Aalborg Universitet

Reliability-Latency Performance of Frameless ALOHA with and without Feedback

Lazaro, Francisco; Stefanovic, Cedomir; Popovski, Petar

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 moderate number of users 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 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 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 by 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 its original version, frameless ALOHA 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]. In [18], an evolution of frameless ALOHA was proposed which makes use of feedback, ZigZag decoding, and a dynamical variation of the slot access probability to improve performance. 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].

The finite-length performance of frame-based coded slotted ALOHA schemes has been studied in [14], [19]–[22]. Here we remark that, due to the fact that in such schemes every user selects a random set of slots in a frame in which to transmit, interdependencies among slots are introduced; thus, the finite-length analytical model is in essence intractable, and one has to resort to approximations. The error floor of coded slotted ALOHA over the collision channel

in the finite length regime was studied in [19]. A frame asynchronous coded slotted ALOHA scheme was analyzed in [14], [20]. The scaling laws of coded slotted ALOHA were derived in [21]. In [22], the performance of several advanced random access protocols was compared over the collision channel and over the additive white Gaussian noise channel.

The focus of this paper on finite-length analysis of frameless ALOHA and of its reliability-latency performance with and without feedback. In particular, the considered feedback takes the form of an update of the slot access probability of frameless ALOHA, which is the default degree of freedom that can be used to optimize performance. This feedback induces multiple slot types in the contention, which substantially expands the design space, but also poses some challenges in terms of modeling and analytical performance optimization. The analysis presented in the paper characterizes statistically the reliability of muli-slot type frameless ALOHA for a predefined latency target, which in our context translates to a fixed number of slots in the contention period. The motivation of this approach stems from the fact that mission-critical services typically feature a given latency budget, which is also reflected in the proverbial reliability requirement of 10^{-5} under the latency of 1 ms in the context of 5G ultra-reliable low-latency service category [23].[1] The feedback scheme proposed in this paper exploits the flexibility that frameless ALOHA offers with respect to the frame-based schemes. Specifically, the considered scheme assesses the reliability at predefined intermediate checkpoint within a contention period and uses feedback to drive the contention process towards maximizing the number of decoded users, and thereby the reliability, 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. The proposed scheme adapts to the reliability-latency performance observed at multiple points, which allows for a finer control of the contention process. This paradigm change is a clear indicator of an enlarged design space compared to the frame-based designs that focus on just a single time instant, placed at the end of the latency budget. Finally, we show that the proposed feedback scheme outperforms frame-based schemes.

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

---

[1]We also remark that extending indefinitely the contention period in frameless ALOHA will always be beneficial in terms of reliability; this is a general feature of the rateless-coding framework. However, a practical design target always involves limited latency, which in this case maps to limited number of overhead slots.

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, the analysis is only feasible for moderate number of users. 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.[2] We investigate the performance of the dynamic version of the scheme and show that it achieves a rather favorable performance, which progressively improves with the number of feedback opportunities at the expense of an increased computational burden.

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 a given duration of the contention period. In Section IV, a low complexity approximate analysis of the frameless ALOHA decoder is presented, which is then used to estimate the packet error rate. Section V 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 VI concludes the paper.

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

---

[2]Note that, although the paper ultimately proposes the approximate method in order to deal with the computational complexity, the nature of the approximation is substantially different from the approximations employed in [14], [19]–[22]. Specifically, in our case, the exact model and analysis of the problem are at hand, as shown in the paper, and one can, in principle, exactly assess the quality of the approximation. In case of the methods employed in [14], [19]–[22], the analysis is from the very beginning approximate due to the interdependencies among slot selections by the users.

Slot 1   Slot 2   Slot 3   Slot 4   Slot 1

User 1

User 2

User 3

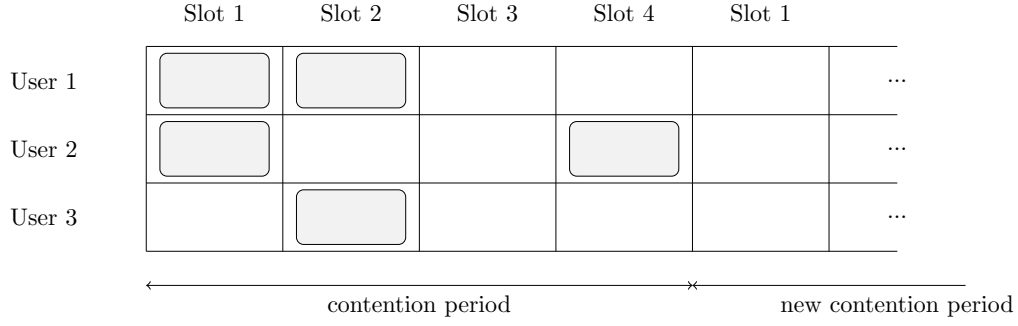← contention period →|← new contention period

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 access point (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.

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 in 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. This could be accomplished by, e.g., enriching the packet header with the seed of a random number generator and the transmit decisions are determined by the output of this generator. Hence, when a packet replica is decoded, it provides information about the timing of all other replicas. The access point (AP) is required to store the waveform of the whole contention period and processes slots sequentially, leveraging SIC to remove the interference caused by replicas of decoded packets. Since every packet contains a pointer to the location of all its replicas, when a packet is successfully decoded, the receiver can determine the location of all its replicas and remove them from the received waveform. This reduces the interference in those slots containing replicas of the decoded packets and it may enable the receiver to decode more packets within those slots. This process is repeated until no more packets can be decoded.
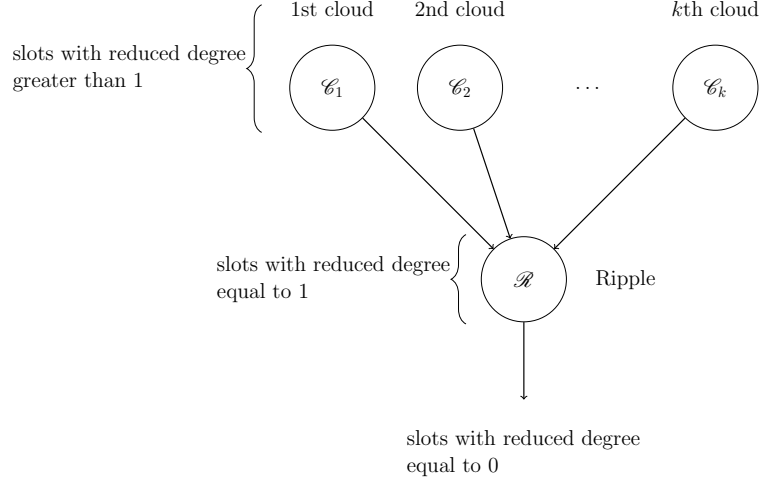
Fig. 2: The clouds and the ripple.

This SIC process is not exclusive to frameless ALOHA, but common to all SIC-enabled slotted ALOHA schemes. The peculiarity of frameless ALOHA is the fact that the contention period (also known as frame) duration need not be a priori defined. The AP can trigger the SIC process after the reception of every slot, and decide whether to end the contention period and start a new one, or alternatively to let the contention period continue. This 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 [15]. 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 $m_1, m_2, ...m_k$ are of type $1, 2, ..., k$. 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. Hence, in a slot of type h, every contending user will be active (send a packet) with probability p_h, independently from its transmission in others slots, and from the actions of the other users. It is easy to verify that β_h is the mean number of users that transmitted in a slot of type h, and, thus, equal to the expected 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.[3]

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 unresolved 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 by $\mathscr{R}$.* The cardinality of the ripple, $|\mathscr{R}|$ is denoted by $\mathtt{r}$ and its associated random variable as R.

Definition 4 (h-th cloud)**.** *The h-th cloud, $\mathscr{C}_h$, is the set of slots of type h with reduced degree $d > 1$.*

The cardinality of the h-th cloud, $|\mathscr{C}_h|$, is denoted by $\mathtt{c}_h$ and the corresponding random variable as C_h.

