

# Finite-Length Analysis of Frameless ALOHA

Francisco Lázaro\*, Čedomir Stefanović†

\*Institute of Communications and Navigation of DLR (German Aerospace Center),  
Wessling, Germany. Email: Francisco.LazaroBlasco@dlr.de

†Department of Electronic Systems, Aalborg University  
Aalborg, Denmark. Email: cs@es.aau.dk

**Abstract**—In this paper we present an exact finite-length analysis of frameless ALOHA that is obtained through a dynamical programming approach. Monte Carlo simulations are performed in order to verify the analysis. Two examples are provided that illustrate how the analysis can be used to optimize the parameters of frameless ALOHA. To the best of the knowledge of the authors, this is the first contribution dealing with an exact finite-length characterization of a protocol from the coded slotted ALOHA family of protocols.

## I. INTRODUCTION

Typical networking scenarios commonly involve uncertainty in terms of the time-instants when the activation of the communicating devices will happen, i.e., when the need for communication will arise. In scenarios in which multiple devices share a common medium and an access point (AP), this type of uncertainty is typically resolved via means of a random access protocol. Slotted ALOHA [1] is an example of such protocol, in which devices, upon activation, randomly and independently choose time slots in which they attempt connecting (i.e., contend for the access) to the AP by transmitting their packets. A collision of two or more packets is typically considered destructive, i.e., all packets involved in a collision are lost. Hence, only slots that contain a single packet (singleton slots) are considered useful and the related packets successfully received. In the above model, known as collision channel model, the asymptotic maximal throughput of slotted ALOHA, defined as the probability of successfully receiving a packet in a slot, is only  $1/e$ , implying that most of the slots are wasted.

The introduction of successive interference cancellation in the above framework significantly changed the perspective on the capabilities of random access protocols [2]. Namely, assume that users are sending multiple replicas of the same packet when contending, embedding in each replica information that allows to determine the position of all other replicas of the same packet. A packet that occurs in a singleton slot is successfully received, enabling the AP to identify the slots in which the other replicas occurred and to remove the replicas from the stored waveform using an interference cancellation

This work has been accepted for publication at the 11th International ITG Conference on Systems, Communications and Coding, SCC 2017.

©2016 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting /republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works

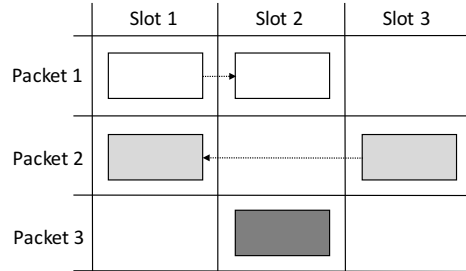


Fig. 1. Example of SIC-enabled slotted ALOHA. Packet 2 is received in singleton slot 3, enabling its recovery and the cancellation of its replica from slot 1. In turn, slot 1 becomes singleton and packet 1 is recovered from it, enabling its cancellation from slot 2. Slot 2 becomes singleton and packet 3 is recovered from it.

algorithm, see Fig. 1. Subsequently, some of the collision slots become singletons, promoting the recovery of new packets and the removal of their replicas. This process is analogous to the iterative belief-propagation decoding of erasure-correcting codes, enabling the use of theory and tools of codes-on-graphs to design and analyze slotted ALOHA-based protocols [3]. In fact, for the collision channel model the asymptotic throughput can be pushed to the ultimate limit of 1 packet per slot [4]. The price to pay is that the AP has to buffer the received signal, and it has to employ more complex signal processing, required for the interference cancellation.

The results presented in [3] inspired a strand of works that applied various concepts from codes-on-graphs to design SIC-enabled slotted ALOHA schemes [5]–[8], which are usually referred to using the umbrella term of coded slotted ALOHA. In this paper we focus on frameless ALOHA [7], [9], which exploits ideas originating from the rateless coding framework [10]. In particular, frameless ALOHA is characterized by (i) a contention period that consists of a number of slots that is not defined a priori, but terminated when the number of resolved<sup>1</sup> users and/or instantaneous throughput reach certain thresholds and (ii) a slot access probability with which a user decides on a slot basis whether to transmit or not its packet replica. An asymptotic optimization of the slot access probability that maximizes the expected throughput was performed in [7]. A joint assessment of the optimal slot access probability and the contention termination criteria in non-asymptotic, i.e., finite-length scenarios were assessed by means of simulations in [9]. Thus, so far the finite-length performance of frameless

<sup>1</sup>Under user resolution we assume recovery/decoding of user packet.

