

# Finite Length Performance of Random MAC Strategies

Konstantinos Dovelos<sup>1</sup>, Laura Toni<sup>2</sup>, and Pascal Frossard<sup>3</sup>,

<sup>1</sup>*Aristotle University of Thessaloniki (AUTH), Greece*

<sup>2</sup>*Electronic & Electrical Eng. Department, UCL, UK*

<sup>3</sup>*Signal Processing Laboratory (LTS4), EPFL, Switzerland*

kdovelos@ece.auth.gr, l.toni@ucl.ac.uk, pascal.frossard@epfl.ch

**Abstract**—The Internet of Things (IoT) is fueling innovation in nearly every part of our lives. From smart homes, cars, and cities, the Internet of Things is creating a more convenient, secure, intelligent, and personalized experience. While for any final user this IoT vision is a substantial innovation step, for communication providers is a compelling thread with massive number of dynamic devices connected to the Internet. Multiple connected devices sharing common wireless resources might create interference if they access the channel simultaneously. Medium access control protocols generally regulate the access of the devices to the shared channel to limit signal interference. In particular, irregular repetition slotted ALOHA (IRSA) techniques can achieve high-throughput performance when interference cancellation methods are adopted to recover from collisions. In this work, we study the finite length performance for IRSA schemes by building on the analogy between successive interference cancellation and iterative belief-propagation on erasure channels. We use a novel combinatorial derivation based on the matrix-occupancy theory to compute the error probability and we validate our method with simulation results.

## I. INTRODUCTION

In the IoT vision, 50 billions of devices are expected to be connected to the Internet by 2020, smartly generating, sharing and processing data [1]. With current networks reaching congestions with only few hundreds of connected devices [2], evidently there is the need to develop novel communication paradigms to sustain this IoT avalanche. Medium access control (MAC) strategies for example should lead to efficient but also distributed (i.e., scalable) ways of accessing the communication channel. When networked devices share common wireless resources, signal interference might be experience. MAC strategies need to properly control users transmission to limit this interference [3]–[5]. However, in future networks a massive number of devices will be connected to the Internet (e.g., Internet of Things and machine-to-machine communications) and MAC protocols need to be more and more distributed. Random slotted ALOHA (SA) with successive interference cancellation (SIC) strategies, for example, have recently gained attention because they do not require coordination, and they are able to recover from interfering signals.

Bipartite graphs are a useful framework to study random MAC strategies or, more generally, transmission of successive signals from several sources in different time slots. When

edges in the bipartite graph are randomly generated, the analysis of belief propagation (BP) decoding is usually performed asymptotically, i.e., for an infinite number of sources and time slots. Finite length analysis has been investigated when edges are randomly selected from the transmission time slots, as the case of finite length analysis for LDPC codes [6]. However, the reverse case in which the source nodes randomly create the edges is still an open topic that we address in this work.

In this work, we consider random SA with SIC strategies as the main target application, where each source sends information to a central base station (BS) in time slots that are uniformly selected at random independently from the other sources. Packets sent in the same time slot from different users interfere among each others and cannot be immediately decoded. However, SIC strategies are able to mitigate the effect of these collisions through iterative message-passing techniques and recover corrupted data at the decoder. Within this framework, we study the decoding performance of BP schemes in finite length settings, namely for small MAC frame size. Within a MAC frame, each source follows a transmission probability distribution that drives the replication rate of the sources, hence the performance of the system. Our objective is to compute the decoding error probability, i.e., the probability of not decoding correctly the source information. We first introduce a combinatorial derivation of the packet collision probability using the matrix occupancy framework. Then, we evaluate iteratively the decoding error probability by studying the number of collisions that can actually be resolved by interference cancellation. The proposed analysis is exact but it has a computational complexity that grows with the MAC frame size. We therefore show how achieve an approximated but still accurate analysis at a reduced computational cost. Simulation results validate our study in different transmission settings with small MAC frames.

In the seminal work of [7], a key connection has been drawn between SIC strategies in irregular repetition slotted ALOHA (IRSA) and the iterative BP decoder of erasure codes on graphs. This has opened the possibility to apply theory of rateless codes to IRSA schemes and analyze their performance [8], [9], which is essential to optimize users' transmission strategy (e.g., transmission probability) [10]. These works are mainly focused on deriving asymptotic system performance for

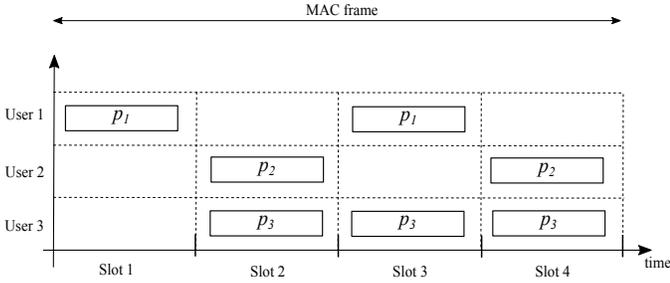


Fig. 1. Transmission example of IRSA strategy with a MAC frame composed of four slots. Source  $i$  sends the source packet  $p_i$ . There are three users attempting a transmission according to the degree vector  $\mathbf{d} = [2, 2, 3]$ .

large MAC size frames. They however cannot be easily applied in optimizing resource allocation strategies in actual IRSA schemes, as shown in [10]. To the best of our knowledge, only the works in [11], [12] investigated finite-length performance analysis for IRSA scheme. Both look at the average stopping sets and derive an upper bound on the error probability in IRSA. These bounds have low computational complexity but they are not necessarily tight for very small MAC frames. In our work, we rather derive a semi-analytic analysis for finite length IRSA schemes, which permits to compute error probabilities exactly, even for small frames.