Upon reception, the reduced degree of a slot is equal to its initial degree, since all users that are active in the slot are unresolved. During the decoding process, the reduced degree of a slot is decreased by 1 whenever one of the unresolved users which are active in the slot is decoded and its interference is cancelled. Whenever the reduced degree of a slot in the h-th cloud becomes 1, the slot leaves the h-th cloud and enters the ripple. Note, however, that when the reduced degree of a slot decreases from, say, 3 to 2, then the slot remains in the h-th cloud. Similarly, when the only unresolved user in a slot belonging to the ripple is resolved, the reduced degree of the slot becomes 0 and the slot leaves the ripple. The slots of reduced degree 0 are not of further use in the analysis, and are thus not considered explicitly. The transition of slots between the clouds and ripple is depicted in Fig. 2.

In order to keep track of the temporal variation of the ripple and cloud sizes as the SIC process progresses, we introduce a subscript u, which corresponds to the number of unresolved users. Thus, initially we have $u = n$, and every time a user is resolved, u is decreased by 1 until we have decoded all users, when $u = 0$. Thus, the ripple when u users are unresolved is denoted

---

[3]This assumption is reasonable for practical interference cancellation methods and moderate to high signal-to-noise ratios [8].

by $\mathscr{R}_u$, its cardinality by r_u and the random variable associated to it by R_u. Similarly, when u users are unresolved, $\mathscr{C}_{h,u}$, $c_{h,u}$ and $C_{h,u}$ denote the h-th cloud, its cardinality and the associated random variable, respectively.

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 slots 1 and 2 have degree 2, hence they belong to the 1-st cloud $\mathscr{C}_1$ and 2-nd cloud $\mathscr{C}_2$, respectively. Similarly, slot 4 is initially in the ripple, $\mathscr{R}$, since it has a degree of one. In the first decoding step, the only active user in slot 4, user 2, gets resolved and the replica transmitted by user 2 in slot 1 is cancelled. As a consequence, 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 step, user 1 is resolved, slot 1 leaves the ripple, and slot 2 leaves the 2-nd cloud and enters the ripple. Thus, the 2-nd cloud becomes empty. Finally, user 3 is resolved, and slot 2 leaves the ripple. In this case, all 3 users could be resolved.

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

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

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 the slots in the ripple in which the resolved user was active.

Thus, in each iteration, the 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], [24], [25], the iterative reception of frameless ALOHA with k different slot types is represented as a finite state machine with state

$$\mathsf{S}_u := (\mathsf{C}_{1,u}, \mathsf{C}_{2,u}, \cdots, \mathsf{C}_{k,u}, \mathsf{R}_u)$$

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 the decoder is in state* $\mathsf{S}_u = (\mathsf{c}_{1,u}, \mathsf{c}_{2,u}, \cdots, \mathsf{c}_{k,u}, \mathsf{r}_u)$, *where* u *users are unresolved, and* $\mathsf{r}_u > 0$ *(i.e., the ripple is not empty), the probability of the receiver transitioning to state* $\Pr\{\mathsf{S}_{u-1} = \mathbf{s}_{u-1}\}$, *where* $u - 1$ *users are unresolved, is given by*

$$\Pr\{\mathsf{S}_{u-1} = (\mathbf{s}_u + \mathbf{w}) | \mathsf{S}_u = \mathbf{s}_u\} = \binom{\mathsf{r}_u - 1}{\mathsf{a}_u - 1} \left(\frac{1}{u}\right)^{\mathsf{a}_u - 1} \times$$

$$\left(1 - \frac{1}{u}\right)^{\mathsf{r}_u - \mathsf{a}_u} \prod_{h=1}^{h=k} \binom{\mathsf{c}_{h,u}}{\mathsf{b}_{h,u}} q_{h,u}{}^{\mathsf{b}_{h,u}} (1 - q_{h,u})^{\mathsf{c}_{h,u} - \mathsf{b}_{h,u}} \tag{2}$$

with

$$\mathbf{s}_u = (\mathsf{c}_{1,u}, \mathsf{c}_{2,u}, \cdots, \mathsf{c}_{k,u}, \mathsf{r}_u)$$

$$\mathbf{w} = \left(-\mathsf{b}_{1,u}, -\mathsf{b}_{2,u}, \cdots, -\mathsf{b}_{k,u}, \sum_{h=1}^{k} \mathsf{b}_{h,u} - \mathsf{a}_u\right)$$

and

$$q_{h,u} = \frac{\displaystyle\sum_{d=k+1}^{n} \Omega_{h,d} \frac{d}{n} \binom{d-1}{k} \frac{\binom{u-1}{k} \binom{n-u}{d-k-1}}{\binom{n-1}{k} \binom{n-k-1}{d-k-1}}}{1 - \displaystyle\sum_{h=0}^{k} \sum_{d=h}^{n} \Omega_{h,d} \frac{\binom{u}{h} \binom{n-u}{d-h}}{\binom{n}{d}}} \tag{3}$$

for $0 \le \mathsf{b}_{h,u} \le \mathsf{c}_{h,u},\ 1 \le \mathsf{a}_u \le \mathsf{r}_u$.

Proof: See Appendix A. ∎

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 $(1 - \Omega_{h,1} - \Omega_{h,0})$, $\Omega_{h,1}$ and $\Omega_{h,0}$. Denoting by $\mathrm{R}_{h,n}$ 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

$$\Pr\{(\mathrm{C}_{h,n} = \mathsf{c}_{h,n}, \mathrm{R}_{h,n} = \mathsf{r}_{h,n}\} = \frac{m_h!}{\mathsf{c}_{h,n}! \, \mathsf{r}_{h,n}! \, (m_h - \mathsf{c}_{h,n} - \mathsf{r}_{h,n})!} \tag{4}$$
$$\times \left(1 - \Omega_{h,1} - \Omega_{h,0}\right)^{\mathsf{c}_{h,n}} \Omega_{h,1}^{\mathsf{r}_n} \Omega_{h,0}^{m_h - \mathsf{c}_{h,n} - \mathsf{r}_{h,n}}$$

for all non-negative $\mathsf{c}_{h,n}, \mathsf{r}_{h,n}$ such that $\mathsf{c}_{h,n} + \mathsf{r}_{h,n} \leq m_h$.

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

$$\mathrm{R}_n = \sum_{h=1}^{k} \mathrm{R}_{h,n}$$

we can obtain from (4) 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

$$\mathsf{P}_u = \Pr\{\mathrm{R}_u = 0\} = \sum_{\mathsf{c}_{1,u}} \sum_{\mathsf{c}_{2,u}} \cdots \sum_{\mathsf{c}_{k,u}} \Pr\{\mathsf{S}_u = (\mathsf{c}_{1,u}, \mathsf{c}_{2,u}, \cdots, \mathsf{c}_{k,u}, 0)\} \tag{5}$$

where the summations is taken over all possible values of $\mathsf{c}_{h,u}$, $h = 1, ..., k$.

Thus, by applying Theorem 1 and then using (5), one obtains the probability mass function (PMF) of the number of unresolved users, given number of users n and the duration of the contention period[4] m.

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

$$\mathsf{P} = \sum_{u=1}^{n} \frac{u}{n} \mathsf{P}_u = \sum_{u=1}^{n} \sum_{\mathsf{c}_{1,u}} \sum_{\mathsf{c}_{2,u}} \cdots \sum_{\mathsf{c}_{k,u}} \frac{u}{n} \Pr\{\mathsf{S}_u = (\mathsf{c}_{1,u}, \mathsf{c}_{2,u}, \cdots, \mathsf{c}_{\mathsf{c},u}, 0)\}. \tag{6}$$

[4]Note that P_u implicitly depends on the initial state distribution that is obtained through (4), while (4) depends on the number of slots of a given type $m^{(h)}$, $h = 1, 2, ..., k$, and thereby on the total number of slots m.
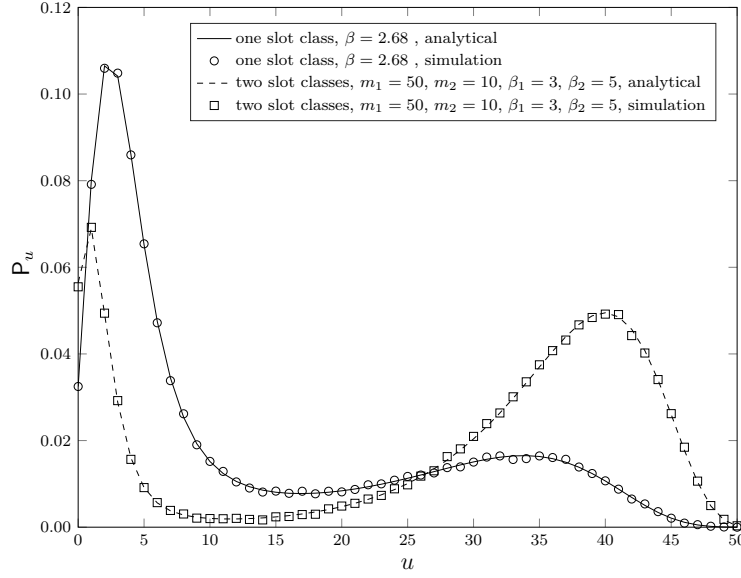
Fig. 3: Examples of probability mass function of the number of undecoded users u for $n = 50$, $m = 60$.

