamma_g^{-1}((g-1)!(1-\frac{1}{n})), r_1\right\},$$

where, r_1 denotes the normalized number of received packets required for forming the first full rank generation. Obviously r_1 is a random variable with mean $\Gamma^{-1}(g,(g-1)!(1-\frac{1}{n}))$, according to Theorem 1. However, the average number of recovered packets when the packet reception stops and the outer-code comes to play its role in the decoding process is concentrated around g as described before. Therefore, conventional fixed rate linear erasure correcting codes such as right regular LDPC codes [13], [14] with rate R are not appropriate choices for our outer-code, since they are known to have poor recovery before having an R fraction of the codeword recovered. However, if the outer-code is capable to recover some more packets at this point, using the linear equations corresponding to the received packets from the non-full rank generations, the decoding process will recover more packets iteratively.

Here, before describing the structure of the proposed outer-code for Gamma codes, we briefly review the problems in the structure of the outer-codes already used in the similar solutions. This brief discussion also defines some of the properties of the appropriate outer-code's structure.

The main problem with right regular LDPC outer-codes is indeed in their right regular nature. The degree of all the check equations in these codes is typically large (six or more) for moderate to high rates. In this scenario, at the beginning of the decoding process when the fraction of known packets is very small it is impossible for the decoder to reduce the outer-code check equations to degree-one and use them to recover new packets. This makes the decoding stuck at the beginning.

A naive solution is to force all the check nodes to be of degree two to maximize the participation of the outer code. This is similar to SRLNC with overlap, where all the check equations of the equivalent repetition outer-code are of degree two. Although this provides a good contribution in the decoding process at the beginning, these outer-codes fail to be as useful when eventually the fraction of recovered packets will grow. The reason for this fact in brief is that the low average degree of check nodes translate into low connectivity of the decoding graph. Hence, starting the decoding does not guarantee the spreading of recovery process to all the segments of the graph, and with high probability some segments will remain unrecoverable. Therefore, successful decoding again needs enough overhead to ensure the reception of some packets among the neighbours of almost every check node. This anyway causes a significant overhead due to the latency of receiving some rare packets.

The design of LT codes [16], and Raptor codes [15] however is based on a carefully selected distribution for the degree of output nodes (equivalent to the check equations), which guarantees a good coverage all over the codeword and simultaneously provides a high probability of reduction to degree one at any intermediate stage during the decoding. Although these codes are originally designed for rateless coding in the binary erasure channels, but the desirable properties inherent in their design can be used for the design of outer-codes in our proposed setting. Inspired by the above discussion we will use a low density generator matrix (LDGM) code [19], with check degree distribution similar to the output degree distribution of Raptor codes, as the outer-code in the Gamma codes.

Raptor codes are essentially a concatenation of a high rate linear erasure correcting code such as right regular LDPC codes, and an LT code with a truncated output degree distribution, where this concatenation allows for keeping the complexity linear along with the desired properties of rateless codes mentioned above. Thus, we also use a concatenation of a high rate right regular LDPC code of rate R_1, and an LDGM code of rate R_2, with a check degree distribution based on a truncated *Soliton* distribution. This concatenation results in an outer-code with rate $R = R_1R_2$ which inherits the behaviour of Raptor codes in providing high probability of reducing check equations to degree one all through the decoding process, and accomplishing the decoding with linear complexity. The LDGM check degree distribution can be characterized by a generating polynomial $P(x)$. More specifically, $P(x) = \sum_{i=2}^{D} p_i x^i$ where p_i is the probability that a randomly selected check equation covers i outer-coded packets. Obviously the minimum degree of $P(x)$ is two, since any check equation should encounter at least two outer-coded packets, and $\sum_{i=2}^{D} p_i = 1$. Moreover, packets contributing in each check equation are considered to be distributed uniformly at random among all the outer-coded packets.