## II. SYSTEM MODEL

We consider a system of  $k$  sources that communicate with a common BS. The IRSA strategy is the adopted MAC protocol [7]. We assume the time axis to be discretized in MAC frames, each of those composed of  $t$  time slots. Within a MAC frame, each source transmits  $d$  replicas of the source packet  $p$ , as depicted in Fig. 1. The  $d$  distinct time slots used for transmission are selected uniformly at random among the  $t$  total available slots. The replication rate  $d$  is randomly selected by each user following the transmission probability distribution  $\Lambda = [\Lambda_1, \dots, \Lambda_{D_{max}}]$ , where  $\Lambda_d$  is the probability that a user transmits  $d$  replicas, and  $D_{max}$  is the maximum number of allowed packet replicas per MAC frame. Within a MAC frame, each source selects its replication rate independently from the others, leading to replication vector (named in the following as source degree vector)  $\mathbf{d} = [d_1, \dots, d_k]$ ,  $d_i \in \{1, \dots, D_{max}\}$  that is experienced with probability

$$P_{\Lambda}(\mathbf{d}) = \prod_{i=1}^k P_{\Lambda}(d_i) = \prod_{i=1}^k \Lambda_{d_i}.$$

Each realization of  $k$  sources accessing the time slots of a MAC frame can be described by a  $k \times t$  binary matrix  $\mathbf{M} = (m_{ij})$ , called *collision matrix*, with rows and columns corresponding to users and slots, respectively. We have  $m_{ij} = 1$  if the  $i$ th user transmits in the  $j$ th slot, and  $m_{ij} = 0$  otherwise. The collision matrix  $\mathbf{M}$  associated with the example in Fig. 1 is given by

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}.$$

The weight of a column  $\mathbf{m}_j$  in  $\mathbf{M}$  is given by  $\sum_{i=1}^k m_{ij}$  and it represents the number of packets sent in the time slot  $j$ . Thus, columns with unity weight, e.g.,  $[100]^T$ , represent singleton slots that allow an immediate decoding of the message. On the contrary, columns with a weight greater than one, e.g.,  $[110]^T$ , represent slots in which messages collide and cannot be directly decoded. Collided messages can however be recovered by SIC strategies. If packets are sent by two users in the same time slot but one of them can be recovered from a singleton slot, then the second packet can be decoded by interference cancelation. For example, message  $p_1$  in  $\mathbf{M}$  is recovered from the first slot, which is a singleton one. Then, canceling the message  $p_1$  from the other interfering messages we obtain

$$\mathbf{M}' = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

and message  $p_3$  can also be decoded. As long as one singleton slot is experienced, the iterative decoding process proceeds. If the SIC process resolves all collisions, then no source packets are lost within the MAC frame of interest. If the SIC process stops before completion, it leaves packets undecoded and the SIC process fails.

In this work, we are interested in evaluating the probability of failure in the SIC process, i.e., the probability that a packet is lost when transmitted through the IRSA protocol. We denote this packet loss rate (PLR) by  $P_L$ , and it can be written as<sup>1</sup>

$$P_L = \sum_{u=2}^k \frac{u}{k} P_{\Lambda}(u) \quad (1)$$

where  $P_{\Lambda}(u)$  is the probability of having  $u$  unrecovered packets when  $k$  users transmit over a frame of  $t$  slots with degree distribution  $\Lambda$ . We condition to a given degree distribution vector as follows

$$\begin{aligned} P_L &= \sum_{\mathbf{d} \in \mathcal{D}} \sum_{u=2}^k \frac{u}{k} P(u | \mathbf{d}) P(\mathbf{d}) \\ &= \sum_{\mathbf{d} \in \mathcal{D}} \sum_{u=2}^k \frac{u}{k} P(u | \mathbf{d}) \prod_{i=1}^k \Lambda_{d_i} \end{aligned} \quad (2)$$

with  $\mathcal{D}$  denoting the set of all the possible packet repetition vectors allowed by the distribution  $\Lambda$ . Denoting by  $D$  the number of possible replication rates, i.e., replication rates with  $\Lambda_d > 0$ ,  $|\mathcal{D}| = D^k$ . In the next section, we compute the PLR  $P_L$  for small MAC frame size  $t$ .