Hence, the expected throughput is simply the ratio of the expected number of resolved users over the number of slots in the contention period. i.e.,

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

As an example, in Fig. 3 we show the PMF of the number of undecoded users u, i.e., P_u for $u = 1, ..., n$, when $n = 50$ and $m = 60$, for frameless ALOHA with (i) a single slot type with mean initial slot degree $\beta = 2.68$, 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.[5] The figure shows analytical results according to Theorem 1 and the outcome of Monte Carlo simulations. It can be observed 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. For the example in Fig. 3, the expected packet error rates for the contention with one and two slot classes correspond to 0.264 and 0.555, respectively.

---

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

IV. APPROXIMATION USING DIFFERENTIAL EQUATIONS

The analysis presented in Section III can provide the exact probability mass function of the number of unresolved users, however, its evaluation is feasible only for a moderate number of users and a small number of slot classes. In some situations, such as when running a computer search to optimize a parameter, it is preferable to have an approximate analysis which can be efficiently evaluated.

Having this in mind, in this section we derive continuous approximations of the first and second moment of the ripple and clouds. These approximations are easy to evaluate and can be used to obtain an insight in the decoding process. Furthermore, we show how these approximations can be used to estimate the packet error rate.

A. State generating Functions

Our starting point to derive approximations of the distribution of the ripple and clouds is writing down the state generating function of the frameless ALOHA decoder, which is simply the probability generating function[6] of the random variable associated to the state of the frameless ALOHA decoder.

Following the works in [24], [26], [27], which analyze the iterative LT decoding process, let us define the state generating function of the frameless ALOHA decoder as

$$P_u(x_1, x_2, \ldots, x_k, y) := \sum_{\mathcal{S}} p_{\mathsf{c}_1, \mathsf{c}_2, \ldots, \mathsf{c}_k \mathsf{r}, u} \, {x_1}^{\mathsf{c}_1} {x_2}^{\mathsf{c}_2} \ldots {x_k}^{\mathsf{c}_k} \, y^{r-1}.$$

where $\mathcal{S}$ represents the set of all valid decoder states,

$$\mathcal{S} = \left\{ (\mathsf{c}_1, \mathsf{c}_2, \ldots, \mathsf{c}_k, \mathsf{r}) | \mathsf{c}_1 \geq 0, \mathsf{c}_2 \geq 0, \ldots \mathsf{c}_k \geq 0, \mathsf{r} \geq 0, \sum_{i=1}^{k} \mathsf{c}_i + \mathsf{r} \leq m \right\}. \tag{7}$$

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

---

[6]The probability generating function is a representation of the probability mass function using power series.

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*

$$P_u(x_1, x_2, \ldots, x_k, y) = \tag{8}$$

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

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

where $q_{h,u}$ is given by

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

and initial condition given by

$$P_n(x_1, x_2, \ldots, x_k, y) = \frac{1}{y}\prod_{h=1}^{k}\left((1-\Omega_{h,1}-\Omega_{h,0})x+\Omega_{h,1}y+\Omega_{h,0}\right)^m - \left((1-\Omega_{h,1}-\Omega_{h,0})x+\Omega_{h,0}\right)^m.$$

$$\tag{9}$$

Proof: See Appendix B. ∎

Theorem 2 can be used to derive the probability of decoding failure in a similar way as done for the LT decoder in [26]. In particular, we have that the probability that the decoder fails when exactly u users are undecoded is $1 - P_u(\mathbf{1})$, where $\mathbf{1}$ 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:

$$\mathsf{P}_u = 1 - P_u(\mathbf{1}).$$

B. First Moment

Denote by $C_h(u)$ and $R(u)$ the expected number of slots in the h-th cloud and the ripple, respectively. We have

$$C_h(u) := \sum_{\mathcal{S}} \mathsf{c}_h\, p_{\mathsf{c}_1, \mathsf{c}_2, \ldots, \mathsf{c}_k \mathsf{r}, u}$$

and

$$R(u) := \sum_{\mathcal{S}} (\mathsf{r} - 1)\, p_{\mathsf{c}_1, \mathsf{c}_2, \ldots, \mathsf{c}_k \mathsf{r}, u}$$

where we recall that $\mathcal{S}$ is the set of valid decoder states given in (7).

It is possible to express $C_h(u)$ and $R(u)$ as the first-order derivatives of the state generating function (8) in Theorem 2, evaluated at $\mathbf{1}$, where $\mathbf{1}$ is the all one vector. In particular, we have

$$C_h(u) := \frac{\partial}{\partial x_h} P_u(x_1, x_2, \ldots, x_k, y)|_{\mathbf{1}} \tag{10}$$

and

$$R(u) := \frac{\partial}{\partial y} P_u(x_1, x_2, \ldots, x_k, y)|_{\mathbf{1}}.$$

Let us first focus on $C_h(u)$. From (10) we have

$$C_h(u-1) = (1 - q_{h,u})C_h(u) - (1 - q_{h,u})\frac{\partial}{\partial x_h} P_u \left(1 - q_{1,u}, \ldots, 1 - q_{k,u}, \frac{1}{u} \right) \tag{11}$$

In [28] it was shown that, for $r \geq 6$, the drift term (11) is $\mathcal{O}(1/n^2)$, leading to

$$C_h(u-1) = (1 - q_{h,u})C_h(u) + \mathcal{O}(1/n^2). \tag{12}$$

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

$$R(u-1) = \left(1 - \frac{1}{u} \right) R(u) + \sum_{h=1}^{k} q_{h,u}C_h(u) - P_u(\mathbf{1}) \tag{13}$$

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

We again make use of a result in [28], where it was shown that for $r \geq 5$ the residual term

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

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

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 [24], [27] 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:

$$q_{h,u} = \frac{1}{n} f_h \left(\frac{u}{n} \right) - \frac{1}{n^2} g_h \left(\frac{u}{n} \right) = \frac{1}{n} f_h \left(\frac{u}{n} \right) + \mathcal{O}(1/n^2) \tag{14}$$

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 (12) and (14) we obtain the following difference equation for the h-th cloud

$$C_h(u) - C_h(u-1) = f_h \left(\frac{u}{n} \right) C_h(u) + \mathcal{O}(1/n^2) \tag{15}$$

Similarly, from (13) and (14) we obtain

$$R(u) - R(u-1) = \frac{1}{u} R(u) - \sum_{h=1}^{k} f_h \left(\frac{u}{n} \right) C_h(u) + 1 + \mathcal{O}(1/n^2) \tag{16}$$

Let us now define the expected normalized size of the clouds and ripple respectively as

$$C_h(\xi) := C_h(u)/m$$

and

$$R(\xi) := R(u)/m.$$

Making use of these definitions, and assuming $m = (1 + \epsilon)n$, we can divide both sides of (15) by m to obtain

$$C_h(\xi) - C_h(\xi - 1/n) = f_h(\xi) C_h(\xi) + \mathcal{O}(1/n^3)$$

and the same can be done for (16) leading to

$$R(\xi) - R(\xi - 1/n) = \frac{1}{\xi n} R(\xi) - \sum_{h=1}^{k} f_h(\xi) C_h(\xi) + \frac{1}{n(1+\epsilon)} + \mathcal{O}(1/n^3).$$