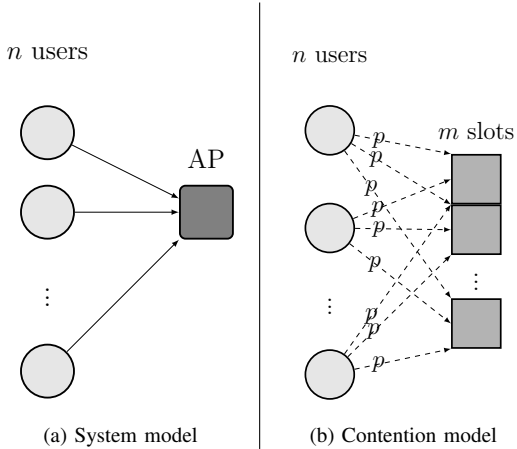


Fig. 2. System and contention models assumed in this work.

ALOHA, like for other SIC-enabled slotted ALOHA protocols, could only be established by means simulations or approximate methods that are usually only accurate in the error floor region, see [11], for example.

In this paper we build up on the approach for finite-length analysis of rateless codes presented in [12], [13], applying it to the context of frameless ALOHA. Specifically, we present an exact finite-length analysis of frameless ALOHA for the collision channel with successive interference cancellation at the receiver. This analysis provides the average throughput and the packet error rate, and, to the best of the knowledge of the authors, is the first exact finite-length analysis of a random access protocol belonging to the coded slotted ALOHA family of protocols. Furthermore, we illustrate by means of examples how the proposed analysis can be used to optimize the parameters of frameless ALOHA.

The rest of the paper is organized as follows. In Section II we describe the system model assumed in this paper. In Section III we present a finite length analysis of frameless ALOHA. Section IV presents two examples in which the presented analysis is used to optimize the parameters of frameless ALOHA in the finite-length regime. Finally, Section V presents the conclusions.

## II. SYSTEM MODEL

We adopt a simple model of traffic arrivals, focusing on the analysis of the performance of the proposed contention mechanism from the system point of view. Specifically, we consider a single instance of batch arrival of  $n$  users, which contend for the access to a single access point, Fig. 2(a).

Contention is performed during a period that consists of  $m$  slots, where  $m$  is not defined a-priori but determined on the fly. The users are slot and contention period synchronous, and they arrive prior to the start of the contention period. A user contends by transmitting replicas of the same packet; for each slot of the contention period a user decides with slot access probability  $p$  whether to transmit a replica or not, independently of any other slot and of any other user, see Fig 2(b).

For the sake of simplicity, we initially assume that  $p$  is uniform over users and slots and that

$$p = \frac{\beta}{n}$$

where  $\beta$  is a suitably chosen constant. Thus, the contention on a user basis is modelled as a Bernoulli trial with  $m$  repetitions with probability  $p$ . Hence, the number of replicas that a user sends follows also a binomial distribution with parameters  $m$  and  $p$ . Denoting as  $\Omega = (\Omega_0, \Omega_1, \Omega_2, \Omega_3, \dots, \Omega_n)$  the slot degree distribution, where  $\Omega_i$  corresponds to the probability of a slot having degree  $i$ , we have

$$\Omega_i = \binom{n}{i} p^i (1-p)^{n-i}.$$

In practice,  $\Omega$  can be tightly approximated by a Poisson distribution whenever  $n$  takes moderate or large values, i.e.,

$$\Omega_i \approx \frac{(np)^i}{i!} e^{-np} = \frac{\beta^i}{i!} e^{-\beta}.$$

We assume a widely adopted collision channel model, so that slots containing only one transmission (singleton slots) can be decoded with probability 1 and slots containing more than one transmission are undecodable with probability 1. Following [3] we describe the iterative successive cancellation process at the receiver using a bipartite graph, assuming  $m$  is fixed. By  $\mathbf{v} = (v_1, v_2, \dots, v_n)$  we denote the  $n$  users, and by  $\mathbf{y} = (y_1, y_2, \dots, y_m)$  the  $m$  slots. We use the notation  $\text{deg}(y)$  to refer to the (original) degree of a slot, that is, the number of users that transmitted in the slot. Furthermore, during the decoding (i.e., SIC) process we introduce the term reduced degree to refer to the number of unresolved user packets that are still present in the slot. Thus, the reduced degree of a slot will be equal or less than its (original) degree. The following definitions are used in the finite-length analysis.

**Definition 1 (Ripple).** We define the ripple as the set of singleton slots (reduced degree 1) and we denote it by  $\mathcal{R}$ .

The cardinality of the ripple is denoted by  $r$  and its associated random variable as  $R$ .

**Definition 2 (Cloud).** We define the cloud as the set of slots with reduced degree  $d \geq 2$  and we denote it by  $\mathcal{C}$ .

The cardinality of the cloud is denoted by  $c$  and the corresponding random variable as  $C$ .

For illustration, in Fig. 3 we provide an example of bipartite graph for  $n = 4$  users and  $m = 4$  slots. We can observe how slots  $y_1$  and  $y_4$  belong to the ripple and slots  $y_2$  and  $y_3$  belong to the cloud.