<sup>1</sup>For the sake of notation, we omit the dependency of the packet loss probabilities on  $(k, t)$ .

$$P(\hat{\mathbf{n}}|\mathbf{d}) = \begin{cases} \left[ \prod_{j=1}^k \binom{t}{d_j} \right]^{-1} \frac{t!}{\prod_{\mathbf{c}_q \in \hat{\mathcal{C}}} n_{\mathbf{c}_q}! \prod_{i=1}^{k+1} f_i(\hat{\mathbf{n}}, \mathbf{d})!} & \text{if } \mathcal{I}(\hat{\mathbf{n}}) = 1 \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

### III. FINITE LENGTH PERFORMANCE

#### A. Matrix-Based Formulation

Because of the source independence, collision matrices are equivalent in terms of PLR upon permutations (both across rows or columns). We can therefore study the IRSA performance by only looking at the column vectors which are present within a given matrix  $\mathbf{M}$ . This is possible exploiting the combinatorial matrix-occupancy theory [13], dealing with sets of balls randomly assigned into groups of bins. Random access channel problems can be viewed as occupancy problems by considering packets and slots as balls and bins, respectively. The number of bins with only one ball, for example, represents the number of singleton slots.

In more details, let  $\mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_{|\mathcal{C}|}\}$  be the set of all possible column vectors that can be present in  $\mathbf{M}$ , with column  $\mathbf{c}_q = [c_q^{(1)}, c_q^{(2)}, \dots, c_q^{(k)}]^T$  taking values in  $\{0, 1\}^k$ . Let us then define the *occupancy vector*

$$\mathbf{n} = [n_{\mathbf{c}_1}, n_{\mathbf{c}_2}, \dots, n_{\mathbf{c}_{|\mathcal{C}|}}]$$

associated with a matrix  $\mathbf{M}$  as a vector that shows how many times each column in  $\mathcal{C}$  is present in  $\mathbf{M}$ . Note that for the sake of notation, we omit the dependency of  $\mathbf{n}$  from  $\mathcal{C}$ . Specifically,  $n_{\mathbf{c}_q}$  is the number of times the column  $\mathbf{c}_q$  is present in the matrix of interest. For example, defining  $\mathbf{c}_1 = [1 \ 0 \ 0]^T$ ,  $\mathbf{c}_2 = [0 \ 1 \ 1]^T$ , and  $\mathbf{c}_3 = [1 \ 0 \ 1]^T$ , the occupancy vector associated with  $\mathbf{M}$  is

$$\mathbf{n} = [n_{\mathbf{c}_1} = 1, n_{\mathbf{c}_2} = 2, n_{\mathbf{c}_3} = 1, n_{\mathbf{c}_{q,q>3}} = 0].$$

Finally, we define  $\mathcal{C}_l \subseteq \mathcal{C}$  as the subset of column vectors with weight  $w(\mathbf{c}_q) = \sum_{j=1}^k c_q^{(j)} = l$ , and  $\mathcal{C}_{l,i} \subseteq \mathcal{C}_l$  as the subset of column vectors with weight  $l$  and  $c_i = 0$ . It is worth noting that each occupancy vector corresponds to multiple collision matrices that are equivalent in terms of PLR.

We are now interested in finding conditions under which an occupancy vector represents a collision matrix in the case of  $k$  sources,  $t$  time slots, and degree vector  $\mathbf{d}$ . First, we impose that exactly  $t$  columns are present in the matrix:

$$\sum_{q: \mathbf{c}_q \in \mathcal{C}} n_{\mathbf{c}_q} = t \quad (3)$$

Then, we impose that the degree vector is respected. This means that an occupancy vector is feasible if it leads to a matrix in which exactly  $d_i$  entries are non-zero in the  $i$ th row of the collision matrix. This translates in the following set of constraints

$$\sum_{l=1}^{D_{max}} \sum_{q: \mathbf{c}_q \in \mathcal{C}_{l,i}} n_{\mathbf{c}_q} = t - d_i, \quad i = 1, \dots, k. \quad (4)$$

Since  $\mathcal{C}_k$  has only one column vector (i.e., the vector with all 1 entries) and  $\mathcal{C}_{k-1}$  has  $k$  possible column vectors (i.e., each vector with only one out of  $k$  null entry), we can impose the above  $k+1$  constraints — (3) and (4) — by properly evaluating the occupancy of the  $k+1$  column vectors in  $\mathcal{C}_k$  and  $\mathcal{C}_{k-1}$ . Let us denote by  $\hat{\mathbf{n}}$  the reduced occupancy vector, defined as the column vectors with weight at most  $k-2$ . Formally,  $\hat{\mathbf{n}} = [n_{\mathbf{c}_q}]_{\mathbf{c}_q \in \hat{\mathcal{C}}}$ , with  $\hat{\mathcal{C}} = \mathcal{C} \setminus \mathcal{C}_{k-1} \cup \mathcal{C}_k$ . We can then decompose any occupancy vector as  $\mathbf{n} = [\hat{\mathbf{n}} \mathbf{f}(\hat{\mathbf{n}}, \mathbf{d})]$ , with  $f(\hat{\mathbf{n}}, \mathbf{d})$  representing the occupancy of the  $k+1$  column vectors in  $\mathcal{C}_k$  and  $\mathcal{C}_{k-1}$ . These  $k+1$  unknown  $\mathbf{f}(\hat{\mathbf{n}}, \mathbf{d}) = [f_1(\hat{\mathbf{n}}, \mathbf{d}), \dots, f_{k+1}(\hat{\mathbf{n}}, \mathbf{d})]$  are derived by imposing the constraints (3) and (4). If  $f_i(\hat{\mathbf{n}}, \mathbf{d}) \geq 0, \forall i$ , then the occupancy vector  $[\hat{\mathbf{n}} \mathbf{f}(\hat{\mathbf{n}}, \mathbf{d})]$  is a feasible one for the transmission settings  $(k, t, \mathbf{d})$ . We define  $\mathcal{I}(\hat{\mathbf{n}})$  an indicator function such that  $\mathcal{I}(\hat{\mathbf{n}}) = 1$  if  $[\hat{\mathbf{n}} \mathbf{f}(\hat{\mathbf{n}}, \mathbf{d})]$  is a feasible one for the transmission settings  $(k, t, \mathbf{d})$ , and  $\mathcal{I}(\hat{\mathbf{n}}) = 0$ , otherwise.