As shown in [28], 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

$$\hat{C}_h'(x) = f_h(x)\hat{C}_h(x) \tag{17}$$

and

$$\hat{R}'(x) = \frac{\hat{R}(x)}{x} - \sum_{h=1}^{k} f_h(x)\hat{C}_h(x) + \frac{1}{1+\epsilon}. \tag{18}$$

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 [28] that as long as u is a constant fraction of n, we have

$$C_h(u)/m = \hat{C}_h(u/k) + \mathcal{O}(1/m) \tag{19}$$

and

$$R(u)/m = \hat{R}(u/k) + \mathcal{O}(1/m). \tag{20}$$

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 (17) and (18), giving rise to the following solutions [27]

$$\hat{C}_h(x) = \mathring{c}_h\left(1 - x\,\Omega_h'(1-x) - \Omega_h(1-x)\right)$$

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

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

$$C_h(n) = \sum_{\mathcal{S}} \mathsf{c}_h\, p_{\mathsf{c}_1,\mathsf{c}_2,\dots,\mathsf{c}_k\mathsf{r},n} = m_h\left(1 - \Omega_{h,0} - \Omega_{h,1}\right)\left(1 - \prod_{l=1}^{k}\left(1 - \Omega_{l,1}\right)^{m_h}\right).$$

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

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

For the ripple we have

$$R(n) = \sum_{\mathcal{S}}(\mathsf{r} - 1)\, p_{\mathsf{c}_1,\mathsf{c}_2,\dots,\mathsf{c}_k\mathsf{r},n} = \sum_{h=1}^{k} m_h\Omega_{h,1} - 1 + \prod_{h=1}^{k}\left(1 - \Omega_{h,1}\right)^{m_h}.$$

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

$$\mathring{r} = \sum_{h=1}^{k} \frac{m_h}{m}\Omega_{h,1}\prod_{h=1}^{k}\left(1 - \Omega_{h,1}\right)^{m_h} - \frac{1}{m}\left(1 - \prod_{h=1}^{k}\left(1 - \Omega_{h,1}\right)^{m_h}\right).$$

C. Second Moment

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

$$\sigma_R(u) = \sum_{\mathcal{S}} (r-1)^2 p_{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_k \mathbf{r}, u} - R(u)^2.$$

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

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

D. Numerical Results

Fig. 4 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.

Fig. 5 shows the absolute approximation error for R, σ_R^2, C_1 and C_2 as a function of n for a setting with $k = 2$ slot types, $\beta_1 = 2.5$ and $\beta_2 = 3$, $m_1/n = m_2/n = 19/n$. In particular, every subfigure shows the error at 4 different decoding instants ($u = n/4, n/2, 3n/4, n$). All curves in the figure were obtained applying Theorem 1. We can observe how in all cases the approximation error decreases as the number of users n increases. The approximation error for the expected ripple and cloud sizes decreases approximately as $1/n$, which is in line with the error terms in (19) and (20). It is also remarkable how the approximation error for the ripple variance σ_R^2 also decreases as the number of users n increases, although this approximation was heuristically derived.

E. 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.

(a) Normalized expected cloud
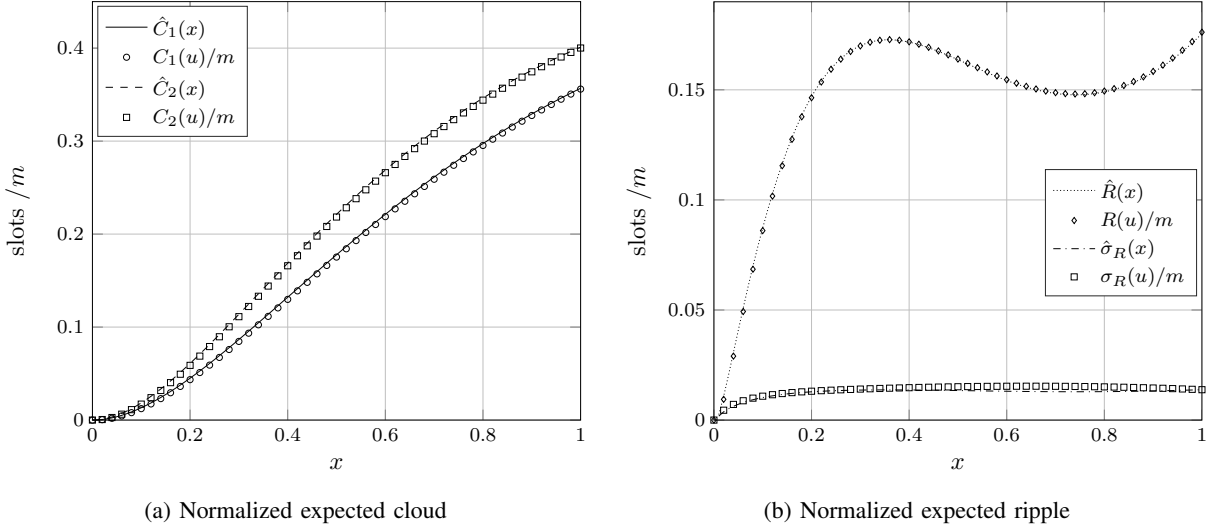
(b) Normalized expected ripple

Fig. 4: 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 $\beta_2 = 3$.

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,

$$\frac{\Pr\{\mathrm{R}_u = \mathbf{r}_u\}}{1 - \sum_{u'=u+1}^{m} \mathsf{P}_{u'}} = \frac{\Pr\{\mathrm{R}_u = \mathbf{r}_u\}}{\sum_{\mathbf{r}_u=0}^{m} \Pr\{\mathrm{R}_u = \mathbf{r}_u\}}.$$