In the next sections, in the ripple and cloud we will add a temporal dimension through the subscript  $u$  that corresponds to the number of unresolved users. Initially, all  $n$  users are unresolved, hence  $u = n$ . At each step, if the ripple is not empty exactly 1 user gets resolved, and thus the subscript decreases by 1. After  $n$  decoding steps, all users are resolved and decoding ends, i.e.,  $u = 0$ . If at any of the  $n$  decoding steps the ripple is empty, decoding fails.

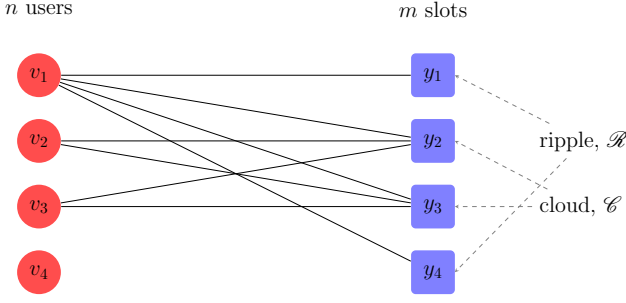


Fig. 3. Bipartite graph representation of slotted ALOHA.

### III. FINITE-LENGTH ANALYSIS

In this section we follow the approach in [12], [13] introduced for LT codes and model the iterative successive cancellation process of frameless ALOHA by means of a finite state machine with state

$$\mathbf{S}_u := (\mathbf{C}_u, \mathbf{R}_u)$$

i.e., the state comprises the cardinalities of the cloud and the ripple at the decoding step in which  $u$  users are unresolved. The following theorem establishes a recursion that can be used to determine the decoder state distribution.

**Theorem 1.** *Given that the decoder is at state  $\mathbf{S}_u = (\mathbf{c}_u, \mathbf{r}_u)$ , when  $u$  users are unresolved and with  $\mathbf{r}_u > 0$ , the probability of the decoder being at state  $\Pr\{\mathbf{S}_{u-1} = (\mathbf{c}_{u-1}, \mathbf{r}_{u-1})\}$  when  $u-1$  users are unresolved is given by*

$$\Pr\{\mathbf{S}_{u-1} = (\mathbf{c}_u - \mathbf{b}_u, \mathbf{r}_u - \mathbf{a}_u + \mathbf{b}_u) | \mathbf{S}_u = (\mathbf{c}_u, \mathbf{r}_u)\} = \binom{\mathbf{c}_u}{\mathbf{b}_u} q_u^{\mathbf{b}_u} (1 - q_u)^{\mathbf{c}_u - \mathbf{b}_u} \binom{\mathbf{r}_u - 1}{\mathbf{a}_u - 1} \times \left(\frac{1}{u}\right)^{\mathbf{a}_u - 1} \left(1 - \frac{1}{u}\right)^{\mathbf{r}_u - \mathbf{a}_u}$$

for  $0 \leq \mathbf{b}_u \leq \mathbf{c}_u$ ,  $\mathbf{b}_u - \mathbf{a}_u \leq \mathbf{r}_u$  and  $\mathbf{a}_u \geq 1$ , and with

$$q_u = \frac{\sum_{d=2}^{n-u-2} \Omega_d d(d-1) \frac{1}{n} \frac{u-1}{n-1} \binom{n-u}{d-2}}{n-u-1} \quad (1)$$

$$1 - \sum_{d=1} \Omega_d u \binom{n-u}{d} - \sum_{d=0} \Omega_d \binom{n-u}{d}$$

*Proof.* The proof reduces to analyzing the variation of the cloud and ripple sizes in the transition from  $u$  to  $u-1$  unresolved users. Since we assume  $\mathbf{r}_u > 0$ , in the transition from  $u$  to  $u-1$  unresolved users exactly 1 user is resolved. All the edges coming out from the resolved user are erased from the decoding graph. As a consequence some slots might leave the cloud and enter the ripple if their reduced degree becomes one, and other slots will leave the ripple if their reduced degree decreases from 1 to 0.