#### B. Packet Loss Probability

Equipped with the matrix-occupancy representation, we can express the error probability  $P(u | \mathbf{d})$  in (2) as

$$P(u | \mathbf{d}) = \sum_{\hat{\mathbf{n}}} Q_u(k, [\hat{\mathbf{n}} \mathbf{f}(\hat{\mathbf{n}}, \mathbf{d})]) P(\hat{\mathbf{n}} | \mathbf{d})$$

where  $P(\hat{\mathbf{n}}|\mathbf{d})$  is the probability of experiencing an occupancy vector  $[\hat{\mathbf{n}} \mathbf{f}(\hat{\mathbf{n}}, \mathbf{d})]$ , when  $k$  users transmit over  $t$  slots given the repetition vector  $\mathbf{d}$ . The indicator function  $Q_u(k, \mathbf{n})$  returns 1 if the the SIC process with a collision matrix associated with  $\mathbf{n}$  stops at  $u$  undecoded packets and returns 0 otherwise. We compute both terms below.

The probability  $P(\hat{\mathbf{n}}|\mathbf{d})$  is zero if  $\mathcal{I}(\hat{\mathbf{n}}) = 0$ , otherwise it is evaluated as the ratio between the number of collision matrices with occupancy vector  $[\hat{\mathbf{n}} \mathbf{f}(\hat{\mathbf{n}}, \mathbf{d})]$  and the total number of collision matrices in the same transmission settings. The former is given by the following multinomial coefficients

$$\frac{t!}{\prod_{\mathbf{c}_q \in \hat{\mathcal{C}}} n_{\mathbf{c}_q}! \prod_{i=1}^{k+1} f_i(\hat{\mathbf{n}}, \mathbf{d})!}$$

while the total number of collision matrices that can be experienced under the settings  $(k, t, \mathbf{d})$  is  $\prod_{j=1}^k \binom{t}{d_j}$  from the independency of the sources. This leads to (5), at the top of the page.

We then derive  $Q_u(k, \mathbf{n})$  iteratively. We consider the  $j$ th iteration of the decoding process, where  $k-j$  packets are undecoded, and  $\mathbf{n}^{(j)} = [n_{\mathbf{c}_1}^{(j)}, n_{\mathbf{c}_2}^{(j)}, \dots]$  is the occupancy vector of the collision matrix at the  $j$ th decoding step. Note that  $\mathbf{n} = \mathbf{n}^{(0)}$  is the occupancy vector before the decoding process starts. At the  $j$ th iteration of the decoding process, one

$$P_L = \sum_{\mathbf{d} \in \mathcal{D}} \sum_{u=2}^k \frac{u}{k} \sum_{\hat{\mathbf{n}} \in \mathcal{N}} Q_u(k, [\hat{\mathbf{n}} \mathbf{f}(\hat{\mathbf{n}}, \mathbf{d})]^{(0)}) \frac{t!}{\prod_{\mathbf{c}_q \in \hat{\mathcal{C}}} n_{\mathbf{c}_q}! \prod_{i=1}^{k+1} f_i(\hat{\mathbf{n}}, \mathbf{d})!} \prod_{i=1}^k \frac{\Lambda_{d_i}}{\binom{t}{d_i}} \quad (9)$$

message is decoded only if there exists at least one weight-1 column vector, i.e., if  $\exists \mathbf{c} \in \mathcal{C}_1$  s.t.  $n_{\mathbf{c}}^{(j)} > 0$ .

If the condition is satisfied, then the decoder can proceed to the next step. At the decoding iteration  $j+1$ , there are  $k-j-1$  undecoded packets and the occupancy vector of the collision matrix is denoted by  $\mathbf{n}^{(j+1)}$ . The latter is derived recursively from  $\mathbf{n}^{(j)}$ . Let us consider the column vector with the  $m$ -th entry being non-zero, i.e.,  $\mathbf{c} \in \mathcal{C} \setminus \cup_l \mathcal{C}_{l,m}$ , and let us denote by  $\bar{\mathbf{c}}^{(m)}$  its complementary in  $m$  a column vector equal to  $\mathbf{c}$  but with the  $m$ -th entry set to zero. For example, if  $\mathbf{c} = [11001]$ , then  $\bar{\mathbf{c}}^{(2)} = [10001]$ . Then, in the case in which the  $m$ -th element of  $\mathcal{C}_1$  has  $n_{\mathbf{c}_m}^{(j)} > 0$ ,  $\mathbf{n}^{(j+1)}$  can be written from  $\mathbf{n}^{(j)}$  as follows