Moreover, the figure also shows the probability density function of Gaussian random variables with mean $m\hat{R}(x)$ and standard deviation $\sqrt{m} * \hat{\sigma}_R(x)$. 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 $\sqrt{m} * \hat{\sigma}_R(x)$ is reasonably close to the actual distribution of the ripple. In particular, we can observe how the Gaussian approximation 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, owing to the aforementioned overestimation of the standard deviation of the ripple size. We would also like to remark that, while random variable R_u is discrete, we are approximating it by a continuous (Gaussian) random variable.
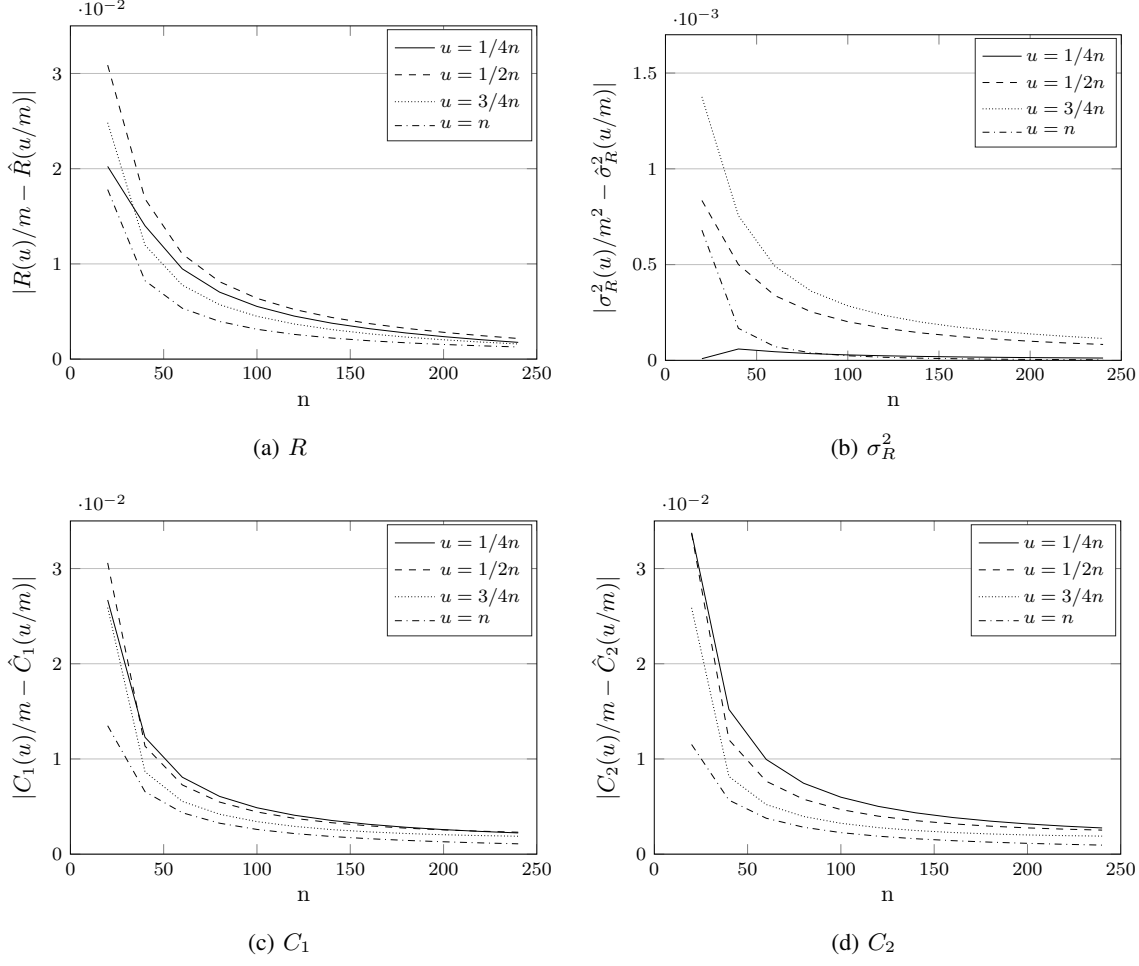
Fig. 5: Absolute approximation error for $k = 2$ slot types, $\beta_1 = 2.5$ and $\beta_2 = 3$, $m_1/n = m_2/n = 19/n$ as a function of n. The different curves represent different decoding instants ($u = n/4, n/2, 3n/4, n$). Subfigures (a), (b), (c) and (d) represent the approximation error for R, σ_R^2, C_1 and C_2, respectively.

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

$$\hat{\mathsf{P}}_u = \mathsf{Q}\left(\frac{\hat{R}(u/m)}{\hat{\sigma}_R(u/m)}\right)\left(1 - \sum_{l=u+1}^{n} \hat{\mathsf{P}}_l\right) \tag{21}$$

for $u < n$ and

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

for $u = n$, where Q is the tail distribution function of the standard normal distribution. The first term in (21) corresponds to the probability that a Gaussian distribution with mean $\hat{R}(u/m)$ and
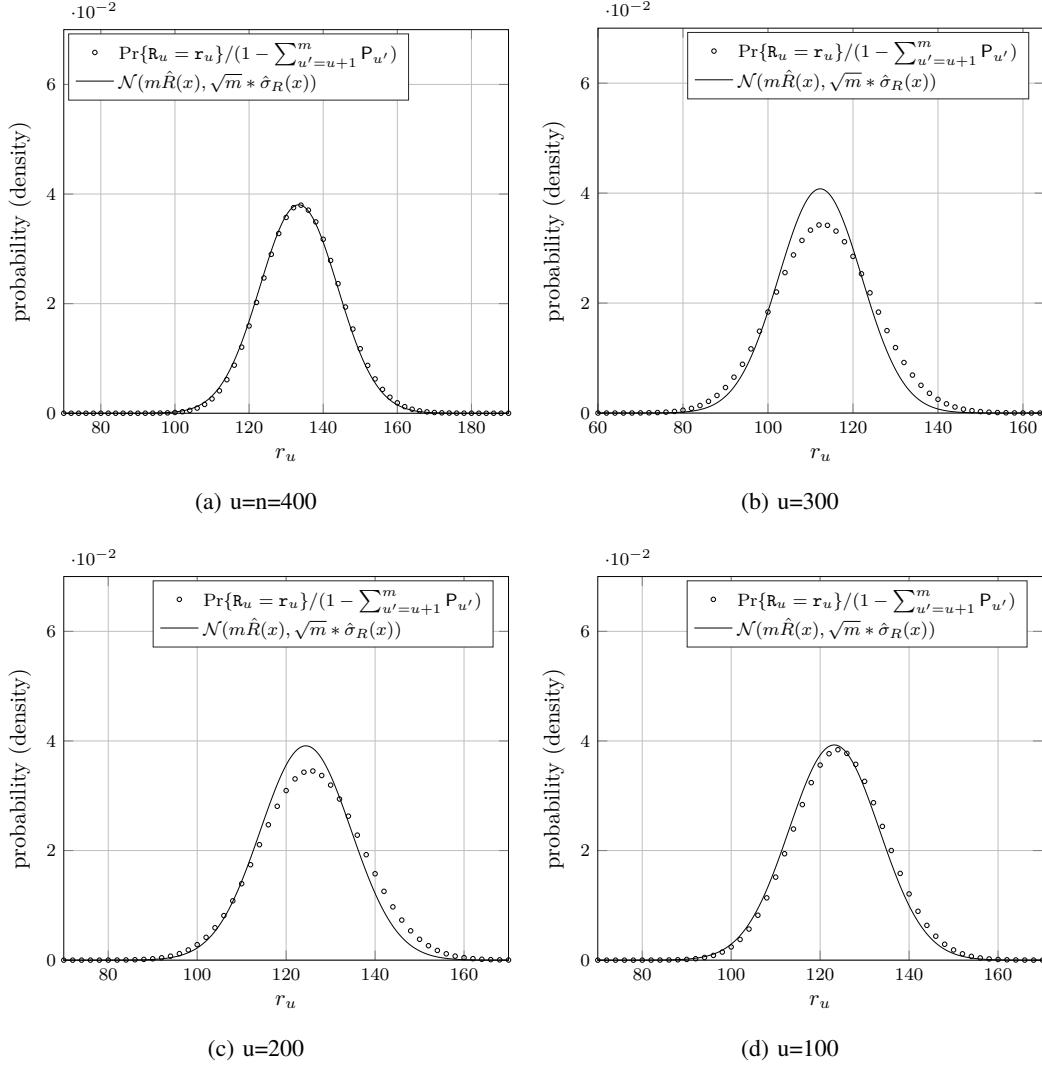
Fig. 6: Probability mass function of the ripple cardinality for $n = 400$ users, $k = 2$ slot types, $m_1 = m_2 = 380$ slots, $\beta_1 = 2.5$ and $\beta_2 = 3$ at different decoding instants. The solid line shows a Gaussian distribution with mean $m\hat{R}(x)$ and standard deviation $\sqrt{m} * \hat{\sigma}_R(x)$.

standard deviation $\hat{\sigma}_R(u/m)$ takes a negative value. Hence, this first term corresponds to the probability that our estimation of the ripple is negative. The second term in (21) implies we are conditioning to the decoding process in the previous decoding stages (n down to $u+1$) being successful. An alternative way of interpreting this is that, according to our definition of P_u, we have $\sum_{u=0}^{n} \mathsf{P}_u = 1$ and by introducing the second term we are enforcing $\sum_{u=0}^{n} \hat{\mathsf{P}}_u = 1$. Thus, we can approximate the expected packet error rate P as

$$\hat{\mathsf{P}} = \sum_{u=0}^{n} u/n \, \hat{\mathsf{P}}_u. \tag{22}$$

Fig. 7: Packet error rate P and its approximation $\hat{\mathsf{P}}$ 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 $\beta_2 = 3$.

