Reliability-Latency Performance of Frameless ALOHA with and without Feedback

Francisco Lázaro*, Čedomir Stefanović†, Petar Popovski†

*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,petarp}@es.aau.dk

Abstract

This paper presents a finite length analysis of multi-slot type frameless ALOHA based on a dynamic programming approach. The analysis is exact, but its evaluation is only feasible for short contention periods due to the computational complexity. The analysis is then extended to derive continuous approximations of its key parameters, which, apart from providing an insight into the decoding process, make it possible to estimate the packet error rate with very low computational complexity. Finally, a feedback scheme is presented in which the slot access scheme is dynamically adapted according to the approximate analysis in order to minimize the packet error rate. The results indicate that the introduction of feedback can substantially improve the performance of frameless ALOHA.

I. INTRODUCTION

Random access protocols find applications in scenarios where there is a number of users sharing a common transmission medium and there exists uncertainty regarding which users are active. They can be used both in the initial phase of grant-based access, where the active users contend with metadata in order to reserve the uplink resources for the subsequent data transmissions, or in grant-free access, where the active users contend directly with packets

This work has been presented in part at ITG SCC, Hamburg, Germany, February 2017 [1], and IEEE Globecom, Singapore, December 2017 [2].
containing data. The former approach forms the basis of mobile cellular access, e.g. [3]. However, the latter approach has been gaining research momentum recently, e.g. [4], due its lower signaling overhead which makes it suitable for systems, such as Internet of things (IoT), where the amount of the exchanged data is small but the number of contending users may be large [5].

The first and still widely used random access protocols are ALOHA and slotted ALOHA [6]. Assuming a collision channel model, these two protocols offer low peak throughput ($1/2e$ and $1/e$ respectively) and high packet error rate (PER) even for low channel load. However, it has been shown how the introduction of successive interference cancellation (SIC) at the receiver can lead to higher performance by leveraging on coding-theoretic tools [7], [8]. In practice, this implies storing and processing the receiver waveform and leads to higher receiver complexity. The results presented in [8] inspired a strand of works that applied various concepts from codes-on-graphs to design SIC-enabled slotted ALOHA schemes [9]–[14], which are usually referred to using the umbrella term of coded slotted ALOHA.

Frameless ALOHA [11], [15] is a version of SIC-enabled slotted ALOHA that exploits ideas originating from the rateless-coding framework [16]. In particular, it is characterized by a contention period that consists of a number of slots that is not defined a priori, and by a slot access probability for the users to independently transmit their packets in a slot. An asymptotic optimization of the slot access probability that maximizes the expected throughput was performed in [11], while a similar optimization in a cooperative, multi-base station scenario was recently considered in [17]. 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 [15]. In particular, the finite-length performance of SIC-enabled slotted ALOHA protocols, has been so far established only by means simulations or approximate methods, see [14], [18]–[21], for example.

The focus of this paper on finite-length analysis of frameless ALOHA and of its reliability-latency performance. Specifically, we formulate the finite-length analysis of multi-slot type frameless ALOHA, i.e., frameless ALOHA with multiple classes of slots, where each class is characterized by a different slot access probability. In the next step, we statistically characterize the reliability of the scheme for a predefined latency target. We then use the flexibility that frameless ALOHA offers with respect to the frame-based schemes and propose a scheme in which the actual reliability performance is progressively assessed at predefined checkpoints during the contention, and feedback is used to drive the contention process towards the maximum possible
reliability (i.e., decoding of as many users as possible) given the latency target. In particular, the feedback sent by the access point to the users consists of an update on the slot access probability that should be subsequently used. This feedback induces a multi-slot type contention, since a new slot type is created whenever the slot access probability is changed. In other words, the proposed scheme adapts to the reliability-latency performance observed at multiple points, instead of focusing on just a single point (i.e., at the very end of the latency budget) like in the frame-based schemes, which allows for a finer control of the contention process. We show that the proposed scheme achieves a superior performance.

This paper builds on preliminary results from [1] and [2]. In [1] an exact finite-length analysis of frameless ALOHA was presented for the case in which all slots are statistically identical. This analysis was then extended in [2] to multiple slot types, sketching how the analysis could be changed to accommodate this extension without actually providing a full proof. The contributions in this paper are the following. We present in detail an exact, self-contained finite-length analysis of multi-slot type frameless ALOHA. Due to computational complexity, this exact analysis is only feasible for moderate contention. This has been the motivation to extend the analysis towards deriving continuous approximations of the expected ripple size (the number of slots containing only one transmission) and its standard deviation. Based on these approximations, we propose a method to estimate the PER with very low complexity. Finally, we exploit this estimation of the packet error rate to propose a feedback based scheme in which the slot access probability of frameless ALOHA is adapted dynamically.

The remainder of the paper is organized as follows. Section II provides a brief overview of frameless ALOHA and describes the system model. Section III presents the finite-length analysis, which can be used to obtain the exact probability mass function of the number of unresolved users for for a given duration of the contention period. In Section IV, the state generating function of the frameless ALOHA decoding process is derived. In Section V, we derive continuous approximations for the expected sizes of the ripple and the different clouds, as well as an approximation of the standard deviation of the ripple. It is also shown how these approximations can be used to estimate the packet error rate. Section VI shows how it is possible to largely improve the performance of frameless ALOHA by introducing feedback and applying the analysis derived in this paper. Finally, Section VII concludes the paper.
Fig. 1: An example of contention in Frameless ALOHA. All three users randomly and independently decide on a slot basis whether to transmit or not. Slot 1 and slot 2 are collision slots; the colliding transmissions can not be decoded and the AP stores the slots (i.e., the signals observed in them) for later use. Slot 4 is a singleton slot and the AP decodes a replica of the packet of user 2 from it. The AP also learns that a replica of packet of user 2 occurred in slot 1, and removes (cancels) it from the stored signal. Slot 1 now becomes singleton and a replica of the packet of user 1 becomes decoded. In the same manner, the successive process of replica removal and decoding of a new packet replica occurs in slot 2. As all three users have become resolved, the AP terminates the contention period after slot 4, and starts a new one.

II. BACKGROUND AND SYSTEM MODEL

A. Background: Frameless ALOHA

Frameless ALOHA [11] can be regarded as a variant of slotted ALOHA with SIC that is inspired by rateless codes [16]. The time in frameless ALOHA is divided into equal-length slots and slots are organized into contention periods, whose length is a-priori not known. In order to transmit a packet, users must first wait until a new contention period starts. Next, in each slot of the contention period, every contending user transmits a replica of its packet with a predefined slot-access probability. This happens independently from the transmission on other slots and independently of the actions of any other contending user. Furthermore, the assumption is made that each packet replica contains information about the slots in which the other replicas of the same packet are placed (e.g., this information could be represented with a seed of a random number generator included in the packet header). The receiver (access point) is required to store the waveform of the whole contention period and processes slots sequentially. In particular,
the receiver attempts to decode transmissions in the last received slot. Whenever a packet is successfully decoded, the contribution of the decoded packet and of all its replicas are removed from the stored waveform. This might reduce the interference suffered by some still undecoded packets, enabling the receiver to decode them. This process is repeated until no more packets can be decoded. At this stage, the access point can decide whether to terminate the contention period or to allow for more slots in the current contention period. The decision is made according to a predefined criterion, e.g., whether the target throughput has been reached and/or a predefined fraction of users have been resolved [15]. The start or termination of a contention period can be signaled to users by means of a beacon signal transmitted by the access point. An example of contention period in frameless ALOHA is depicted in Fig. 1.