Let us first focus of the number of slots leaving  $\mathcal{C}_u$  and entering  $\mathcal{R}_{u-1}$  in the transition, denoted by  $\mathbf{b}_u$  and with associated random variable given by  $\mathbf{B}_u$ . Due to the nature of frameless ALOHA, in the decoding graph every slot chooses

its neighbor users uniformly at random and without replacement. Thus, random variable  $\mathbf{B}_u$  is binomially distributed with parameters  $\mathbf{c}_u$  and  $q_u$ , being  $q_u$  the probability of a generic slot  $y$  leaving  $\mathcal{C}_u$  to enter  $\mathcal{R}_{u-1}$ ,

$$q_u := \Pr\{y \in \mathcal{R}_{u-1} | y \in \mathcal{C}_u\} = \frac{\Pr\{y \in \mathcal{R}_{u-1}, y \in \mathcal{C}_u\}}{\Pr\{y \in \mathcal{C}_u\}}. \quad (2)$$

We shall first focus on the enumerator of (2) and we shall condition it to the slot having degree  $d$ ,  $\Pr\{y \in \mathcal{R}_{u-1}, y \in \mathcal{C}_u | \deg(y) = d\}$ . This corresponds to the probability that one of the  $d$  edges of slot  $y$  is connected to the user being resolved at the transition, one edge is connected to one of the  $u-1$  unresolved users after the transition and the remaining  $d-2$  edges are connected to the  $n-u$  unresolved users before the transition. In other words, slot  $y$  must have *reduced degree 2 before* the transition and *reduced degree 1 after* the transition. It is easy to see how this probability corresponds to

$$\Pr\{y \in \mathcal{R}_{u-1}, y \in \mathcal{C}_u | \deg(y) = d\} = \frac{d}{n} (d-1) \frac{u-1}{n-1} \frac{\binom{n-u}{d-2}}{\binom{n-2}{d-2}} \quad (3)$$

for  $d \geq 2$ . In the complementary case,  $d < 2$ , it is obvious that the slot cannot enter the ripple. Thus, we have

$$\Pr\{y \in \mathcal{R}_{u-1}, y \in \mathcal{C}_u | \deg(y) = d\} = 0$$

for  $d < 2$ .

Let us now concentrate on the denominator of (2), that corresponds to the probability that a slot  $y$  is in the cloud when  $u$  users are still unresolved. This is equivalent to the probability of slot  $y$  not being in the ripple or having reduced degree zero (all edges connected to resolved users). Hence, we have

$$\Pr\{y \in \mathcal{C}_u\} = 1 - \sum_{d=1}^n \Omega_d u \frac{\binom{n-u}{d-1}}{\binom{n}{d}} - \sum_{d=0}^n \Omega_d \frac{\binom{n-u}{d}}{\binom{n}{d}} \quad (4)$$

where the first summation on the right hand side corresponds to the probability of a slot being the ripple, and the second summation corresponds to the probability of a slot having reduced degree zero. Inserting (3) and (4) in (2), the expression of  $q_u$  in (1) is obtained, and the variation of size of the cloud, i.e., random variable  $\mathbf{B}_u$ , is determined.

We focus next on the variation of size of the ripple in the transition from  $u$  to  $u-1$  resolved users. In the transition some slots enter the ripple ( $\mathbf{b}_u$  slots) but there are also slots leaving the ripple. Let us denote by  $\mathbf{a}_u$  the number of slots leaving the ripple in the transition from  $u$  to  $u-1$  unresolved users, and let us refer to the associated random variable as  $\mathbf{A}_u$ . Assuming that the ripple is not empty<sup>2</sup>, the decoder will select at uniformly random one slot from the ripple, that we denote as  $y$ . The only neighbour of  $y$ ,  $c$  will get resolved. All slots in the ripple that are connected to  $c$  leave the ripple in the transition. Hence, we have that slot  $y$  leaves the ripple. Additionally, the remaining  $\mathbf{r}_u - 1$  slots in the ripple will leave

<sup>2</sup>If the ripple is empty,  $\mathbf{r}_u = 0$ , no slots can leave the ripple. Moreover, decoding stops, so there is no transition.

the ripple with probability  $1/u$ , which is the probability that they have  $c$  as neighbour. Thus, the probability mass function of  $A_u$  is given by

$$\Pr\{A_u = a_u | R_u = r_u\} = \binom{r_u - 1}{a_u - 1} \left(\frac{1}{u}\right)^{a_u - 1} \left(1 - \frac{1}{u}\right)^{r_u - a_u}.$$

The proof is completed simply by observing that by definition,

$$r_{u-1} = r_u - a_u + b_u$$

and

$$c_{u-1} = c_u - b_u.$$

□

The initial state of the decoder corresponds to a multinomial distribution with  $m$  experiments (slots) and three possible outcomes for each experiment, the slot being in the cloud, the ripple or having degree 0, with respective probabilities  $(1 - \Omega_1 - \Omega_0)$ ,  $\Omega_1$  and  $\Omega_0$ . Thus, we have

$$\Pr\{S_n = (c_n, r_n)\} = \frac{m!}{c_n! r_n! (m - c_n - r_n)!} (1 - \Omega_1 - \Omega_0)^{c_n} \Omega_1^{r_n} \Omega_0^{m - c_n - r_n} \quad (5)$$

for all non-negative  $c_n, r_n$  such that  $c_n + r_n \leq m$ .