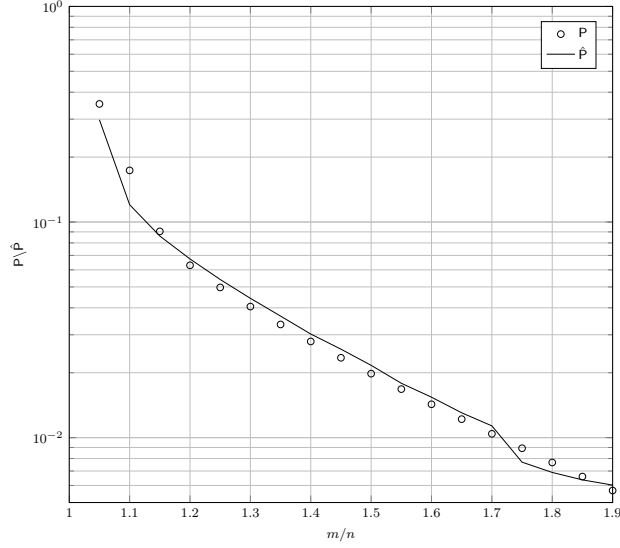Fig. 7 show the packet error rate P as well as its approximation $\hat{\mathsf{P}}$ as a function of m/n, for $n = 400$ users, $k = 2$ slot types, $m_1 = m_2 = m/2$, $\beta_1 = 2.5$ and $\beta_2 = 3$. The packet error rate P has been obtained using (6) and applying Theorem 1, which is exact, whereas the results for $\hat{\mathsf{P}}$ have been obtained using (22). 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.

V. 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.[7] We assume

---

[7]The idea of using feedback in frameless ALOHA to adapt the slot access probability has also been explored in [18], where the slot access probability is adapted after every slot in order to keep the expected number of unresolved *active* users per slot constant. However, the setting considered in [18] is different to the one considered in this paper, since it assumes that collisions of size two can be detected and resolved by employing ZigZag decoding.
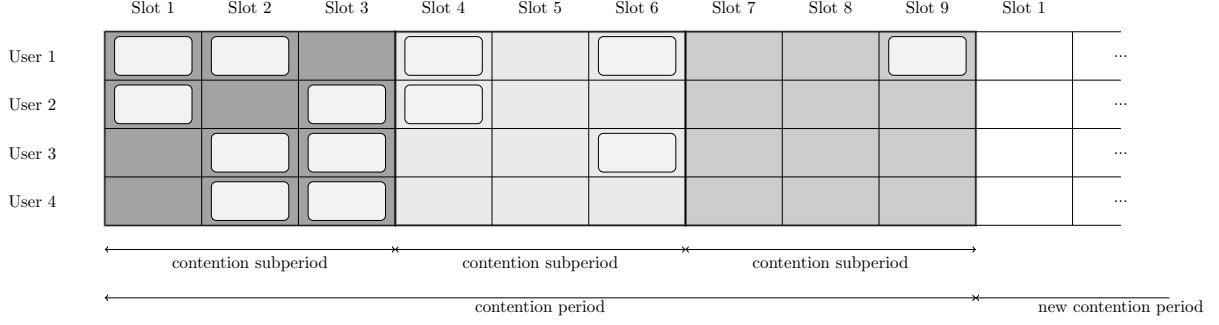
Fig. 8: An example of contention in frameless ALOHA with dynamic feedback with four users, where the contention subperiods have length $t = 3$ slots. In this example the slot access probability for the first contention subperiod, p_1, happens to be set too high. As a consequence, all the slots in the first contention subperiod present collisions. After the first $t = 3$ slots the AP notifies the 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 can be decoded. The AP then notifies the users of 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 AP to decode all users leveraging on SIC. The AP notifies all users that a new contention period starts.

that a perfect and zero-delay feedback channel exists from the access point to the terminals. This channel is used to notify users of 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, as depicted in Fig. 8. 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 AP runs the decoding algorithm to decode as many users as possible (until the ripple is empty). At this stage, denote by u the number of users which are yet unresolved, and by S_u the decoder state, with

$$\mathsf{S}_u := \left(\mathsf{c}_{1,u}, \mathsf{c}_{2,u}, \cdots, \mathsf{c}_{k,u}, \mathsf{r}_u = 0 \right)$$

where $\mathsf{c}_{i,u}$ is the number of slots in the i-th cloud. Next, based on the number of unresolved users u, the decoder state S_u, and using as latency target a contention period duration of m_T slots, the decoder computes the slot access probability to be used in the next subperiod, p_{k+1}, and signals its to the users.

In particular, in order to determine the value of p_{k+1} to be used, the AP assumes that the

remaining slots up to m_T will be of class $k + 1$. Thus, it sets $m_{k+1} = m_T - kt$. The AP now needs to determine the initial state of the frameless ALOHA decoder, given m_{k+1}, u and S_u. The first step is determining the slot degree distribution of the slots of class $k+1$, $\Omega'_{k+1,j}$, taking into account that u out of n users have already been decoded. Let us denote by I the random variable associated to the number of users which are active in a slot and by i 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

$$\Omega'_{k+1,j} = \Pr\{\mathsf{J} = \mathsf{j}\} = \sum_{\mathsf{i}=j}^{u} \Pr\{\mathsf{I} = \mathsf{i}\} \Pr\{\mathsf{J} = \mathsf{j} | \mathsf{I} = \mathsf{i}\} \tag{23}$$

$$= \sum_{\mathsf{i}=j}^{u} \binom{n}{\mathsf{i}} (p_{k+1})^{\mathsf{i}} (1 - p_{k+1})^{n-\mathsf{i}} \binom{\mathsf{i}}{\mathsf{j}} \left(\frac{u}{n}\right)^{\mathsf{j}} \left(1 - \frac{u}{n}\right)^{\mathsf{i}-\mathsf{j}}.$$

Finally, the initial state distribution of the frameless ALOHA decoder, S_u', is given by

$$\Pr\{\mathsf{S}_u' = \mathsf{c}_{1,u}, \mathsf{c}_{2,u}, \cdots, \mathsf{c}_{k+1,u}, \mathbf{r}_u\} = \frac{m_{k+1}!}{\mathsf{c}_{k+1,u}! \, \mathbf{r}_u! \, (m_{k+1} - \mathsf{c}_{k+1,n} - \mathbf{r}_u)!}$$

$$\times (1 - \Omega_{k+1,1} - \Omega_{k+1,0})^{\mathsf{c}_{k+1,u}} \, \Omega_{k+1,1}{}^{\mathbf{r}_u} \, \Omega_{k+1,0}{}^{m_{k+1} - \mathsf{c}_{k+1,u} - \mathbf{r}_u}$$

for $\mathsf{C}_{1,u} = \mathsf{c}_{1,u}$, $\mathsf{C}_{2,u} = \mathsf{c}_{2,u}$, ..., $\mathsf{C}_{k,u} = \mathsf{c}_{k,u}$, and for all non-negative $\mathsf{c}_{k+1,u}, \mathbf{r}_{k+1,u}$ such that $\mathsf{c}_{k+1,u} + \mathbf{r}_{k+1,u} \leq m_{k+1}$, with $m_{k+1} = m_T - kt$. Otherwise, we have

$$\Pr\{\mathsf{S}_u' = \mathsf{c}_{1,u}, \mathsf{c}_{2,u}, \cdots, \mathsf{c}_{k+1,u}, \mathbf{r}_u\} = 0.$$

Hence, we initialize the $\mathsf{C}_{1,u}$, $\mathsf{C}_{2,u}$ ..., $\mathsf{C}_{k,u}$ according to the decoder state at the AP (which is deterministically known), and $\mathsf{C}_{k+1,u}$ and R_u are initialized according to a multinomial distribution, similarly as in (4).

Thus, p_{k+1} is obtained formally as

$$p_{k+1} = \underset{p_{k+1}}{\arg\min} \, \mathsf{P}\left(n, m_T, u, \Omega'_{k+1,j}, \mathsf{S}_u'\right)$$

where we have stressed that P is a function of n, m_T, u, $\Omega'_{l,j}$ and S_u'. Note that $\Omega'_{l,j}$ depends implicitly on p_{k+1} through (23), and that P is obtained by using Theorem 1. Hence, it is in principle possible to run a computer search to find the value of p_{k+1} which minimizes the packet error rate. However, running such a search is computationally complex because it would require

evaluating Theorem 1 multiple times[8]. Instead, we propose minimizing the approximation of the packet error rate $\hat{P}$ derived in Section IV-E. This approach is suboptimal given the fact that we are minimizing an approximation of P, but it makes the search for p_{k+1} much faster. Thus, we obtain p_{k+1} as

$$p_{k+1} = \arg\min_{p_{k+1}} \hat{P}\left(n, m_T, u, \Omega'_{k+1,\mathbf{j}}, \mathbf{s}_u{}'\right)$$

where we remark again that that $\Omega'_{l,\mathbf{j}}$ depends implicitly on p_{k+1} through (23).