B. System Model

We denote by $n$ the number of users contending for access to a single access point. The duration of the contention period in slots is denoted by $m$; note that $m$ is not a-priori fixed. Furthermore, we shall assume that $k$ different slot types exists. In particular, we assume that, out of the total $m$ slots, exactly $m_1, m_2, \ldots, m_k$ are of type $1, 2, \ldots, k$. Slots of type $h$, are characterized by a slot access probability $p_h$, given by $p_h = \frac{\beta_h}{n}$, which is equal for all users. It is easy to verify that $\beta_h$ is the mean number of users that transmitted in a slot of type $h$, and, thus, the mean number of transmissions contained in the slot.
A collision channel model will be assumed. Hence, singleton slots, i.e., slots containing a single transmission, are decodable with probability 1, and collision slots, i.e., slots containing two or more transmissions, are not decodable with probability 1. Perfect interference cancellation will be assumed, i.e., the removal of replicas from the slots leaves no residual transmission power.\(^1\)

In order to model the successive interference cancellation process at the receiver, we introduce the following definitions:

**Definition 1** (Initial slot degree). *The initial slot degree is the number of transmissions originally occurring in the slot.*

**Definition 2** (Reduced slot degree). *The reduced slot degree is the current number of transmissions in the slot, over the iterations of the reception algorithm.*

**Definition 3** (Ripple). *The ripple is the set of slots of reduced degree 1, and it is denoted it by \(R\).*

The cardinality of the ripple, \(|R|\) is denoted by \(r\) and its associated random variable as \(R\).

**Definition 4** (\(h\)-th cloud). *The \(h\)-th cloud, \(C_h\), is the set of slots of type \(h\) with reduced degree \(d > 1\).*

The cardinality of the \(h\)-th cloud, \(|C_h|\), is denoted by \(c_h\) and the corresponding random variable as \(C_h\).

Upon reception, the reduced degree of a slot is equal to its initial degree. During the decoding process, the reduced degree of a slot is decreased by 1 whenever one of the undecoded packets present the slot is decoded and its interference cancelled. Note that the interference cancellation process may cause some of the slots of the \(h\)-th cloud to leave the \(h\)-th cloud and enter the ripple, if the slot becomes of reduced degree 1. Furthermore, the interference cancellation process also causes slots in the ripple to become of reduced degree 0 and leave the ripple. These slots are of no further use in the decoding process, thus, we do not consider them explicitly in our analysis. This process is depicted in Fig. 2.

Let us consider the example in Fig. 1, and assume that there are two slot types. Further, let us assume that slot 1 and 3 are of type 1, whereas slot 2 and 4 are of type 2. Initially, slot 1 is

\(^1\)This assumption is reasonable for practical interference cancellation methods and moderate to high signal-to-noise ratios [8].
in the 1-st cloud \( C_1 \), slot 2 is in the 2-nd cloud \( C_2 \), and slot 4 in the ripple, \( R \). After the first round of decoding and replica removal, slot 4 leaves the ripple, while slot 1 leaves the 1-st cloud and enters the ripple. Hence, the 1-st cloud becomes empty. In the next round of decoding and replica removal, slot 1 leaves the ripple, while the slot 2 leaves the 2-nd cloud and enters the ripple. Thus, the 2-nd cloud becomes empty. The reception algorithm stops when slot 2 leaves the ripple.

Finally, we denote the slot degree distribution of slot type \( h \) by \( \Omega_h = \{ \Omega_{h,1}, \Omega_{h,2}, \ldots, \Omega_{h,n} \} \), \( h = 1, 2, \ldots, k \), where \( \Omega_{h,j} \) is the probability that a slot of type \( h \) has initial degree \( j \). It is straightforward to verify that \( \Omega_{h,j}, j = 1, 2, \ldots, n \), is given by

\[
\Omega_{h,j} = \binom{n}{j} (p_h)^j (1-p_h)^{n-j} = \binom{n}{j} \left( \frac{\beta_h}{n} \right)^j \left( 1 - \frac{\beta_h}{n} \right)^{n-j} .
\]

Hence, the probability that a slot of type \( h \) initially contains no transmission at all is \( \Omega_{h,0} \), the probability that it initially belongs to the ripple is \( \Omega_{h,1} \), and the probability that it initially belongs to the \( h \)-th cloud is \( 1 - \Omega_{h,0} - \Omega_{h,1} \).

### III. Finite-Length Analysis

For the sake of convenience and without loss of accuracy, we shall assume that the receiver works iteratively. If the ripple is empty, the receiver simply stops. Otherwise, it carries out the following steps:

- Selects at random one of the slots in the ripple;
- Resolves the user that was active in that slot (i.e., decodes its packet);
- Cancels the interference contributed by the resolved user from all other slots in which its packet replicas were transmitted. This may cause some slots to leave the cloud and enter the ripple. Furthermore, some slots from the ripple may become degree zero and leave the ripple. These last slots correspond to slots in the ripple in which the resolved user was active.

Thus, in each iteration, reception algorithm either fails, or exactly one user gets resolved. These assumptions are made to ease the analysis and have no impact on the performance.

Following the approach in [1], [22], [23], the iterative reception of frameless ALOHA with \( k \) different slot types is represented as a finite state machine with state

\[
S_u := (C_{1,u}, C_{2,u}, \ldots, C_{k,u}, R_u)
\]
i.e., the state comprises the cardinalities from the first to \( k \)-th cloud and the ripple at the reception step in which \( u \) users are unresolved. Each iteration of the reception algorithm corresponds to a state transition. The following proposition establishes a recursion that can be used to determine the state distribution.

**Theorem 1.** Given that its is state \( S_u = (c_{1,u}, c_{2,u}, \ldots, c_{k,u}, r_u) \), where \( u \) users are unresolved, and \( r_u > 0 \) (i.e., the ripple is not empty), the probability of the receiver transitioning to state \( \Pr \{ S_{u-1} = s_{u-1} \} \), where \( u-1 \) users are unresolved, is given by

\[
\Pr \{ S_{u-1} = (s_u + w)|S_u = s_u \} = \left( \frac{r_u - 1}{a_u - 1} \right) \left( \frac{1}{u} \right)^{a_u-1} \times \left( 1 - \frac{1}{u} \right)^{r_u - a_u} \prod_{h=1}^{k} \left( \frac{c_{h,u}}{b_{h,u}} \right)^{q_{h,u}} \frac{b_{h,u}}{1 - q_{h,u}}^{c_{h,u} - b_{h,u}}
\]

with

\[
s_u = (c_{1,u}, c_{2,u}, \ldots, c_{k,u}, r_u)
\]

\[
w = (-b_{1,u}, -b_{2,u}, \ldots, -b_{k,u}, \sum_{h=1}^{k} b_{h,u} - a_u)
\]

and

\[
q_{h,u} = \frac{\sum_{d=h+1}^{n} \Omega_{h,d} d \left( \binom{u-1}{k} \binom{n-u}{d-k-1} \right)^{k}}{1 - \sum_{h=0}^{k} \sum_{d=h}^{n} \Omega_{h,d} \left( \binom{n}{d-h} \binom{u}{d} \right)^{k}} \quad \text{(2)}
\]

for \( 0 \leq b_{h,u} \leq c_{h,u}, 1 \leq a_u \leq r_u \).