The selected generating polynomial for the LDGM part in Gamma codes is then based on a selected rate R_2 as follows,

$$P(x) = \sum_{i=2}^{D^*} \frac{1}{i(i-1)} x^i + \frac{1}{D^*} x^{(D^*+1)}, \tag{4}$$

where the rate of this code is selected to be

$$R_2 = \frac{\Gamma_g^{-1}((g-1)!(1-\frac{1}{n}))}{gR_1}, \tag{5}$$

and D^* is chosen according to the following,

$$D^* = \left\lceil \frac{1}{1-R_2} \right\rceil. \tag{6}$$

This generating polynomial was originally introduced in [20], and has already been used in Raptor code design [21]. In the next section we provide simulation results for a practical setting to compare the performance of the proposed Gamma codes and other existing solutions.

IV. Example Codes and Finite Length Evaluation

In this section we compare the performance of the proposed Gamma codes with two best existing linear-complexity network codes, namely the Random Annex code which outperforms all the other existing SRLNC with overlapping generation [10], and the SRLNC with LDPC outer-codes proposed in [9]. In order to keep the comparison closer to a practical setting, we use bounded block lengths and binary alphabet for simulations. It is remarked that large alphabet size and block

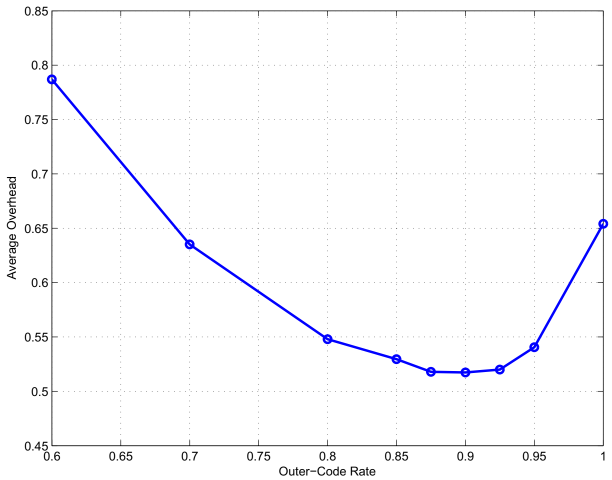

Fig. 2. The average overhead as a function of the outer-code rate in SRLNC with LDPC with $n = 67,\ g = 25,\ q = 2$.

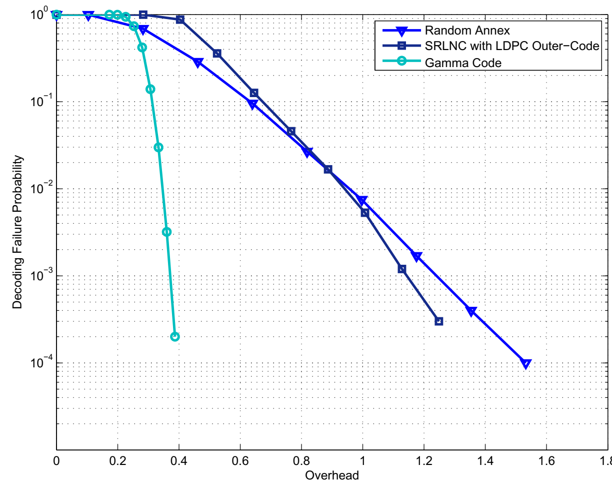

Fig. 3. The decoding failure probability as a function of the reception overhead for three different linear complexity network codes with $n = 67,\ g = 25,\ q = 2$.

length improve the performance in all of these codes, but our comparison trend will not be affected in general. The main measure of performance as widely accepted in the literature is the average overhead required for successful decoding. As mentioned earlier, the average overhead is defined as $\bar{O} = \mathbb{E}\{(N_r - K)/K\}$, where N_r and K denote the total number of received packets required for successful decoding, and the total number of information packets, respectively.

We set the number of generations, and the generation size to be $n = 67,\ g = 25$, respectively. Therefore, we will have a total encoded block length $N = 1675$. For this setting, it is shown in [10] that the optimal annex size is 10, and hence the optimal number of information packets to be transmitted is $K = 1000$, leading to the outer-code rate $R = 0.5970$.