The decoder state probabilities can be determined by initializing the finite state machine according to (5) and applying recursively Theorem 1. However, rather than in state probabilities, in random access one is interested in the Packet Error Rate (PER), i.e., the probability that a user is not resolved when the decoding process ends, denoted as  $P_e$ . By observing that the decoding process ends at stage  $u$  whenever  $r_u = 0$ , leaving  $u$  users unresolved and  $n - u$  resolved users, we have

$$P_e = \sum_{u=1}^n \sum_{c_u} \frac{u}{n} \Pr\{S_u = (c_u, 0)\}.$$

A complementary figure of merit is the throughput, denoted by  $T$ , which is the number of resolved users normalized by the number of slots. The expression of  $T$  corresponds to

$$T = \frac{n(1 - P_e)}{m} = \frac{1 - P_e}{m/n}.$$

Fig. 4 and Fig. 5 show, respectively, the throughput  $T$ , and packet error rate  $P_e$  as a function of  $m/n$  for  $\beta = 2.5$  and  $n = 100$ . The figures show the outcome of the finite state machine analysis as well as the outcome of simulations. The simulation results were obtained by simulating 10000 contention periods. It can be observed how the match is exact down to the simulation error.

#### IV. OPTIMIZATION

As shown in [7], the throughput  $T$  of frameless ALOHA typically shows a behavior like the one in Fig. 4, where three distinct phases can be recognised. Initially,  $T$  increases slowly with  $m/n$ . When  $m/n$  becomes close to 1, the successive interference cancellation process kicks in and  $T$  increases sharply till a maximum value is achieved. In the third phase,

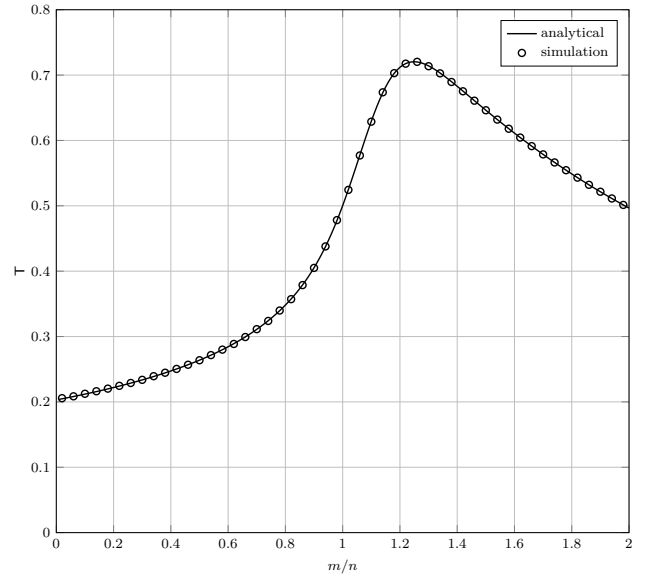


Fig. 4. Throughput  $T$  as a function of  $m/n$  for  $\beta = 2.5$ , for  $n = 100$ . The solid line represents the outcome of the finite state machine analysis and the markers are the outcome of Monte Carlo simulations.

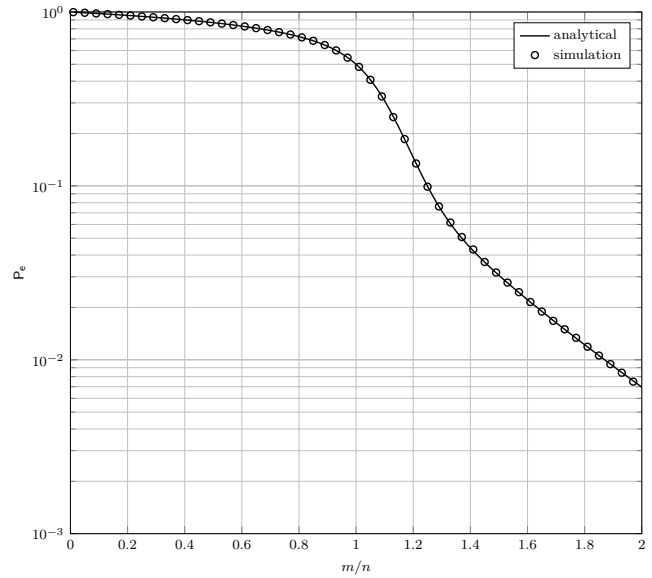


Fig. 5. Packet error rate  $P_e$  as a function of  $m/n$  for  $\beta = 2.5$ , for  $n = 100$ . The solid line represents the outcome of the finite state machine analysis and the markers are the outcome of Monte Carlo simulations.

the throughput  $T$  starts decreasing. This decrease is caused by the fact that  $P_e$  gets close to 0, see Fig. 5. Thus, when  $m/n$  increases the throughput  $T$  decreases approximately as  $1/(m/n)$ . This last phase coincides with the error floor of the packet error rate  $P_e$ , as depicted in Fig. 5.