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 $n = 50$ users. Five 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 $m = 100$. This first curve was obtained from (6) after applying the finite length analysis in Theorem 1 for the particular case with only one slot type. The figure also shows the performance of a dynamic frameless ALOHA scheme in which β is changed dynamically every t slots in order to minimize $\hat{P}$ at $m_T = 100$. In particular, four different values of t are considered, 5, 10, 20 and 50 slots. The corresponding curves were obtained by means of simulations: for every value of m, the simulation run until 250 unsuccessful contention periods had been collected (a contention period is successful only if all n users are correctly decoded). Finally, the figure shows also the performance of an IRSA scheme with degree distribution $\Delta(x) = 0.25x^2 + 0.6x^3 + 0.15x^8$, taken from [8]. We can observe how the introduction of dynamic feedback decreases considerably the packet error rate. Furthermore, as one would expect, the performance of the dynamic scheme improves as t decreases, that is, as the number of feedback messages increases. The curve for $t = 20$ is particularly interesting, since one can observe how P improves after every feedback and then it degrades as time elapses; the effect is clearly visible around $m = 80$. In the simulated setting, it seems that sending feedback every $t = 10$ slots may represent a good trade-off in terms of performance vs number of feedback messages sent. Finally, if we compare the performance of IRSA to that of frameless ALOHA, we can observe that, for the latency target of $m_T = 100$, IRSA outperforms static frameless ALOHA, as well as dynamic frameless ALOHA with $t = 50$, which only uses one feedback message. However, if we allow for more feedback messages,

---

[8]If we let m/n be a constant, the number of unresolved users u be a constant fraction of n and the number of slot classes k be a constant, we have that the number of possible decoder states when u users are unresolved is $\mathcal{O}(n^{k+1})$. Furthermore, from a given decoder state when u users are unresolved, the number to states that the decoder can transit to is also $\mathcal{O}(n^{k+1})$. This is a clear indicator that the complexity of the analysis does not scale well with the number of users.
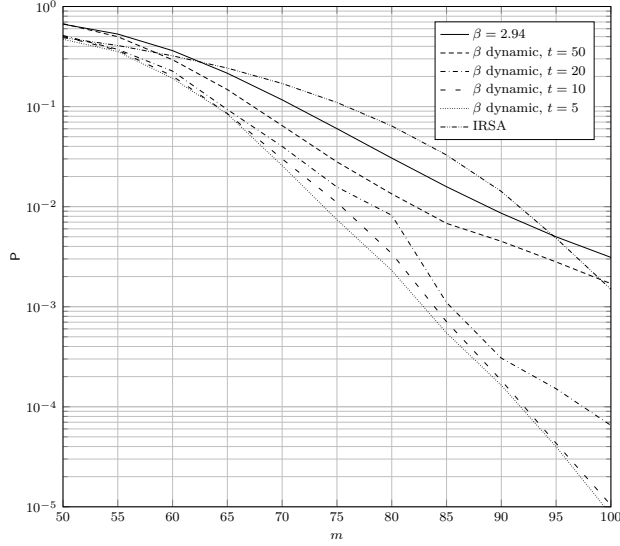
Fig. 9: Packet error rate P as a function of m for $n = 50$ users frameless ALOHA in a static and dynamic setting as well as for IRSA. Static frameless ALOHA uses $\beta = 2.94$. The different curves for dynamic frameless ALOHA represent settings in which β is changed dynamically every 5, 10, 20 and 50 slots such that the $\hat{P}$ at the end of contention-period is minimized, and the target contention period duration is set to $m_T = 100$ slots. For IRSA the degree distribution used is $\Delta(x) = 0.25x^2 + 0.6x^3 + 0.15x^8$, taken from [8].

dynamic frameless ALOHA outperforms IRSA. For example, dynamic frameless ALOHA with $t = 10$ outperforms IRSA by more than two orders of magnitude.

The results in Fig. 9 indicate how the introduction of feedback can greatly improve the performance of frameless ALOHA. Additional improvements of the performance could be achieved by assuming a more capable receiver, such as one that can distinguish collisions of size 2 [18] or a receiver which can decode collisions of size $\ell > 1$ [29].

VI. 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 moderate number of users. 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.
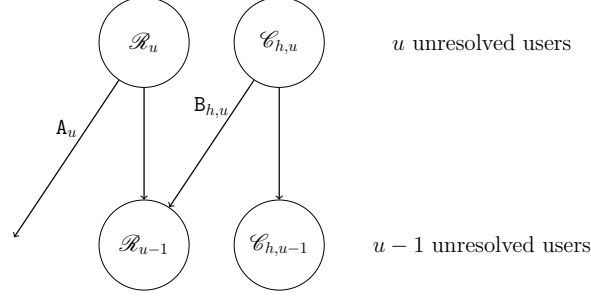
Fig. 10: Evolution of the ripple and the h-th cloud through decoding and removal of replicas. Resolution of a single user, i.e., decoding a replica of its packet and removal of the other replicas, causes $\mathtt{A}_u$ slots to leave the ripple and $\mathtt{B}_{h,u}$ slots to leave the h-th cloud and enter the ripple.

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.

APPENDIX A
PROOF OF THEOREM 1

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 are 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 $\mathtt{r}_u > 0$, 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 $\mathscr{C}_{h,u}$ and entering $\mathscr{R}_{u-1}$ in the transition, denoted by $\mathtt{b}_{h,u}$ and with associated random variable given by $\mathtt{B}_{h,u}$. 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[9]. Thus, random variable $\mathtt{B}_{h,u}$ is binomially distributed with parameters

---

[9]In fact, it is users who choose their neighbor slots uniformly at random and without replacement.

$\mathsf{c}_{h,u}$ and $q_{h,u}$, being $q_{h,u}$ the probability of a generic slot y of type h leaving $\mathscr{C}_{h,u}$ to enter $\mathscr{R}_{u-1}$,

$$q_{h,u} := \Pr\{y \in \mathscr{R}_{u-1} | y \in \mathscr{C}_{h,u}\} = \frac{\Pr\{y \in \mathscr{R}_{u-1}\,,\, y \in \mathscr{C}_{h,u}\}}{\Pr\{y \in \mathscr{C}_{h,u}\}}. \tag{24}$$

We shall first focus on the numerator of (24) and we shall condition it to the slot having degree d, $\Pr\{y \in \mathscr{R}_{u-1}\,,\, y \in \mathscr{C}_{h,u} | \deg(y) = d\}$. 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 $n-u$ 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

$$\Pr\{y \in \mathscr{R}_{u-1}\,,\, y \in \mathscr{C}_{h,u} | \deg(y) = d\} = \frac{d}{n}(d-1)\frac{u-1}{n-1}\frac{\binom{n-u}{d-2}}{\binom{n-2}{d-2}} \tag{25}$$

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

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

Let us now concentrate on the denominator of (24), 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

$$\Pr\{y \in \mathscr{C}_{h,u}\} = 1 - \sum_{d=1}^{n} \Omega_{h,d} u \frac{\binom{n-u}{d-1}}{\binom{n}{d}} - \sum_{d=0}^{n} \Omega_{h,d} \frac{\binom{n-u}{d}}{\binom{n}{d}} \tag{26}$$

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 (25) and (26) in (24), the expression of $q_{h,u}$ in (3) is obtained, and the variation in size of the clouds is determined, i.e., the PMFs of the random variables $\mathsf{B}_{h,u}$ are obtained.