For the SRLNC with LDPC outer-code, we have performed a search to find the optimal outer-code rate. The result shown in Fig. 2 reveals that the optimal outer-code rate for this block length and generation size is $R = 0.9$, which corresponds to $K = 1508$.

In the case of Gamma code, from (3) we have the outer-code rate $R = 0.6161$, which corresponds to $K = 1032$, however to find the best combination of two components of the outer-code rate, $R_1,\ R_2$, we have performed another search. The results presented in table I shows that while the difference in the average overhead is not vary significant, the best selection for R_1 (for the high rate LDPC code) is 0.9701.

Setting $R_1 = 0.9701$, values of $P(x),\ R_2,\ D^*$ will be derived from (4) to (6). Fig. 3 represents the probability of failure in decoding as a function of the reception overhead. Simulations are done for 10000 runs for each code. As can be inferred easily from the figure, Gamma code outperforms the other linear complexity network codes. The average overhead required for successful decoding of the three codes under study are provided in table II. This signifies the importance of a careful outer-code design for SRLNC.

V. Conclusion

In this work, we introduced a new family of linear-complexity network codes based on the idea of sparse random linear network coding with outer-code. Key to our design was a joint decoder for which we devised a proper combination of an outer-code and a basic SRLNC. It was shown through simulations that the proposed codes outperform the best existing linear-complexity network coding solutions both in terms of the average overhead and the probability of decoding failure.

Appendix
Proof of Theorem 1

Let B_r be a random variable equal to the number of received coded packets pertaining to a randomly selected generation, when the normalized number of received coded packets is r. For any randomly selected received coded packet we assume the probability that it belongs to a certain generation has uniform distribution on the set of all the generations. Hence, it is obvious that B_r has a binomial probability distribution as

$$P(B_r = i) = \binom{rn}{i}\left(\frac{1}{n}\right)^i\left(\frac{n-1}{n}\right)^{rn-i},\ i = 0, 1, \ldots, rn.$$

TABLE I
Average overhead for Gamma codes,
$n = 67,\ g = 25,\ R = 0.6161,\ q = 2$.

R_1	$\frac{66}{67}$	$\frac{65}{67}$	$\frac{64}{67}$	$\frac{63}{67}$	$\frac{62}{67}$
$\bar{O}$	0.2764	0.2737	0.2884	0.2899	0.2878

TABLE II
Average overhead for different linear complexity network codes, $n = 67,\ g = 25,\ q = 2$.

Code	SRLNC with LDPC	Random Annex code	Gamma code
$\bar{O}$	0.5164	0.3944	0.2737

It is easy to see that

$$P(B_r = i) \simeq \frac{e^{-r} r^i}{i!},$$

where the error in the approximation vanishes as n goes to infinity. Indeed the approximation is very tight even for values of n as small as a few tens, and hence by a slight abuse of notations we will use this approximation instead of the exact value for finite number of generations n. However, this notation is exact for the asymptotic case. In addition, take $A_{r,i},\ 0 \le r,\ i = 1, \cdots, n$, as the event that we have received at least g coded packets from the i^{th} generation when the normalized number of received coded packets is r. Therefore the following expression describes the latency of the first full rank generation.

$$L_1 = \inf \left\{ r \mid \mathbb{E} \left\{ \sum_{i=1}^{n} I_{A_{r,i}}(\omega) \right\} \ge 1 \right\}$$

where I_A is the indicator function of the event A, i.e.

$$I_A(\omega) = \begin{cases} 1, & \text{if } \omega \in A \\ 0, & \text{if } \omega \notin A \end{cases},$$

Now, using the linearity of the expected value we have

$$\begin{aligned} L_1 &= \inf \left\{ r \mid \sum_{i=1}^{n} \mathbb{E}\{I_{A_{r,i}}(\omega)\} \ge 1 \right\} \\ &= \inf \left\{ r \mid \Pr\left[B_r \ge g\right] \ge \frac{1}{n} \right\} \\ &= \inf \left\{ r \mid \Pr\left[B_r \le g-1\right] \le 1 - \frac{1}{n} \right\} \\ &= \inf \left\{ r \mid \sum_{i=0}^{g-1} \frac{e^r r^i}{i!} \le 1 - \frac{1}{n} \right\} \\ &= \Gamma_g^{-1}((g-1)!(1 - \frac{1}{n})) \end{aligned}$$

■

References

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

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

[3] R. Koetter and M. Medard, "An algebraic approach to network coding," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 782–795, Oct. 2003.