$$\begin{aligned} n_{\bar{\mathbf{c}}^{(m)}}^{(j+1)} &= n_{\mathbf{c}}^{(j)} + n_{\bar{\mathbf{c}}^{(m)}}^{(j)} & \forall \mathbf{c} \in \mathcal{C} \setminus \cup_l \mathcal{C}_{l,m} \\ n_{\bar{\mathbf{c}}^{(m)}}^{(j+1)} &= 0 \\ n_{\mathbf{c}}^{(j+1)} &= n_{\mathbf{c}}^{(j)} & \forall \mathbf{c} \in \{\cup_l \mathcal{C}_{l,m} \setminus \bar{\mathbf{c}}^{(m)}\} \end{aligned} \quad (6)$$

We thus recursively evaluate the indicator function  $Q_u$  as

$$Q_u(k-j, \mathbf{n}^{(j)}) = Q_u(k-j-1, u, \mathbf{n}^{(j+1)}). \quad (7)$$

If there are no weight-one columns in the collision matrix, the decoder terminates at iteration  $j$  with  $k-j$  undecoded packets and  $Q_u$  becomes

$$Q_u(k-j, \mathbf{n}^{(j)}) = \begin{cases} 1, & k-j = u \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

Finally, denoting by  $\mathcal{N}$  the set of reduced occupancy vectors  $\hat{\mathbf{n}}$  such that  $\mathcal{I}(\hat{\mathbf{n}}) = 1$ , the decoding error probability of (2) results in (9), provided at the top of the page.

We now comment on the complexity of the proposed semi-analytical study. Both the combinatorial and iterative steps in (9) are performed over all possible degree vectors  $\mathbf{d} \in \mathcal{D}$  and all possible reduced occupancy vectors  $\hat{\mathbf{n}} \in \mathcal{N}$ . The cardinality of  $\mathcal{D}$  and  $\mathcal{N}$  is given respectively by

$$|\mathcal{D}| = D^k \quad \text{and} \quad |\mathcal{N}| \leq \binom{\hat{\mathcal{C}} + t - 1}{t}$$

with  $\hat{\mathcal{C}} = \sum_{n=0}^{k-2} \binom{t}{n}$ . The upper bound on  $|\mathcal{N}|$  is derived as follows. We first recall that  $\hat{\mathcal{C}}$  is the dimension of the reduced occupancy vector  $\hat{\mathbf{n}}$  and that the entries of  $\hat{\mathbf{n}}$  need to satisfy (3). Looking at the problem as  $t$  balls into  $R$  bins, the number of possible combinations of the reduced vector is  $\binom{\hat{\mathcal{C}}+t-1}{t}$ . Among these, only the reduced occupancy vectors that satisfy (4) belong to  $\mathcal{N}$ .

It is worth noting that the cardinality of  $\mathcal{D}$  and  $\mathcal{N}$  both scales with  $k$  and  $t$ . However, the probability of experiencing a given reduced vector and a given degree vector can be easily

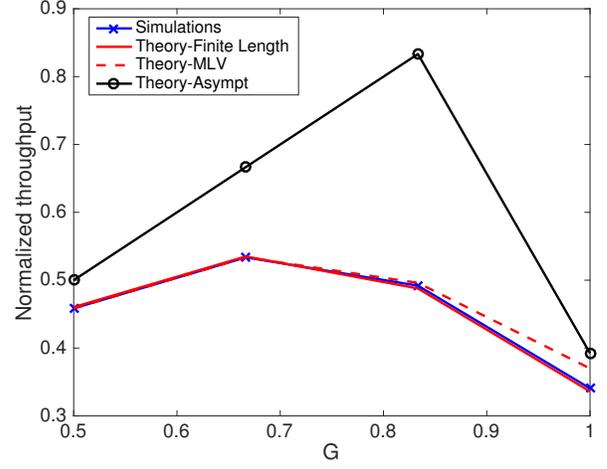


Fig. 2. Comparison of the theoretical and simulation results for different  $(k, t)$  pairs in the case of  $t = 6$  and  $\Lambda(x) = 0.2x + 0.5x^2 + 0.3x^4$ .

derived from (2). Therefore, an approximated PLR can be evaluated by performing the iterative procedure  $Q_u(k, \mathbf{n})$  only for the most likely reduced vectors. This substantially reduces the computational complexity while preserving accuracy.

### C. Ripple-Based Analysis

We propose a ripple-based analysis similar to the one in the finite-length analysis of Luby-Transform (LT) codes. Let us first introduce some essential notation. In particular, the MAC frame status can be conveniently represented by a bipartite graph  $\mathcal{G} = (B, S, E)$ , consisting of a set  $B$  of  $k$  burst nodes (one for each user), a set  $S$  of  $N$  slot nodes (one for each slot), and a set  $E$  of edges. An edge connects a burst node  $b_i \in B$  to a sum node  $s_j \in S$  if and only if a burst replica of the  $i$ -th user is transmitted in the  $j$ -th slot.