In some applications it is desirable to maximize the (overall) throughput  $T$  in order to use the channel as efficiently as possible. However, in other settings it might be more important to minimize the packet error rate  $P_e$  for the given ratio of  $m/n$ . In the following we provide two examples that illustrate how the analysis presented in Section III can be used to customise the frameless ALOHA protocol.

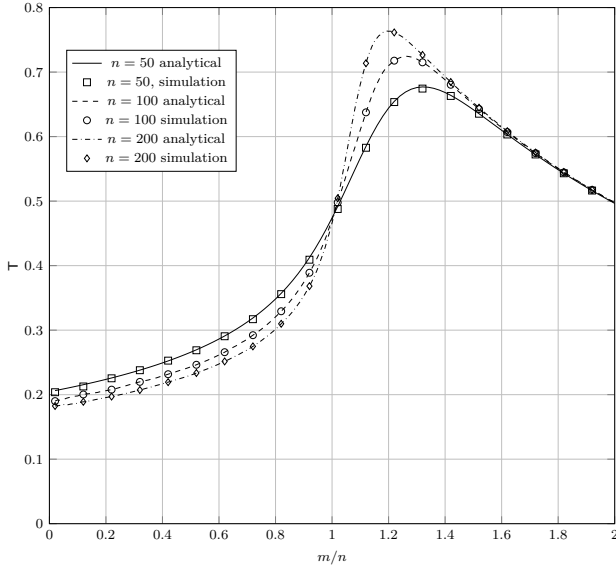


Fig. 6. Throughput  $T$  as a function of  $m/n$  for  $n = 50, 100$  and  $200$  and  $\beta = \beta_{\max}$ . The lines represent the outcome of the finite-length analysis and the markers the result of Monte Carlo simulations.

TABLE I  
OPTIMAL PARAMETERS FOR FRAMELESS ALOHA

$n$	50	100	200
$\beta_{\max}$	2.47	2.62	2.71
$T_{\max}(\beta_{\max})$	0.67	0.72	0.76
$m_{\max}$	66	126	240

### A. Peak Throughput Maximization

In this section we use the finite-length analysis to find the value of  $\beta$  that maximizes the peak throughput  $T$ , which we refer to as  $\beta_{\max}$ . Formally, given  $n$  and treating the throughput as a bivariate function of  $\beta$  and  $m/n$ , i.e.,  $T(\beta, m/n)$ , we define  $\beta_{\max}$  as:

$$\beta_{\max} = \operatorname{argmax}_{\beta} T_{\max}(\beta)$$

where

$$T_{\max}(\beta) = \max_{m/n} T(\beta, m/n).$$

The optimization was carried out for  $n = 50, 100$  and  $200$ . The result of the optimization can be found in Table I, where  $m_{\max}$  is the number of slots that maximizes the throughput for  $\beta = \beta_{\max}$ . These results are inline with the results obtained by means of simulation in [7]; in particular, the trend is that  $\beta_{\max}$  increases with  $n$ , where the optimal  $\beta_{\max} \approx 3.1$  when  $n \rightarrow \infty$  [7].

For illustration, we also present the evolution of the throughput  $T$  obtained for  $\beta = \beta_{\max}$ , and  $n = 50, 100$  and  $200$  in Fig. 6, where the corresponding loci of the maxima are given in Table I. The figure shows the outcome of the finite-length analysis and also the result of Monte Carlo simulations. Concretely, for each value of  $n$  and  $m$  10000 contention periods were simulated. The related evolution of the PER is shown in Fig. 7, we can observe how when  $\beta_{\max}$  is used  $P_e$  presents a rather high error floor, implying that maximization

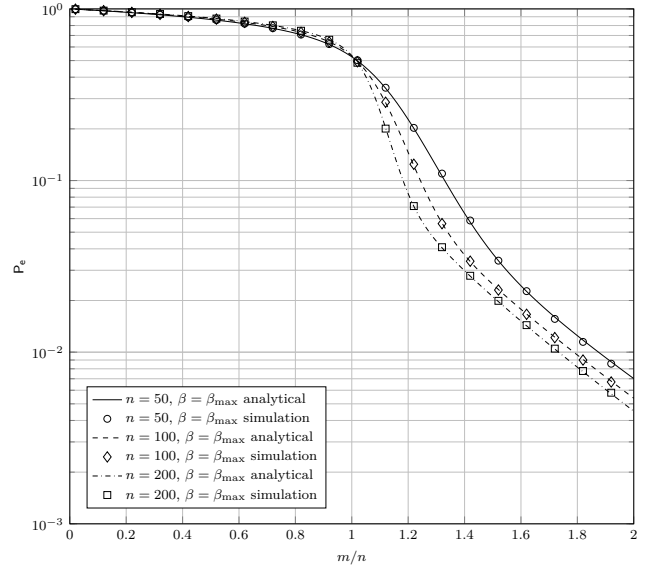


Fig. 7. Throughput  $T$  as a function of  $m/n$  for  $n = 50, 100$  and  $200$  and  $\beta = \beta_{\max}$ . The lines represent the outcome of the finite-length analysis and the markers the result of Monte Carlo simulations.