[4] T. Ho, R. Koetter, M. Medard, D. R. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *Proc. IEEE International Symposium on Information Theory (ISIT '03),*, Yokohama, Japan, Jun./Jul. 2003, p. 442.

[5] T. Ho, M. Medard, R. Koetter, M. Effros, J. Shi, and B. leong, "A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.

[6] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. 41st Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, USA, Oct. 2003, pp. 40–49.

[7] A. R. Bharambe, C. Herly, and V. N. Padmanabhan, "Analyzing and improving a BitTorrent network's performance mechanisms," in *Proc. the 25th IEEE International Conference on Computer Communications (INFOCOM '06),*, Barcelona, Spain, Apr. 2006, pp. 1–12.

[8] J. Xu, J. Zhao, X. Wang, and X. Xue, "Swifter: Chunked network coding for peer-to-peer content distribution," in *Proc. IEEE International Conference on Communication (ICC '08),*, Beijing, China, May 2008, pp. 5603–5608.

[9] P. Maymounkov, N. Harvey, and D. S. Lun, "Methods for efficient network coding," in *Proc. 44th Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, USA, Sep. 2006, pp. 482–491.

[10] Y. Li, E. Soljanin, and P. Spasojević, "Effect of the generation size and overlap on throughput and complexity in randomized linear network coding," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 1111–1123, Feb. 2011.

[11] D. Silva, W. Zeng, and F. Kschischang, "Sparse network coding with overlapping classes," in *Proc. Workshop on Network Coding, Theory, and Applications (NetCod '09)*, Lausanne, Switzeland, Jun. 2009, pp. 74–79.

[12] A. Heidarzadeh and A. H. Banihashemi, "Overlapped chuncked network coding," in *Proc. IEEE Information Theory Workshop (ITW '10),*, Cairo, Egypt, Jan. 2010, pp. 1–5.

[13] P. Oswald and A. Shokrollahi, "Capacity-achieving sequences for erasure channel," *IEEE Trans. Inf. Theory*, vol. 48, no. 12, pp. 3017–3028, Dec. 2002.

[14] M. A. Shokrollahi, "New sequences of linear time erasure codes approaching the channel capacity," in *Proc. the 13th International Symposium on Applied Algebra, Algebraic Algorithms and Error-Correcting Codes (AAECC-13),*. London, UK: Springer-Verlag, 1999, pp. 65–76.

[15] A. Shokrollahi, "Raptor codes," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2551–2567, Jun. 2006.

[16] M. Luby, "LT codes," in *Proc. the 43rd Annu. IEEE Symposium on Foundations of Computer Science (FOCS),*, Vancouver, BC, Canada, Nov. 2002, pp. 271–280.

[17] N. Kaplan, "A generalization of a result of Erdos and Renyi," *Journal of Applied Probability*, vol. 14, pp. 212–216, 1977.

[18] L. Flatto, "Limit theorems for some random variables associated with urn models," *The Annals of Probability*, vol. 10, no. 4, pp. 927–934, 1982.

[19] J. F. Cheng and R. J. Mceliece, "Some high-rate near capacity codecs for the gaussian channel," in *Proc. 34th Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, USA, Oct. 1996, pp. 494–503.

[20] S. Sanghavi, "Intermediate performance of rateless codes," in *Proc. IEEE Information Theory Workshop (ITW '07),*, Sep. 2007, pp. 478–482.

[21] K. Mahdaviani, M. Ardakani, and C. Tellambura, "On Raptor code design for inactivation decoding," *IEEE Trans. Commun.*, 2012, accepted for publication.