**Proof:**

Let us start by remarking that, given the fact that every users decides whether to transmit or not in a slot independently from other users, all users and independent and statistically identical. Furthermore, all slots are mutually independent. Thus, if we look at the decoder when \( u \) users are unresolved, the set of unresolved users is obtained by selecting \( u \) users at random among the total of \( n \) users. In particular, the proof analyzes the variation of the clouds and ripple sizes in the transition from \( u \) to \( u - 1 \) unresolved users. Since we assume \( r_u > 0 \), 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 on the number of slots leaving $C_{h,u}$ and entering $R_{u-1}$ in the transition, denoted by $b_{h,u}$ and with associated random variable given by $B_{h,u}$. Due to the nature of frameless ALOHA, in the decoding graph the neighbor users of a slot are selected uniformly at random and without replacement\(^2\). Thus, random variable $B_{h,u}$ is binomially distributed with parameters $c_{h,u}$ and $q_{h,u}$, being $q_{h,u}$ the probability of a generic slot $y$ of type $h$ leaving $C_{h,u}$ to enter $R_{u-1}$,

\begin{equation}
q_{h,u} := \Pr\{y \in R_{u-1} \mid y \in C_{h,u}\} = \frac{\Pr\{y \in R_{u-1}, y \in C_{h,u}\}}{\Pr\{y \in C_{h,u}\}}. \tag{3}
\end{equation}

We shall first focus on the numerator of (3) and we shall condition it to the slot having degree $d$, $\Pr\{y \in R_{u-1}, y \in C_{h,u}\mid \deg(y) = d\}$. 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 $n-u$ resolved 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, conditioned to $\deg(y) = d$, corresponds to

\begin{equation}
\Pr\{y \in R_{u-1}, y \in C_{h,u}\mid \deg(y) = d\} = \frac{d}{n} (d-1) \frac{u-1}{n-1} \frac{\binom{n-u}{d-2}}{\binom{n-2}{d-2}} \tag{4}
\end{equation}

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

$$\Pr\{y \in R_{u-1}, y \in C_{h,u}\mid \deg(y) = d\} = 0$$

\(^2\)In fact, it is users who choose their neighbor slots uniformly at random and without replacement.
for $d < 2$.

Let us now concentrate on the denominator of (3), that corresponds to the probability that a slot $y$ is in the $h$-th 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

$$\Pr\{y \in \mathcal{C}_{h,u}\} = 1 - \sum_{d=1}^{n} \Omega_{h,d} u \left(\frac{n-u}{d}\right) - \sum_{d=0}^{n} \Omega_{h,d} \left(\frac{n-u}{d}\right)$$

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 (4) and (5) in (3), the expression of $q_{h,u}$ in (2) is obtained, and the variation of size of the cloud, i.e., random variable $B_u$, is determined.

We focus next on the variation of size of the ripple in the transition from $u$ to $u-1$ resolved users. In this transition, some slots enter the ripple (in total $\sum_{h} b_{h,u}$ slots), but there are also slots leaving the ripple. Let us denote by $a_u$ 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 $A_u$. Assuming that the ripple is not empty, the decoder will select uniformly at 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 $c$ (including $y$) leave the ripple in the transition. Additionally, the remaining $r_u - 1$ slots in the ripple will leave the ripple independently 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

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

The proof is completed by observing that from the definition of $a_u$ and $b_{h,u}$ it follows

$$r_{u-1} = r_u - a_u + \sum_{h=1}^{k} b_{h,u}$$

and

$$c_{h,u-1} = c_{h,u} - b_{h,u}.$$  

If the ripple is empty, $r_u = 0$, no slots can leave the ripple. Moreover, decoding stops, so there is no transition.
Recall that out of the \( m \) slots in the contention period, exactly \( m_1, m_2, \ldots, m_k \) belong to slot type \( 1, 2, \ldots, k \). We focus on slots of type \( h \), of which there are \( m_h \). The initial state distribution corresponds to a multinomial with \( m_h \) experiments (slots) and three possible outcomes for each experiment: the slot being in the cloud, the ripple or having degree 0, with respective probabilities \((1 - \Omega_{h,1} - \Omega_{h,0}), \Omega_{h,1}, \Omega_{h,0}\). Denoting by \( R_{h,n} \) the random variable associated to the number of slots of type \( h \) of reduced degree 1 when all \( n \) users are still undecoded, we have

\[
\Pr\{ (C_{h,n} = c_{h,n}, R_{h,n} = r_{h,n}) \} = \frac{m_h!}{c_{h,n}! r_{h,n}! (m_h - c_{h,n} - r_{h,n})!} \\
\times (1 - \Omega_{h,1} - \Omega_{h,0})^{c_{h,n}} \Omega_{h,1}^{r_{h,n}} \Omega_{h,0}^{m_h - c_{h,n} - r_{h,n}}
\]

for all non-negative \( c_{h,n}, r_{h,n} \) such that \( c_{h,n} + r_{h,n} \leq m_h \).

If we observe that, when all \( n \) users are still undecoded, the total number of degree one slots \( R_n \) is given by

\[
R_n = \sum_{h=1}^{k} R_{h,n}
\]

we can obtain from (6) the initial state distribution of the receiver.

By applying recursively Theorem 1 and initializing as described the finite state machine one obtains the state probabilities.

Let us denote by \( P_u \) the probability that exactly \( u \) users remain unresolved after a contention period of \( m \) slots. Obviously, the event that exactly \( u \) users remain unresolved corresponds to the event that the user resolution ends at stage \( u \). The probability of this event is simply the probability that the ripple is empty when \( u \) users are still unresolved. Formally we have

\[
P_u = \Pr\{ R_u = 0 \} = \sum_{c_{1,u}} \sum_{c_{2,u}} \cdots \sum_{c_{k,u}} \Pr\{ S_u = (c_{1,u}, c_{2,u}, \ldots, c_{k,u}, 0) \}
\]

where the summations is taken over all possible values of \( c_{h,u}, h = 1, \ldots, k \).

Thus, by applying Theorem 1 and then using (7), one obtains the probability mass function of the number of unresolved users for the given number of users \( n \) and duration of the contention period in \( m \).

As an example, in Fig. 4 we show the pmf of the number of undecoded users \( u \), i.e., \( P_u \) for \( u = 1, \ldots, n \), when \( n = 50 \) and \( m = 60 \), for (i) one slot type with mean initial slot degree

\[\text{Note that } P_u \text{ implicitly depends on the initial state distribution that is obtained through (6), while (6) depends on the number of slots of a given type } m^{(h)}, h = 1, 2, \ldots, k, \text{ and thereby on the total number of slots } m.\]
Fig. 4: Examples of probability mass function of the number of undecoded users $u$ for $n = 50$, $m = 60$.

$\beta = 2.68$, and (ii) two slot types, with $m_1 = 50$ slots of the first type, $m_2 = 10$ slots of the second type, with mean initial degrees $\beta_1 = 3$ and $\beta_2 = 5$, respectively. The figure shows analytical results according to Theorem 1 and the outcome of Monte Carlo simulations. We see that the match is tight down to simulation error (100,000 contention periods were simulated). In this particular example, we can observe how $P_u$ has a bimodal distribution. Thus, there are two points in the decoding process in which the ripple has a higher probability of becoming empty.

The expected packet error rate $P$, i.e., the probability that a user is not resolved, can also be derived from Theorem 1. In particular, we have

$$P = \sum_{u=1}^{n} \frac{u}{n} P_u = \sum_{u=1}^{n} \sum_{c_1,u} \sum_{c_2,u} \cdots \sum_{c_k,u} \frac{u}{n} \Pr\{S_u = (c_{1,u}, c_{2,u}, \ldots, c_{c,u}, 0)\}.$$

For the example in Fig. 4, the expected packet error rates for the contention with one and two slot classes correspond to 0.264 and 0.555, respectively.