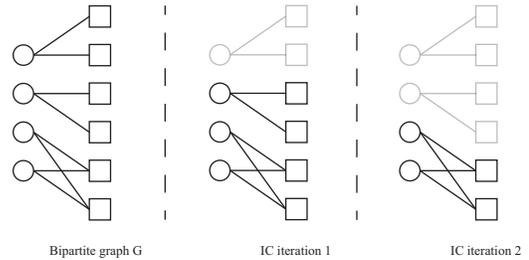


Fig. 3. Graph representation of the IC process for  $k = 4$  and  $N = 4$ .

We will call step  $u$  the time point where there are  $u$  unknown (not decoded yet) BNs in the graph. Also, we will

call the ripple the set of BNs that are connected to at least one SN of degree one and will denote its size at step  $u$  by  $X_u$ . Now, we can express the failure probability  $P(u)$  by means of the ripple size. More particularly, the IC process will fail with  $u$  undecoded users if the ripple size at step  $u$  is zero. Therefore, we can write

$$P(u-1) = \Pr(u-1, X_{u-1} = 0) \quad (9)$$

We model the IC process by the discrete-time stochastic process  $\{u, X_u, N_u, \mathbf{d}_u\}_{0 \leq u \leq k}$ , with  $N_u$  denoting the number of SNs that are present in the bipartite graph at step  $u$ . Looking in detail the decoding algorithm, we can see that there is a dependency between successive states, i.e., the states  $(u-1, X_{u-1}, N_{u-1}, \mathbf{d}_{u-1})$  and  $(u, X_u, N_u, \mathbf{d}_u)$ . Let us recall that during the iteration  $u \rightarrow u-1$ , one BN leaves the ripple and the associated edges are removed from the graph. This might lead some other BNs, i.e., those that have common SNs of degree two with the selected ripple element, to join the ripple. Let  $Y_u$  denote the number of BNs that join the ripple during the iteration  $u \rightarrow u-1$ . Then,  $X_{u-1}$  is

$$X_{u-1} = X_u + Y_u - 1 \quad (10)$$

with  $0 \leq Y_u \leq u-1$ . Similarly, there is a change in the number of SNs that are available in the bipartite graph. By definition the selected ripple BN is connected to at least one SN of degree one. After the deletion of its edges, those degree-1 SNs will be degree-0 nodes and hence they will not participate in the reduced graph. Let  $Z_u$  denote the number of the SNs that become of reduced degree zero after the iteration  $u \rightarrow u-1$ . Then,  $N_{u-1}$  is

$$N_{u-1} = N_u - Z_u \quad (11)$$

with  $1 \leq Z_u \leq l_{max}$ , where  $l_{max}$  is the maximum degree allowed by the distribution  $\{\Lambda_l\}$ .

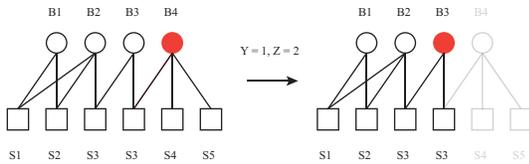


Fig. 4. One-step evolution of the bipartite graph. Red circles denote BNs that are in the ripple.

Based on equations (10) and (11), we may assume that the sequence

$$(u-1, X_{u-1}, N_{u-1}, \mathbf{d}_{u-1}), (u, X_u, N_u, \mathbf{d}_u), \dots, (k, X_k, N_k, \mathbf{d}_k)$$

forms a *Markov chain*. By exploiting the Markov property, we can derive a recursion for the evaluation of the distribution  $Q(u-1, r, n, \mathbf{d}) = \Pr(u-1, X_{u-1} = r, N_{u-1} = n, \mathbf{d})$ . The

recursion of the state distribution at step  $u-1$  is given by

$$\begin{aligned} Q(u-1, r, n, \mathbf{d}_{u-1}) &= \sum_{\mathbf{d}_u \in \mathcal{D}_u} \sum_s \sum_m \Pr(u-1, X_{u-1} = r, N_{u-1} = n, \mathbf{d}_{u-1} | u, X_u = r_u, N_u = n_u, \mathbf{d}_u = \mathbf{d}_u) \\ &= \sum_{\mathbf{d}_u \in \mathcal{D}_u} \sum_s \sum_m \Pr(Y_u = r - s + 1, Z_u = n - n_u, \mathbf{d}_{u-1} | u, X_u = r_u, N_u = n_u, \mathbf{d}_u = \mathbf{d}_u) \end{aligned} \quad (12)$$

where we neglected the limits of the sums for the sake of notation simplicity. Finally, the failure probability  $P(u-1)$  is given by

$$\begin{aligned} P(u-1) &= \Pr(u-1, X_{u-1} = 0) \\ &= \sum_n \sum_{\mathbf{d}_{u-1} \in \mathcal{D}_{u-1}} Q(u-1, 0, n, \mathbf{d}_{u-1}) \end{aligned} \quad (13)$$

Let us notice that the decoder fails to complete the IC operation at step  $u$ ,  $u = k, \dots, 2$ , independently on the order in which decodes the BNs of the ripple. This enable us to evaluate equation (13) assuming that always the first element of the ripple is selected for being decoded during an iteration. Furthermore, we can make similar assumptions about the configuration of the BNs that compose the ripple and the BNs that join the ripple at each step of the iterative decoding process. The probability of having a specific combination of BNs in the ripple, or in the set of nodes that join the ripple, is the same whatever combination is considered, since each BN chooses randomly its degree following  $\{\Lambda_l\}$  and the edges are distributed uniformly among the SNs. With out loss of generality, we may assume the followings:

- 1) Let  $\{B_1, \dots, B_k\}$  be the set of the BNs of the bipartite graph. The IC process starts with  $B_1, \dots, B_r$  being in the ripple and  $B_{r+1}, \dots, B_k$  being in the cloud.
- 2) At each subsequent step  $u \leq k$  of the process, the  $Y_u$  BNs that join the ripple during the iteration  $u \rightarrow u-1$  are the first ones from the cloud.