of the peak throughput does not guarantee a favorable PER performance. Specifically, a lower bound on PER is given by

$$P_e \geq \binom{m}{n} \left(1 - \frac{\beta}{n}\right)^m \quad (6)$$

where the expression on the right-hand side refers to the probability that a user does not transmit at all in any of the slots. For large  $m$  one can approximate the lower bound in (6) as

$$P_e \gtrsim e^{-\beta \frac{m}{n}}$$

Thus, if we fix  $m/n$ , the error floor can be lowered by increasing  $\beta$ . In the following section, we show how this error floor can be lowered substantially by increasing  $\beta$  after the peak throughput has been reached.

### B. Pushing Down the Error Floor

In this section we consider frameless ALOHA schemes where the parameter  $\beta$  varies with the slot number. Concretely, we consider

$$\beta = \begin{cases} \beta_1, & \text{if } m \leq m^* \\ \beta_2, & \text{if } m > m^* \end{cases}.$$

Thus, the slot access probability also depends on the slot number

$$p = \begin{cases} p_1, & \text{if } m \leq m^* \\ p_2, & \text{if } m > m^* \end{cases}$$

with  $p_1 = \beta_1/n$  and  $p_2 = \beta_2/n$ . As a consequence, the slot degree distribution will also vary with the slot number. For  $m \leq m^*$  we have

$$\Omega_i = \binom{n}{i} p_1^i (1 - p_1)^{n-i}$$

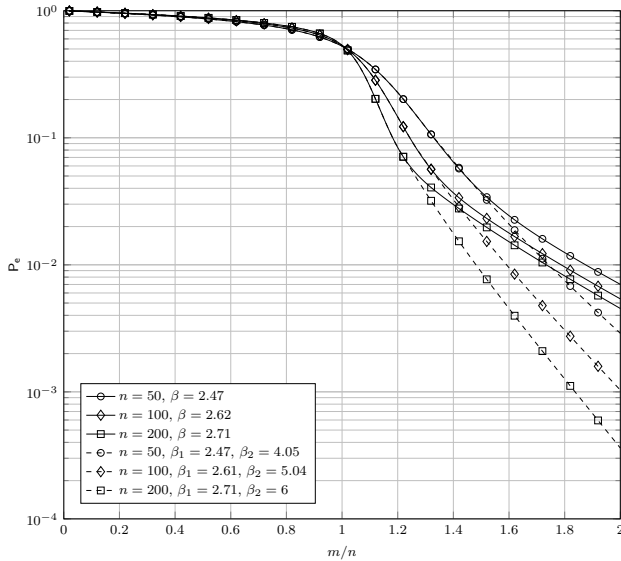


Fig. 8. Throughput  $\Gamma$  as a function of  $m/n$  for  $n = 50, 100$  and  $200$ . The solid lines represent the result for  $\beta = 2.5$ , the dashed lines represent the result for frameless ALOHA schemes with an initial phase with  $\beta_1 = \beta_{\max}$  and a second phase with  $\beta_2$  optimized to minimize  $P_e$  at  $m = 2 \cdot n$ .

whereas for  $m > m^*$  we have

$$\Omega_i = \frac{m^*}{m} \binom{n}{i} p_1^i (1 - p_1)^{n-i} + \frac{m - m^*}{m} \binom{n}{i} p_2^i (1 - p_2)^{n-i}.$$

In order to illustrate the usefulness of the finite-length analysis derived in Section III, we shall use the analysis to minimize the PER,  $P_e$ . We will consider frameless ALOHA schemes in which in  $\beta_1 = \beta_{\max}$  and  $m^* = m_{\max}$ . The parameter  $\beta_2$  is then chosen as the one that minimizes the packet error rate  $P_e$  at  $m/n = 2$ . The optimization was carried out for  $n = 50, 100$  and  $200$ . The parameters obtained through the optimization can be found in Table II. It can be observed that  $\beta_2$  grows approximately logarithmically with  $m$ . This result is inline with known results from LT code literature, where, in order for LT decoding to be successful with high probability, the average output degree (the equivalent of  $\beta$  for LT codes) needs to grow as  $\mathcal{O}(\log(k))$ , where  $k$  is the number of input symbols [10].