Hence, the expected (normalized) throughput is simply

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

\[5\]Recall that slot access probability of a type $h$ is $p_h = \beta_h/n$. 

IV. STATE GENERATING FUNCTIONS

Following the works in [22], [24], [25], which analyze the iterative LT decoding process, let us define the state generating function of the frameless ALOHA decoder as the probability generating function of the random variable associated with the state of the frameless ALOHA decoder,

\[ P_u(x_1, x_2, \ldots, x_k, y) := \sum_{c_1 \geq 0, c_2 \geq 0, \ldots, c_k \geq 0, r \geq 1} p_{c_1, c_2, \ldots, c_k, r, u} x_1^{c_1} x_2^{c_2} \ldots x_k^{c_k} y^{r-1}. \]

The following theorem establishes a recursion for the state generating function.

**Theorem 2.** Consider a contention period with \( m \) slots of \( k \) different types and \( n \) contending users, which in a slot of type \( h \) are active with probability \( p_h \). For \( u = n, n-1, \ldots, 1 \), we have

\[ P_u(x_1, x_2, \ldots, x_k, y) = \]

\[ \frac{1}{y} \left[ P_u \left( x_1(1 - q_{1,u}) + yq_{1,u}, x_2(1 - q_{2,u}) + yq_{2,u}, \ldots, x_k(1 - q_{k,u}) + yq_{k,u}, \frac{1}{u} + y \left( 1 - \frac{1}{u} \right) \right) \right. \]

\[ - P_u \left( x_1(1 - q_{1,u}), x_2(1 - q_{2,u}), \ldots, x_k(1 - q_{k,u}), \frac{1}{u} \right) \]

where \( q_{h,u} \) is given by

\[ q_{h,u} = \frac{\sum_{d=2}^{n-u+2} \Omega_{h,d} d(\frac{d-1}{d-2})^{\frac{u-1}{n-1}}^{\frac{n-u}{d-2}}}{1 - \sum_{d=1}^{n-u} \Omega_{h,d} u^{\frac{n-u}{d}} - \sum_{d=0}^{n-u} \Omega_{h,d}^{\frac{n-u}{d}}} \]

and initial condition given by

\[ P_n(x_1, x_2, \ldots, x_k, y) = \]

\[ \frac{1}{y} \prod_{h=1}^{k} ((1 - \Omega_{h,1} - \Omega_{h,0})x + \Omega_{h,1}y + \Omega_{h,0})^m - ((1 - \Omega_{h,1} - \Omega_{h,0})x + \Omega_{h,0})^m. \]

**Proof:**
We provide a proof only for the case of \( k = 2 \) slot types. The proof follows closely [24, Theorem 1]. By definition, the state generating function of the frameless ALOHA decoder when \( u - 1 \) users are undecoded can be written as

\[
P_{u-1}(x_1, x_2, y) = \sum_{c'_1 \geq 0} \sum_{c'_2 \geq 0} \sum_{r' \geq 1} p_{c'_1, c'_2, r', u-1} x_1^{c'_1} x_2^{c'_2} y^{r'-1}
\]

Our objective is expressing \( P_{u-1}(x_1, x_2, y) \) as a function of \( P_u(x_1, x_2, y) \). For this purpose, we must consider the contribution of all previous states \( s_u = (c_{1,u}, c_{2,u}, r_u) \) to the future state \( s_{u-1} = (c'_{1,u-1}, c'_{2,u-1}, r'_{u-1}) \). In the remainder of the proof, we shall drop the subscript \( u \) for the sake of readability. As done in the proof of Theorem 1, we denote by \( b_h \) the number of slots leaving \( C_h, u \) in the transition from \( u \) to \( u - 1 \) undecoded slots. Furthermore, we denote by \( a \) the number of slots leaving the ripple in the transition. Obviously, we have

\[
c_1 = c'_1 + b_1 \\
c_2 = c'_2 + b_2 \\
r = r' + a - b_1 - b_2.
\]

Thus, we can rewrite (10) as follows

\[
P_{u-1}(x_1, x_2, y) = \frac{1}{y} \sum_{c_1 \geq 0} \sum_{c_2 \geq 0} \sum_{r \geq 1} p_{c_1, c_2, r, u} x_1^{c_1-b_1} x_2^{c_2-b_2} y^{r-a+b_1+b_2} \\
\times \left( \frac{c_1}{b_1} \right) q_{1,u}^{b_1} (1 - q_{1,u})^{c_1-b_1} \left( \frac{c_2}{b_2} \right) q_{2,u}^{b_2} (1 - q_{2,u})^{c_2-b_2} \left( \frac{r - 1}{a} \right) \left( \frac{1}{u} \right)^{a-1} \left( 1 - \frac{1}{u} \right)^{r-a}.
\]

By grouping different terms together, we obtain

\[
P_{u-1}(x_1, x_2, y) = \frac{1}{y} \sum_{c_1 \geq 0} \sum_{c_2 \geq 0} \sum_{r \geq 1} p_{c_1, c_2, r, u} \left( \frac{c_1}{b_1} \right) (y q_{1,u})^{b_1} (x_1 (1 - q_{1,u}))^{c_1-b_1} \\
\times \left( \frac{c_2}{b_2} \right) (y q_{2,u})^{b_2} (x_2 (1 - q_{2,u}))^{c_2-b_2} \left( \frac{r - 1}{a} \right) \left( \frac{1}{u} \right)^{a-1} \left( y \right) \left( 1 - \frac{1}{u} \right)^{r-a}.
\]
We now take out the term $a = r$ from the summation. In case that $a = r$, $b_1$ and $b_2$ cannot be both zero at the same time. Thus, we have

$$P_{u-1}(x_1, x_2, y) = \frac{1}{y} \sum_{c_1 \geq 0, c_2 \geq 0, r \geq 1} \sum_{0 \leq b_1 \leq c_1, 0 \leq b_2 \leq c_2} \left( \begin{array}{c} c_1 \\ b_1 \\ \end{array} \right) (y q_{1,u})^{b_1} (x_1(1 - q_{1,u}))^{c_1-b_1} \times \left( \begin{array}{c} c_2 \\ b_2 \\ \end{array} \right) (y q_{2,u})^{b_2} (x_2(1 - q_{2,u}))^{c_2-b_2} \times \left( \begin{array}{c} r - 1 \\ a - 1 \\ \end{array} \right) \left( \begin{array}{c} 1 \\ u \\ \end{array} \right)^{a-1} \left( y \left( 1 - \frac{1}{u} \right) \right)^{r-a}$$

(11)

By using in (11) the results of the following two finite sums

$$\sum_{b_1 = 0}^{c_1} \left( \begin{array}{c} c_1 \\ b_1 \\ \end{array} \right) (y q_{1,u})^{b_1} (x_1(1 - q_{1,u}))^{c_1-b_1} = (y q_{1,u} + x_1(1 - q_{1,u}))^{c_1}$$

$$\sum_{a = 1}^{r-1} \left( \begin{array}{c} r - 1 \\ a - 1 \\ \end{array} \right) \left( \begin{array}{c} 1 \\ u \\ \end{array} \right)^{a-1} \left( y \left( 1 - \frac{1}{u} \right) \right)^{r-a} = \left( \frac{1}{u} + y \left( 1 - \frac{1}{u} \right) \right)^{r-1} - \left( \frac{1}{u} \right)^{r-1}$$

after some manipulation, we obtain

$$P_{u-1}(x_1, x_2, y) = (y q_{1,u} + x_1(1 - q_{1,u}))^{c_1} (y q_{2,u} + x_2(1 - q_{2,u}))^{c_2} \times \left( \frac{1}{u} + y \left( 1 - \frac{1}{u} \right) \right)^{r-1} - \left( \frac{1}{u} \right)^{r-1} (x_1(1 - q_{1,u}))^{c_1} (x_2(1 - q_{2,u}))^{c_2}$$

which yields (8).