Using assumptions 1 and 2 we can evaluate the Markov chain on a fixed chain path regarding the BNs in the ripple and the BNs that join the ripple during an iteration and after multiply the result by a product of binomial coefficients in order to take into account all the equally probable chain paths. In sequel, we show how the idea of the fixed chain path is mathematically expressed.

Let  $r_u$  and  $n_u$  denote the values of the ripple size and the length of the MAC frame at step  $u$  respectively. Then,

$$\begin{aligned} &= \sum_{\mathbf{d}_u \in \mathcal{D}_u} \sum_{r_u} \sum_{n_u} \Pr(u-1, X_{u-1} = 0, N_{u-1} = n_{u-1}, \mathbf{d}_{u-1} | u, X_u = r_u, N_u = n_u, \mathbf{d}_u = \mathbf{d}_u) \\ &= \sum_{\mathbf{d}_u \in \mathcal{D}_u} \sum_{r_u} \sum_{n_u} \Pr(Y_u = 0, Z_u = n_u - n_{u-1}, \mathbf{d}_{u-1} | u, X_u = r_u, N_u = n_u, \mathbf{d}_u = \mathbf{d}_u) \\ &= \binom{u-r_u}{0} \sum_{l: \Lambda_l \neq 0} \sum_{r_u} \sum_{n_u} \Pr(Y_u = 0, Z_u = n_u - n_{u-1} | u, X_u = r_u, N_u = n_u, \mathbf{d}_u = \mathbf{d}_u) \end{aligned} \quad (14)$$

TABLE I  
 $k = 4, t = 6, \Lambda(x) = 0.25x^2 + 0.75x^3$ .

u	Pr[U = u] theory	Pr[U = u] sim
2	0.140730	0.141390
3	0.130158	0.130110
4	0.094203	0.093460
$P_L$	0.262186	0.261738

TABLE II  
 $k = 5, t = 5, \Lambda(x) = 0.45x^2 + 0.55x^3$ .

u	Pr[U = u] theory	Pr[U = u] sim
2	0.078781	0.078510
3	0.177389	0.176970
4	0.346640	0.346430
5	0.397189	0.398090
$P_L$	0.812448	0.812820

#### IV. NUMERICAL RESULTS

We now provide the simulation results to validate the proposed solution in finite-length systems, i.e., with small size MAC frames  $t \in [4, 7]$ . We consider different settings with  $k$  sources and  $t$  time slots. For each  $(k, t)$  pair we consider different transmission probabilities, i.e., different degree distributions  $\Lambda(x)$ , following [7]. For each of these scenarios, we evaluate the decoding error probability from (9). Then, for each  $\Lambda(x)$ , we generate 1000 realizations of collision matrices and simulate the IRSA protocol and the SIC decoding with belief propagation and we evaluate  $u/k$ . We then average this ratio over the 1000 realization to evaluate the average loss probability.

We now provide simulation results in terms of normalized throughput, defined as  $(1 - P_L)k/t$ . This metric is usually adopted to evaluate the performance of MAC strategies and it directly reflects the error probability  $P_L$ . In Fig. 2, we provide the normalized throughput as a function of the traffic  $G = k/t$  for a scenario with  $t = 6$  and  $\Lambda(x) = 0.2x + 0.5x^2 + 0.3x^4$ . Results are provided for both simulation results and theoretical ones, namely the finite length analysis proposed in this work and the asymptotic analysis derived in [7]. We also provide an approximated solution (labeled MLV — most likely vectors), where the iterative evaluation of  $Q_u$  in (9) is performed only over the occupancy vector with a probability  $P(\mathbf{d}) \geq 10^{-3}$ . The results show a weak match between asymptotic theory and the simulations results, from here the need for finite length analysis. From the results, we also observe a good match between finite length theory (both exact and approximated) and simulations, showing the accuracy of our study. The model is validated also in the results provided from Table I, where we provide the final packet loss rate  $P_L$  but also a partial performance of the decoding process (i.e., the probability of stopping the decoding step at  $u$  unknown denoted by  $Pr[U = u]$ ). The good match between theory and simulation is confirmed in these experiments as well as in the ones of Table II, where a different  $\Lambda(x)$  has been simulated.