Fig. 8 shows the PER for frameless ALOHA schemes optimized for peak throughput (see previous Section) and schemes with two phases, where in the second phase  $\beta_2$  is chosen to minimize  $P_e$ . It can be observed how the PER becomes substantially lower in this second phase. For clarity of presentation, Fig. 8 shows only the PER values obtained using the finite-length analysis. Nevertheless, Monte Carlo simulations were carried out in order to verify the results and the match is also exact down to simulation error as in the rest of the examples in this paper.

The scenario chosen for this optimization, trying to maximize the peak throughput in the first phase and then in the second phase pushing down the error floor, might seem somewhat arbitrary. However, it illustrates the fact that the proposed analysis can be used in many ways to tailor frameless ALOHA to specific needs. For example one could define 3 or more phases or even change  $\beta$ , and hence  $p$ , slot by slot.

TABLE II  
OPTIMAL PARAMETERS FOR TWO-STAGE FRAMELESS ALOHA

$n$	50	100	200
$\beta_1$	2.47	2.62	2.71
$\beta_2$	4.05	5.04	6
$m^*$	66	126	240

## V. CONCLUSIONS

In this paper we have presented an exact finite-length analysis of frameless ALOHA based on dynamical programming approach. The analysis builds up on results from finite-length analysis of rateless codes. In contrast to most of the works in literature, the analysis applies not only in the error floor region but also in the waterfall region. Simulations have been performed to verify the analysis. Furthermore, two examples have been presented that illustrate how the proposed analysis can be used to optimize the parameters of frameless ALOHA.

## ACKNOWLEDGEMENT

The research presented in the paper was supported in part by the Danish Council for Independent Research, Grant No. DFF-4005-00281 “Evolving wireless cellular systems for smart grid communications”.

## REFERENCES

- [1] L. G. Roberts, “Aloha packet system with and without slots and capture,” *SIGCOMM Comput. Commun. Rev.*, vol. 5, no. 2, pp. 28–42, Apr. 1975.
- [2] E. Cassini, R. D. Gaudenzi, and O. del Rio Herrero, “Contention Resolution Diversity Slotted ALOHA (CRDSA): An Enhanced Random Access Scheme for Satellite Access Packet Networks,” *IEEE Trans. Wireless Commun.*, vol. 6, no. 4, pp. 1408–1419, Apr. 2007.
- [3] G. Liva, “Graph-Based Analysis and Optimization of Contention Resolution Diversity Slotted ALOHA,” *IEEE Trans. Commun.*, vol. 59, no. 2, pp. 477–487, Feb. 2011.
- [4] E. Paolini, G. Liva, and M. Chiani, “Coded Slotted ALOHA: A Graph-Based Method for Uncoordinated Multiple Access,” *IEEE Trans. Inf. Theory*, vol. 61, no. 12, pp. 6815 – 6832, Dec. 2015.
- [5] —, “High Throughput Random Access via Codes on Graphs: Coded Slotted ALOHA,” in *Proc. of IEEE ICC 2011*, Kyoto, Japan, Jun. 2011.
- [6] G. Liva, E. Paolini, M. Lentmaier, and M. Chiani, “Spatially-Coupled Random Access on Graphs,” in *Proc. of IEEE ISIT 2012*, Boston, MA, USA, Jul. 2012.
- [7] C. Stefanovic, P. Popovski, and D. Vukobratovic, “Frameless ALOHA protocol for Wireless Networks,” *IEEE Comm. Lett.*, vol. 16, no. 12, pp. 2087–2090, Dec. 2012.
- [8] E. Paolini, C. Stefanovic, G. Liva, and P. Popovski, “Coded Random Access: How Coding Theory Helps to Build Random Access Protocols,” *IEEE Commun. Mag.*, vol. 53, no. 6, pp. 144–150, Jun. 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, Nov. 2013.
- [10] M. Luby, “LT codes,” in *Proc. 43rd Annual IEEE Symp. on Foundations of Computer Science*, Vancouver, Canada, Nov. 2002, pp. 271–282.
- [11] M. Ivanov, F. Brännström, A. Graell i Amat, and P. Popovski, “Error floor analysis of coded slotted ALOHA over packet erasure channels,” *IEEE Commun. Lett.*, vol. 19, no. 3, pp. 419–422, Mar. 2015.
- [12] R. Karp, M. Luby, and A. Shokrollahi, “Finite length analysis of LT codes,” in *Proc. 2004 IEEE International Symp. on Inf. Theory*, Chicago, Illinois, US, Jun. 2004.
- [13] F. Lázaro, G. Liva, and G. Bauch, “Inactivation decoding analysis for LT codes,” in *Proc. 52nd Annu. Allerton Conf. on Commun., Control, and Computing*, Monticello, Illinois, USA, Oct. 2015.