In order to complete the proof, we only need to provide a proof for (9), which specifies the initial condition of the recursive expression given in (8). By definition, $p_{c_1, c_2, r, n}$ is simply the probability that before decoding starts ($u = n$) we have exactly $c_1$ slots in the first cloud, $c_2$ slots in the second cloud and $r$ slots in the ripple. Let us decompose $r$ as $r = r_1 + r_2$, where $r_1$ and $r_2$ represent, the number of slots of type 1 and type 2 in the ripple, respectively, with $r_1 \geq 0$ and $r_2 \geq 0$. We have

$$P_n(x_1, x_2, y) = \sum_{c_1, c_2, r} p_{c_1, c_2, r, n} x_1^{c_1} x_2^{c_2} y^{r-1} = \sum_{c_1, c_2, r_1, r_2} p_{c_1, c_2, r_1 + r_2, n} x_1^{c_1} x_2^{c_2} y^{r_1 + r_2 - 1}.$$
From (6) we have that

\[
p_{c_1, c_2, r_1 + r_2, n} = \prod_{h=1}^{2} \frac{m_h!}{c_h! \, r_h! \, (m_h - c_h - r_h)!} \left(1 - \Omega_h, 1 - \Omega_h, 0\right)^{c_h} \Omega_h, 1 \, r_n \, \Omega_h, 0 \, m_h - c_h - r_h.
\]

If we replace this expression into (6) and make use of the multinomial Theorem, one obtains (9). This completes the proof for \(k = 2\). The proof for generic \(k\) uses the same reasoning and follows exactly the same steps.

Theorem 8 can be used to derive the probability of decoding failure in a similar way as done for the LT decoder in [24]. In particular, we have that the probability that the decoder fails when exactly \(u\) users are undecoded is \(1 - P_u(1)\), where 1 is the all one vector. This corresponds to the probability that the ripple is empty. Hence, the probability that exactly \(u\) users remain unresolved after a contention period of \(m\) slots, \(P_u\), corresponds to:

\[P_u = 1 - P_u(1).
\]

V. APPROXIMATION USING DIFFERENTIAL EQUATIONS

While the analysis presented in Sections III and IV can provide the exact probability mass function of the number of unresolved users, its evaluation is computationally complex. In this section, we derive continuous approximations of the first and second moment of the ripple and clouds. These approximations can be indirectly used to obtain an assessment of the number of unresolved users. In particular, they can help us identify if the decoding process is likely to stop or continue until a large fraction of the users has been decoded. Furthermore, the approximations are very easy to evaluate, since they consist of polynomials and logarithms.

A. First Moment

Following [22], [25], we define the expected size of the clouds and ripple, respectively, as

\[C_h(u) := \sum_{\substack{c_1 \geq 0 \\cdots \\ c_k \geq 0 \\ r \geq 1}} c_h \, p_{c_1, c_2, \ldots, c_k, r, u} = \frac{\partial}{\partial x_h} P_u(x_1, x_2, \ldots, x_k, y)\big|_1\]

and

\[R(u) := \sum_{\substack{c_1 \geq 0 \\cdots \\ c_k \geq 0 \\ r \geq 1}} (r - 1) \, p_{c_1, c_2, \ldots, c_k, r, u} = \frac{\partial}{\partial y} P_u(x_1, x_2, \ldots, x_k, y)\big|_1\]
Thus, we can obtain a recursion for $C_h(u)$ if we differentiate both sides of the recursion in Theorem 8 with respect to $x_h$, and evaluate the expression at 1, which yields

$$C_h(u - 1) = (1 - q_{h,u})C_h(u) - (1 - q_{h,u})\frac{\partial}{\partial x_h}P_u \left(1 - q_{1,u}, \ldots, 1 - q_{k,u}, \frac{1}{u}\right)$$

In [26] it was shown that, under the assumption $r \geq 6$, the drift term (12) is $O(1/n^2)$, leading to

$$C_h(u - 1) = (1 - q_{h,u})C_h(u) + O(1/n^2).$$

In a similar way, the expression for $R(u)$ can be obtained differentiating both sides of the recursion in Theorem 8 with respect to $y$ and evaluating the expression at 1,

$$R(u - 1) = \left(1 - \frac{1}{u}\right)R(u) + \sum_{h=1}^{k} q_{h,u}C_h(u) - P_u(1)$$

+ $P_u(1 - q_{1,u}, 1 - q_{2,u}, \ldots, 1 - q_{k,u}, 1/u)$. We again make use of a result in [26], where it was shown that for $r \geq 5$ the residual term

$$-P_u(1) + P_u(1 - q_{1,u}, 1 - q_{2,u}, \ldots, 1 - q_{k,u}, 1/u)$$

can be approximated as $-1 + O(1/n^2)$.

Recall that in frameless ALOHA, due to the contention mechanism, in a slot of type $h$, users are active with a probability $p_h$. This induces a binomial slot degree distribution given by eq. (1). If for a fixed contention period length, we look at the contention graph of frameless ALOHA from the perspective of slots, this is equivalent to saying that slots choose their neighbours uniformly at random and without replacement (a user can not be active multiple times in a slot).

Following [22], [25] the frameless ALOHA decoder can be approximated by introducing the assumption that slots nodes choose their neighbors with replacement in the bipartite graph representation of the contention process, i.e., the same user node can be chosen several times in the same slot. If a slot node has an odd number of edges connected to a user node, the user node will be active in that slot. If the number of edges is even $\{0, 2, \ldots\}$, the user is not active in that slot. This approximation results in a slight decrease of the slot access probability $p_h$. Nevertheless, intuitively the approximation becomes tighter as the number of users $n$ increases, since it becomes less and less likely that a slot chooses several times the same user.

Under the replacement assumption, the expression of $q_{h,u}$ becomes:

$$q_{h,u} = \frac{1}{n} f_h \left(\frac{u}{n}\right) - \frac{1}{n^2} g_h \left(\frac{u}{n}\right) = \frac{1}{n} f_h \left(\frac{u}{n}\right) + O(1/n^2)$$
where
\[ f_h(x) = \frac{x \Omega''_h(1-x)}{1 - x \Omega'_h(1-x) - \Omega_h(1-x)}, \]
\[ g_h(x) = \frac{f_h(x)}{x}, \]
and \( \Omega_h(x) \) is the generator polynomial of the slot degree distribution of slots of type \( h \),
\[ \Omega_h(x) = \sum_d \Omega_{h,d} x^d. \]
From (13) and (15) we obtain the following difference equation for the \( h \)-th cloud
\[ C_h(u) - C_h(u-1) = f_h \left( \frac{u}{n} \right) C_h(u) + O(1/n^2) \tag{16} \]
Similarly, from (14) and (15) we obtain
\[ R(u) - R(u-1) = \frac{1}{u} R(u) - \sum_{h=1}^{k} f_h \left( \frac{u}{n} \right) C_h(u) + 1 + O(1/n^2) \tag{17} \]
Let us now define the expected normalized size of the clouds and ripple respectively as
\[ C_h(\xi) := \frac{C_h(u)}{m} \]
and
\[ R(\xi) := \frac{R(u)}{m}. \]
Making use of these definitions, and assuming \( m = (1 + \epsilon) n \), we can divide both sides of (16) by \( m \) to obtain
\[ C_h(\xi) - C_h(\xi - 1/n) = f_h(\xi) C_h(\xi) + O(1/n^3) \]
and the same can be done for (17) leading to
\[ R(\xi) - R(\xi - 1/n) = \frac{1}{\xi n} R(\xi) - \sum_{h=1}^{k} f_h(\xi) C_h(\xi) + \frac{1}{n(1+\epsilon)} + O(1/n^3). \]
As shown in [26], it is possible to approximate \( C_h(\xi) R(\xi) \) respectively by \( \hat{C}_h(\xi) \hat{R}(\xi) \), which are the solutions to the following differential equations
\[ \hat{C}_h'(x) = f_h(x) \hat{C}_h(x) \tag{18} \]
and
\[ \hat{R}'(x) = \frac{\hat{R}(x)}{x} - \sum_{h=1}^{k} f_h(x) \hat{C}_h(x) + \frac{1}{1 + \epsilon}. \tag{19} \]
These approximations are tight during almost all the decoding process except for the last few users that are decoded. In particular, it was shown in [26] that as long as \( u \) is a constant fraction of \( n \), we have