Finally, in Table VI we compare our analysis with the

TABLE III  
 $k = 4, t = 6, \Lambda(x) = 0.6x^2 + 0.4x^3$ .

u	Pr[U = u] theory	Pr[U = u] Markovian	Pr[U = u] sim
2	0.1992	0.1972	0.1993
3	0.12063	0.2063	0.2062
4	0.1627	0.1627	0.1628

TABLE IV  
 $k = 5, t = 6, \Lambda(x) = 0.35x^2 + 0.65x^3$ .

u	Pr[U = u] theory	Pr[U = u] Markovian	Pr[U = u] sim
2	0.1209	0.1265	0.1210
3	0.1778	0.1901	0.1777
4	0.2359	0.2359	0.2363
5	0.1820	0.1280	0.1815

asymptotic analysis of [7] and the finite-length analysis of [11]. We see that, especially for small value of the traffic network  $G$ , the asymptotic analysis is far away from the actual performance, and that our study is more precise than [11] especially for large values of the traffic network  $G$ . This accuracy comes at a price of a large computational complexity. Because of the complexity factor, (9) might be too expensive to evaluate for realistic MAC frames (hundreds of time slots). However, in Table VI we observe that the approximated solution MLV nicely scales with the MAC frame without significantly affecting the accuracy.

#### V. CONCLUSIONS

We carried out an evaluation of the IRSA performance in finite-length settings, using combinatorial theory and matrix-occupancy theory. Simulation results validate the derived analysis for small MAC frames and show the improved match between theory and simulation results with respect to the state of the art performance studies. This work is a first step in the unbeaten research path of IRSA performance in finite-length settings. The current study leads to a mix of combinatorial and analytical performance. Future works will focus on providing a pure theoretical evaluation based as well as low-complexity solution that can be extended to larger number of MAC frames.

#### VI. ACKNOWLEDGMENTS

This work was partially funded by the Swiss National Science Foundation (SNSF) under the CHIST- ERA project CONCERT (A Context-Adaptive Content Ecosystem Under Uncertainty), project nr. FNS 20CH21 151569.

#### REFERENCES

- [1] D. Evans, "The internet of things: How the next evolution of the internet is changing everything," *Cisco*, 2011.
- [2] M. Polese, M. Centenaro, A. Zanella, and M. Zorzi, "M2M massive access in LTE: RACH performance evaluation in a smart city scenario," in *Proc. IEEE Int. Conf. on Commun.*, 2016.
- [3] A. S. Gupta and A. Singer, "Successive interference cancellation using constellation structure," *IEEE Trans. on Signal Processing*, vol. 55, no. 12, pp. 5716–5730, Dec 2007.

TABLE V  
COMPARISON OF THE DECODING FAILURE PROBABILITY  $P_L$  BOTH FROM THEORY OR BY SIMULATION.

$G$	Simulation	[7]	[11]	(9)	MLV
0.5	0.13	0	0.17	0.14	0.14
0.67	0.34	0	0.58	0.35	0.35
0.8	0.75	0.44	0.98	0.74	0.77

TABLE VI  
COMPARISON OF THE DECODING FAILURE PROBABILITY  $P(u)$  BOTH FROM OPTIMAL AND SUBOPTIMA THEORY OR BY SIMULATION.

$G$	Simulation	[7]	[11]	(9)	MLV
0.5	0.13	0	0.17	0.14	0.14
0.67	0.34	0	0.58	0.35	0.35
0.8	0.75	0.44	0.98	0.74	0.77

- [4] C. H. Yih, "Iterative interference cancellation for OFDM signals with blanking nonlinearity in impulsive noise channels," *IEEE Signal Processing Letters*, vol. 19, no. 3, pp. 147–150, March 2012.
- [5] P. Moulin, "Signal transmission with known-interference cancellation," *IEEE Signal Processing Magazine*, vol. 24, no. 1, pp. 134–136, Jan 2007.
- [6] T. Richardson and R. Urbanke, *Modern coding theory*. Cambridge University Press, 2008.
- [7] G. Liva, "Graph-based analysis and optimization of contention resolution diversity slotted ALOHA," *IEEE Trans. Commun.*, vol. 59, no. 2, pp. 477–487, 2011.
- [8] E. Paolini, G. Liva, and M. Chiani, "Coded slotted ALOHA: A graph-based method for uncoordinated multiple access," *IEEE Trans. Inform. Theory*, vol. 61, no. 12, pp. 6815–6832, 2015.
- [9] C. Stefanovic and P. Popovski, "ALOHA random access that operates as a rateless code," *IEEE Trans. Commun.*, vol. 61, no. 11, pp. 4653–4662, November 2013.
- [10] L. Toni and P. Frossard, "Prioritized random MAC optimization via graph-based analysis," *IEEE Trans. Commun.*, vol. 63, no. 12, pp. 5002–5013, Dec 2015.
- [11] E. Paolini, "Finite length analysis of irregular repetition slotted aloha (IRSA) access protocols," in *Proc. IEEE Int. Conf. Communication Workshop*, 2015, pp. 2115–2120.
- [12] M. Ivanov, F. Brannstrom, A. G. i Amat, and P. Popovski, "Broadcast coded slotted ALOHA for vehicular communications: A finite frame length analysis," *ArXiv*, vol. /1511.00418, 2015.
- [13] P. J. Eicker, M. M. Siddiqui, and J. Mielke, Paul W., "A matrix occupancy problem," *The Annals of Mathematical Statistics*, vol. 43, no. 3, pp. 988–996, 1972.