\[
C_h(u)/m = \hat{C}_h(u/k) + \mathcal{O}(1/m)
\]

and

\[
R(u)/m = \hat{R}(u/k) + \mathcal{O}(1/m).
\]

Thus, the approximations become tighter for increasing \( m \).

Finally, the expression for \( \hat{C}_h(x) \) and \( \hat{R}(x) \) are obtained by solving the differential equations (18) and (19), which gives rise to the following solutions [25]

\[
\hat{C}_h(x) = \hat{c}_h \left( 1 - x \Omega'_h(1-x) - \Omega_h(1-x) \right)
\]

\[
\hat{R}(x) = x \left( \sum_{h=1}^{k} \hat{c}_h \Omega'_h(1-x) + \frac{n}{m} \log(x) + \hat{r} \right)
\]

where the values of the parameters \( \hat{c}_h \) and \( \hat{r} \) are determined by the initial conditions to the differential equations. In particular, for the clouds we have

\[
C_h(n) = \sum_{c_1 \geq 0, c_2 \geq 0, r \geq 1} p_{c_1, c_2, \ldots, c_k, r} = m_h \left( 1 - \Omega_{h,0} - \Omega_{h,1} \right) \left( 1 - \prod_{l=1}^{k} (1 - \Omega_{l,1})^{m_h} \right).
\]

Hence, by imposing \( \hat{C}_h(x=1) = C_h(n)/m \), we obtain

\[
\hat{c}_h = \frac{m_h}{m} \left( 1 - \prod_{l=1}^{k} (1 - \Omega_{l,1})^{m_l} \right).
\]

For the ripple we have

\[
R(n) = \sum_{c_1 \geq 0, c_2 \geq 0, r \geq 1} (r-1) p_{c_1, c_2, \ldots, c_k, r} = \sum_{h=1}^{k} m_h \Omega_{h,1} - 1 + \prod_{h=1}^{k} (1 - \Omega_{h,1})^{m_h}
\]

Thus, imposing \( \hat{R}(x=1) = R(n)/m \) yields

\[
r^\ast = \sum_{h=1}^{k} \frac{m_h}{m} \Omega_{h,1} \prod_{h=1}^{k} (1 - \Omega_{h,1})^{m_h} - \frac{1}{m} \left( 1 - \prod_{h=1}^{k} (1 - \Omega_{h,1})^{m_h} \right).
\]
B. Second Moment

In the following, we shall focus on the variance of the ripple, \( \sigma_R(u) \). By definition, we have

\[
\sigma_R(u) = \sum_{c_1 \geq 0}^{c_k \geq 0} \sum_{r \geq 1} (r - 1)^2 p_{c_1, c_2, \ldots, c_k, r, u} - R(u)^2.
\]

Experimentally, we made the observation that the distribution of the ripple \( R_u \) is approximately binomial (see Fig. 6). Under the assumption that \( R_u \) is binomially distributed with parameters \( m \) and \( \rho = R(u)/m \), a continuous approximation of the variance of the normalized ripple \( \sigma_R^2(u)/m^2 \) is given by

\[
\hat{\sigma}_R^2(x) = \frac{\rho(1 - \rho)m}{m^2} = \frac{\rho(1 - \rho)}{m}.
\]

C. Numerical Results

Fig. 5 shows the expected normalized ripple and clouds sizes, as well as the normalized standard deviation of the ripple and their continuous approximation. The setting considered is \( n = 400 \) users, \( k = 2 \) slot types, \( m_1 = m_2 = 380 \) slots, \( \beta_1 = 2.5 \) and \( \beta_2 = 3 \). We can observe how the continuous approximation \( \hat{R}(x) \) is very close to the actual expected normalized ripple size \( R(u)/m \). There exists also a tight match between the normalized clouds \( C_1(u)/m \) and \( C_2(u)/m \) and their continuous approximations \( \hat{C}_1(c) \) and \( \hat{C}_2(x) \). We can also observe how the match between \( \sigma_R(u)/m \) and \( \hat{\sigma}_R(x) \) is tight.

D. Approximation to the PER

We shall now see how the continuous approximations derived in this section can be used in order to estimate the expected packet error rate, \( P \), i.e., the probability that a user is not resolved. In particular, we shall make use of the continuous approximations of the expected ripple \( \hat{R}(x) \) and of its standard deviation \( \hat{\sigma}_R(x) \).

Fig. 6 shows the distribution of the ripple for \( n = 400 \) users, \( m_1 = m_2 = 380 \), \( \beta_1 = 2.5 \) and \( \beta_2 = 3 \) at 4 different instants during the decoding process. In particular, the figure shows the probability mass function of the ripple cardinality when \( u \) users are undecoded conditioned to the decoding process not failing for \( u' > u \). That is,

\[
\frac{\Pr\{R_u = r_u\}}{1 - \sum_{u' = u+1}^m \Pr\{R_{u'}\}} = \frac{\Pr\{R_u = r_u\}}{\sum_{r_u = 0}^m \Pr\{R_u = r_u\}}.
\]
Fig. 5: Normalized expected ripple and clouds, as well as the normalized standard deviation of the ripple and their continuous approximations for $n = 400$ users, $k = 2$ slot types, $m_1 = m_2 = 380$ slots, $\beta_1 = 2.5$ and $\beta_2 = 3$.

Moreover, the figure also shows Gaussian curves with mean $m \hat{R}(x)$ and standard deviation $\sqrt{m \sigma_R(x)}$. We make several observations in relation to this figure. The first is that the ripple cardinality has a bell shaped distribution across the whole decoding process, which resembles a Gaussian distribution. The second is that a Gaussian curve with mean $m \hat{R}(x)$ and standard deviation $\sqrt{m \sigma_R(x)}$ is reasonably close to the actual distribution of the ripple. In particular, we can observe how the mean ripple size is tightly approximated by $m \hat{R}(x)$. Finally, we observe that the approximation of the standard deviation of the ripple size is tight at the beginning and the end of the decoding process, but it is not as tight in the middle of the decoding process. In particular, in this regime the estimate of the standard deviation of the ripple is greater than its actual value.

Based on this observation, we can approximate the probability that exactly $u$ users remain unresolved after the end of the contention period, $P_u$, as

$$
\hat{P}_u = Q \left( \frac{\hat{R}(u/m)}{\sigma_R(u/m)} \right) \left( 1 - \sum_{l=u+1}^{n} \hat{P}_l \right)
$$

for $u < n$ and

$$
\hat{P}_n = Q \left( \frac{\hat{R}(n/m)}{\sigma_R(n/m)} \right)
$$

(20)
Fig. 6: Probability mass function of the ripple cardinality for $n = 400$ users, $k = 2$ slot types, $m_1 = m_2 = 380$ slots, $\beta_1 = 2.5$ and $\beta_2 = 3$ at different decoding instants. The solid line shows a Gaussian distribution with mean $m \hat{R}(x)$ and standard deviation $\sqrt{m} \hat{\sigma}(x)$.

for $u = n$, where $Q$ is the tail distribution function of the standard normal distribution. The first term in (20) corresponds to the probability that a Gaussian distribution with mean $\hat{R}(u/m)$ and standard deviation $\hat{\sigma}(u/m)$ takes a negative value. Hence, this first term corresponds to the probability that our estimation of the ripple is negative. The second term in (20) implies we are
conditioning to the decoding process in the previous decoding stages \((n \text{ down to } u + 1)\) being successful. An alternative way of interpreting this is that, according to our definition of \(P_u\), we have
\[
\sum_{u=1}^{n} P_u = 1
\]
and by introducing the second term we are enforcing
\[
\sum_{u=1}^{n} \hat{P}_u = 1.
\]

Thus, we can approximate the expected packet error rate \(P\) as
\[
\hat{P} = \sum_{u=1}^{n} \frac{u}{n} \hat{P}_u.
\]

Fig. 7 show the packet error rate \(P\) as well as its approximation \(\hat{P}\) as a function of \(m/n\), for \(n = 400\) users, \(k = 2\) slot types, \(m_1 = m_2 = m/2\), \(\beta_1 = 2.5\) and \(\beta_2 = 3\). We can observe how the approximation of the packet error rate is tight. The fact that the estimation of the packet error rate is tight despite the fact that the standard deviation of the ripple is overestimated in the
middle of the decoding process can be attributed to the fact that the dominating error event is that decoding stops at the end of the decoding process (when almost all users have been recovered). As we saw in Fig. 6, the estimate of the standard deviation of the ripple in this regime (at the end of the decoding process) is tight.

VI. DYNAMIC FEEDBACK

In this section show how the approximation of the analysis presented in this paper can be used together with feedback in order to improve the performance of frameless ALOHA. We assume that a perfect and zero-delay feedback channel exists from the access point to the terminals. This channel is used to notify users the value of the slot access probability which has to be used in upcoming slots. We also assume that the contention period is divided into subperiods of duration $t$ slots. All slots belonging to the same subperiod are of the same type, i.e., they are characterized by the same slot access probability. At the end of the $k$-th subperiod (after $k \cdot t$ slots have elapsed), the receiver runs the decoding algorithm to decode as many users as possible (until the ripple is empty). Next, based on the decoder state $S_u$, and using as a target a contention period duration of $mT$ slots, the decoder computes the slot access probability to be used in the next subperiod, $p_{k+1}$, and signals its to the users.

Fig. 8 shows an example of frameless ALOHA with dynamic feedback with four users. In this example the contention subperiods have length $t = 3$ slots. In this example the slot access probability for the first contention subperiod, $p_1$, happens to be set too high. As a consequence, all the slots in the first contention subperiod turn out to be collided. After the first $t = 3$ slots...
the receiver notifies users to use $p_2$ in the next contention subperiod, which spans slots 4, 5 and 6. However, after the second contention subperiod still no users could be decoded. The receiver then notifies users the slot access probability to be used in the next contention subperiod, $p_3$. Finally, in the third contention subperiod a singleton slot is received, which allows the receiver to decode all users leveraging on SIC. At this stage the receiver notifies all users that a new contention period starts.

For our analysis we will make the assumption that $p_{k+1}$ is chosen in order to minimize the expected packet error rate $\hat{P}$ for a total contention duration $m_T$ and assuming that all future slots use slot access probability $p_{k+1}$. We assume that users receive $p_{k+1}$ through instantaneous and error free signaling, and from slot $k \cdot t + 1$ on, they use slot access probability $p_{k+1}$. Note that we assume that all users keep on transmitting, even those which have been decoded.

Let us consider the case in which, after $k \cdot t$ slots have elapsed, the decoder stops at state

$$S_u := (c_{1,u}, c_{2,u}, \ldots, c_{k,u}, r_u = 0).$$

The decoder now signals the terminals a new slot access probability $p_{k+1}$ to be used for the remaining slots until slot $m_T$. This contention can be modelled as a standard Frameless ALOHA contention with $u$ users, $m' = m_T - k \cdot t + \sum_{i=1}^{k} c_{i,u}$ slots, and slot degree distribution $\Omega_{l,j}$. Let us denote by $l$ the random variable associated to the number of users which are active in a slot and by $j$ its realization. Let us also denote by $J$ the random variable associated to the number of unresolved users which are active in a slot and by $j$ its realization. We have

$$\Omega_{l,j} = \Pr\{J = j\} = \sum_{i=j}^{u} \Pr\{l = i\} \Pr\{J = j|l = i\} = \sum_{i=j}^{u} \binom{n}{i} (p_l)^i (1 - p_l)^{n-i} \binom{u}{j} \binom{n}{j} (1 - \frac{u}{n})^{i-j}.$$ 

Furthermore, the initial state distribution of the decoding process is given by

$$\Pr\{S_u = c_{1,u}, c_{2,u}, \ldots, c_{k+1,u}, r_u\} = \frac{m_{k+1}!}{c_{k+1,u}! r_u! (m_{k+1} - c_{k+1,u} - r_u)!} \times (1 - \Omega_{k+1,1} - \Omega_{k+1,0})^{c_{k+1,u}} \Omega_{k+1,1}^{r_u} \Omega_{k+1,0}^{m_{k+1} - c_{k+1,u} - r_u}.$$  

$^6$We assume that the receiver does not signal which users have been decoded. However, replicas from a decoded user will be cancelled from the received waveform at the receiver. Thus, the performance obtained is equivalent to the case in which the receiver signals which users have been decoded, and they stop transmitting.
for $C_{1,u} = c_{1,u}$, $C_{2,u} = c_{2,u}$, $\ldots$, $C_{k,u} = c_{k,u}$, and

$$\Pr\{S_u = c_{1,u}, c_{2,u}, \ldots, c_{k+1,u}, r_u\} = 0$$

otherwise. Hence, we initialize the $C_{1,u}$, $C_{2,u}$, $\ldots$, $C_{k,u}$ according to the decoder state at the receiver (which is deterministically known), and $C_{k+1,u}$ and $B_u$ are initialized according to a multinomial distribution, similarly as in (6).

Using this initial conditions, and given $p_{k+1}$, it is possible to use Theorem 1 in order to obtain the (exact) expected packet error rate $P$. Hence, it is possible to run a search to find the value of $p_{k+1}$ which minimizes the packet error rate. However, running such a search is computationally complex. Instead, we propose minimizing the approximation of the packet error rate $\hat{P}$ derived in Section V-D. This approach is suboptimal given the fact that we are minimizing an approximation of $P$, but it makes the search for the optimal value of $p_{k+1}$ much faster.

Fig. 9 shows the packet error rate $P$ as a function of the duration of the contention period in slots $m$, for a system with $n = 50$. Two curves are shown. The first one shows the performance of a static setting with $\beta = 2.94$, which is the value that minimizes $P$ for $m = 100$. The second curve shows the performance of a dynamic frameless ALOHA scheme in which $\beta$ is changed dynamically every $t = 10$ slots in order to minimize $\hat{P}$ at $m_T = 100$. This second curve was obtained by means of simulations. In particular, for every value of $m$ simulations were run until 50 unsuccessful contention periods were collected (a contention period is successful only if all $n$ users are correctly decoded). We observe for $m = 100$ that the packet error rate of the dynamic scheme is more than 2 orders of magnitude lower for the static scheme.

VII. CONCLUSIONS

In this paper we have presented a finite length analysis of multi-slot type frameless ALOHA. The analysis is exact, but its evaluation is computationally expensive and only feasible for short contention period sizes. The analysis has been extended to derive continuous approximations of the expected ripple size and its standard deviation. The approximations have been proven to be tight and have also been compared with simulation results. It has also been shown how these approximations can be used to accurately estimate the packet error rate in frameless ALOHA, making it possible to analyze the performance of frameless ALOHA for large contention periods. Finally, it has been shown how the performance of frameless ALOHA can be substantially improved by introducing feedback and adapting the slot access probability dynamically using the approximate analysis.
Fig. 9: Packet error rate $P$ as a function of $m$ for $n = 50$ users. The solid line represents a static setting for $\beta = 2.94$. The dashed line represents a dynamic frameless ALOHA setting where $\beta$ is changed every $t = 10$ slots, and the target contention period duration is set to $m_T = 100$ slots.

APPENDIX

INITIAL CONDITIONS FOR DYNAMIC FEEDBACK

Let us first consider the clouds, $C_h(u)$. For $h \leq k$

$$C_h(u) = \sum_{c_1 \geq 0} \sum_{c_k \geq 0} c_h p_{c_1,c_2,\ldots,c_k} = c_{h,u} \left( 1 - (1 - \Omega_{k+1})^{m_{k+1}} \right)$$

thus, we have

$$\dot{c}_h = \frac{c_{h,u}}{m^r} \frac{m - (1 - \Omega_{k+1})^{m_{k+1}}}{1 - \Omega_{h,1} - \Omega_{h,0}}.$$  

For $h = k + 1$ we have

$$C_h(u) = m_h \left( 1 - \Omega_{h,0} - \Omega_{h,1} \right) \left( 1 - (1 - \Omega_{h,1})^{m_h} \right)$$

which yields

$$\dot{c}_h = \frac{m_h}{m} \left( 1 - (1 - \Omega_{h,1})^{m_h} \right).$$
For the ripple we have

\[
R(u) = \sum_{c_1 \geq 0} \cdots \sum_{c_k \geq 0} c_{h,u} + m_{k+1} \Omega_{k+1,1} - 1 + \left(1 - \Omega_{k+1,1}\right)^{m_{k+1}}
\]

hence, imposing \( \hat{R}(x = 1) = R(u)/m' \) yields

\[
\hat{\nu} = \sum_{h=1}^{k} \frac{c_{h,u}}{m'} \left(1 - \frac{\left(1 - (1 - \Omega_{k+1,1})^{m_{k+1}}\right)}{1 - \Omega_{h,1} - \Omega_{h,0}}\right) + \frac{m_{k+1}}{m'} \Omega_{k+1,1} - \frac{1}{m'} \left(1 - (1 - \Omega_{k+1,1})^{m_{k+1}}\right).
\]

REFERENCES

[1] F. Lázaro and C. Stefanovic, “Finite-Length Analysis of Frameless ALOHA,” in Proc. 11th International ITG Conf. on Sys., Commun. and Coding, Hamburg, Germany, Feb. 2017.

[2] C. Stefanovic, F. Lázaro, and P. Popovski, “Frameless ALOHA with reliability-latency guarantees,” in Proc. of IEEE Globecom 2017, Singapore, Dec. 2017, pp. 1–6.

[3] 3GPP, “3GPP TS 36.321 V15.4.0: Medium Access Control (MAC) protocol specification; (Release 14),” Tech. Rep., Jan. 2019.

[4] 3GPP, “Discussion on the reliability enhancement for grant-free transmission,” Tech. Rep., Aug. 2018.

[5] A. Laya, L. Alonso, and J. Alonso-Zarate, “Is the Random Access Channel of LTE and LTE-A Suitable for M2M Communications? A Survey of Alternatives,” IEEE Commun. Surv. Tut., vol. 16, no. 1, pp. 4–16, First Quarter 2014.

[6] 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.

[7] 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.

[8] 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.

[9] E. Paolini, G. Liva, and M. Chiani, “High Throughput Random Access via Codes on Graphs: Coded Slotted ALOHA,” in Proc. of IEEE ICC 2011, Kyoto, Japan, Jun. 2011.

[10] 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.

[11] C. Stefanovic, P. Popovski, and D. Vukobratovic, “Frameless ALOHA protocol for Wireless Networks,” IEEE Commun. Lett., vol. 16, no. 12, pp. 2087–2090, Dec. 2012.

[12] E. Paolini, C. Stefanovic, G. Liva, and P. Popovski, “Coded Random Access: Applying Codes on Graphs to Design Random Access Protocols,” IEEE Commun. Mag., vol. 53, no. 6, pp. 144–150, Jun. 2015.

[13] D. Jakovetić, D. Bajočić, D. Vukobratović, and V. Crnojević, “Cooperative slotted aloha for multi-base station systems,” IEEE Trans. Commun., vol. 63, no. 4, pp. 1443–1456, Apr. 2015.

[14] E. Sandgren, A. Graell i Amat, and F. Brännström, “On Frame Asynchronous Coded Slotted ALOHA: Asymptotic, Finite Length, and Delay Analysis,” IEEE Trans. Commun., vol. 65, no. 2, pp. 691–704, Feb. 2017.

[15] 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.
[16] M. Luby, “LT codes,” in Proc. 43rd Annual IEEE Symp. on Foundations of Computer Science, Vancouver, Canada, Nov. 2002, pp. 271–282.

[17] S. Ogata, K. Ishibashi, and G. T. F. de Abreu, “Optimized Frameless ALOHA for Cooperative Base Stations With Overlapped Coverage Areas,” IEEE Trans. Wireless Commun., vol. 17, no. 11, pp. 7486–7499, Nov. 2018.

[18] 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.

[19] A. G. i Amat and G. Liva, “Finite-length analysis of irregular repetition slotted ALOHA in the waterfall region,” IEEE Commun. Lett., vol. 22, no. 5, pp. 886–889, May 2018.

[20] M. Fereydounian, X. Chen, H. Hassani, and S. S. Bidokhti, “Non-asymptotic coded slotted ALOHA,” 2019, preprint, available online at https://www.seas.upenn.edu/˜mferey/ms.pdf.

[21] A. Mengali, R. De Gaudenzi, and Č. Stefanović, “On the Modeling and Performance Assessment of Random Access With SIC,” IEEE J. Select. Areas Commun., vol. 36, no. 2, pp. 292–303, Feb. 2018.

[22] R. Karp, M. Luby, and A. Shokrollahi, “Finite length analysis of LT codes,” in Proc. IEEE ISIT ’04, Chicago, IL, USA, Jun. 2004.

[23] F. Lázaro, G. Liva, and G. Bauch, “Inactivation decoding of LT and Raptor codes: Analysis and code design,” IEEE Trans. Commun., vol. 65, no. 10, pp. 4114–4127, Oct. 2017.

[24] A. Shokrollahi, “Theory and applications of Raptor codes,” Mathknow, vol. 3, pp. 59–89, 2009.

[25] G. Maatouk and A. Shokrollahi, “Analysis of the second moment of the LT decoder,” IEEE Trans. Inf. Theory, vol. 58, no. 5, May 2012.

[26] G. Maatouk, “Graph-based codes and generalized product constructions,” Ph.D. dissertation, École Polytechnique Fédérale de Lausanne, Switzerland, August 2013.