We focus next on the variation in size of the ripple in the transition from u to $u-1$ unresolved users. In this transition, some slots enter the ripple (in total $\sum_h \mathsf{b}_{h,u}$ 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[10], 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 and all slots in the ripple that are connected to c will leave the ripple in the transition. Hence, slot y will leave the ripple, and, additionally, the remaining $\mathbf{r}_u - 1$ 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 $\mathbf{A}_u$ is given by

$$\Pr\{\mathbf{A}_u = \mathbf{a}_u | \mathbf{R}_u = \mathbf{r}_u\} = \binom{\mathbf{r}_u - 1}{\mathbf{a}_u - 1}\left(\frac{1}{u}\right)^{\mathbf{a}_u - 1}\left(1 - \frac{1}{u}\right)^{\mathbf{r}_u - \mathbf{a}_u}.$$

Note that, by definition, we have that $\mathbf{r}_{u-1} = \mathbf{r}_u - \mathbf{a}_u + \sum_{h=1}^{k} \mathbf{b}_{h,u}$ and $\mathbf{c}_{h,u-1} = \mathbf{c}_{h,u} - \mathbf{b}_{h,u}$. Hence, the probability of transiting from state $\mathbf{s}_u = (\mathbf{c}_{1,u}, \mathbf{c}_{2,u}, \cdots, \mathbf{c}_{k,u}, \mathbf{r}_u)$ to state $\mathbf{s}_u + \mathbf{w}$, with $\mathbf{w}$ given by

$$\mathbf{w} = \left(-\mathbf{b}_{1,u}, -\mathbf{b}_{2,u}, \cdots, -\mathbf{b}_{k,u}, \sum_{h=1}^{k} \mathbf{b}_{h,u} - \mathbf{a}_u\right)$$

corresponds to

$$\Pr\{\mathbf{B}_{1,u} = \mathbf{b}_{1,u}, \mathbf{B}_{2,u} = \mathbf{b}_{2,u}, \cdots, \mathbf{B}_{k,u} = \mathbf{b}_{k,u}, \mathbf{A}_u = \mathbf{a}_u\}. \tag{27}$$

The proof is finalized by observing that the joint distribution in (27), due to independence of $\mathbf{B}_{h,u}$, $h = 1, 2, \cdots, k$, and $\mathbf{A}_u$, is obtained as the product of the PMFs of the individual random variables, which yields (2). ∎

APPENDIX B

PROOF OF THEOREM 2

We provide a proof only for the case of $k = 2$ slot types. The proof follows closely [26, Theorem 1]. By definition, the state generating function of the frameless ALOHA decoder when $u - 1$ users are undecoded can be written as

$$P_{u-1}(x_1, x_2, y) = \sum_{\substack{\mathbf{c}_1' \geq 0 \\ \mathbf{c}_2' \geq 0 \\ r' \geq 1}} p_{\mathbf{c}_1', \mathbf{c}_2', r', u-1} x_1^{\mathbf{c}_1'} x_2^{\mathbf{c}_2'} y^{r'-1} \tag{28}$$

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 $\mathbf{s}_u = (\mathbf{c}_{1,u}, \mathbf{c}_{2,u}, \mathbf{r}_u)$ to the future state $\mathbf{s}_{u-1} = (\mathbf{c}_{1,u-1}', \mathbf{c}_{2,u-1}', \mathbf{r}_{u-1}')$. In the remainder of the proof, we shall drop the subscript u for the

---

[10]If the ripple is empty, $\mathbf{r}_u = 0$, no slots can leave the ripple. Moreover, decoding stops, so there is no transition.

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 (28) as follows

$$P_{u-1}(x_1, x_2, y) = \frac{1}{y} \sum_{\substack{c_1 \geq 0 \\ c_2 \geq 0 \\ r \geq 1}} p_{c_1, c_2, r, u} \sum_{\substack{1 \leq a \leq r \\ 0 \leq b_1 \leq c_1 \\ 0 \leq b_2 \leq c_2 \\ r-a+b_1+b_2 > 0}} x_1^{c_1-b_1} x_2^{c_2-b_2} y^{r-a+b_1+b_2}$$

$$\times \binom{c_1}{b_1} q_{1,u}^{b_1} (1-q_{1,u})^{c_1-b_1} \binom{c_2}{b_2} q_{2,u}^{b_2} (1-q_{2,u})^{c_2-b_2} \binom{r-1}{a-1} \left(\frac{1}{u}\right)^{a-1} \left(1-\frac{1}{u}\right)^{r-a}.$$

By grouping different terms together, we obtain

$$P_{u-1}(x_1, x_2, y) = \frac{1}{y} \sum_{\substack{c_1 \geq 0 \\ c_2 \geq 0 \\ r \geq 1}} p_{c_1, c_2, r, u} \sum_{\substack{1 \leq a \leq r \\ 0 \leq b_1 \leq c_1 \\ 0 \leq b_2 \leq c_2 \\ r-a+b_1+b_2 > 0}} \binom{c_1}{b_1} (y\, q_{1,u})^{b_1} (x_1(1-q_{1,u}))^{c_1-b_1}$$

$$\times \binom{c_2}{b_2} (y\, q_{2,u})^{b_2} (x_2(1-q_{2,u}))^{c_2-b_2} \binom{r-1}{a-1} \left(\frac{1}{u}\right)^{a-1} \left(y\left(1-\frac{1}{u}\right)\right)^{r-a}.$$

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

$$P_{u-1}(x_1, x_2, y) = \frac{1}{y} \sum_{\substack{c_1 \geq 0 \\ c_2 \geq 0 \\ r \geq 1}} p_{c_1, c_2, r, u} \left(\sum_{\substack{1 \leq a < r \\ 0 \leq b_1 \leq c_1 \\ 0 \leq b_2 \leq c_2}} \binom{c_1}{b_1} (y\, q_{1,u})^{b_1} (x_1(1-q_{1,u}))^{c_1-b_1} \right. \tag{29}$$

$$\times \binom{c_2}{b_2} (y\, q_{2,u})^{b_2} (x_2(1-q_{2,u}))^{c_2-b_2} \binom{r-1}{a-1} \left(\frac{1}{u}\right)^{a-1} \left(y\left(1-\frac{1}{u}\right)\right)^{r-a} \bigg)$$

$$+ \left(\frac{1}{u}\right)^{r-1} \left(\sum_{\substack{0 \leq b_1 \leq c_1 \\ 1 \leq b_2 \leq c_2}} \binom{c_1}{b_1} (y\, q_{1,u})^{b_1} (x_1(1-q_{1,u}))^{c_1-b_1} \right.$$

$$\times \binom{c_2}{b_2} (y\, q_{2,u})^{b_2} (x_2(1-q_{2,u}))^{c_2-b_2}$$

$$+ (x_2(1-q_{2,u}))^{c_2} \sum_{1 \leq b_1 \leq c_1} \binom{c_1}{b_1} (y\, q_{1,u})^{b_1} (x_1(1-q_{1,u}))^{c_1-b_1} \bigg)$$

By using in (29) the results of the following two finite sums

$$\sum_{\mathtt{b}_1=0}^{\mathtt{c}_1} \binom{\mathtt{c}_1}{\mathtt{b}_1} (y\, q_{1,u})^{\mathtt{b}_1} (x_1(1-q_{1,u}))^{\mathtt{c}_1-\mathtt{b}_1} = (y\, q_{1,u} + x_1(1-q_{1,u}))^{\mathtt{c}_1}$$

$$\sum_{\mathtt{a}=1}^{\mathtt{r}-1} \binom{\mathtt{r}-1}{\mathtt{a}-1} \left(\frac{1}{u}\right)^{\mathtt{a}-1} \left(y\left(1-\frac{1}{u}\right)\right)^{\mathtt{r}-\mathtt{a}} = \left(\frac{1}{u} + y(1-\frac{1}{u})\right)^{\mathtt{r}-1} - \left(\frac{1}{u}\right)^{\mathtt{r}-1}$$

after some manipulation, we obtain

$$P_{u-1}(x_1, x_2, y) = (y\, q_{1,u} + x_1(1-q_{1,u}))^{\mathtt{c}_1} (y\, q_{2,u} + x_2(1-q_{2,u}))^{\mathtt{c}_2}$$
$$\times \left(\frac{1}{u} + y(1-\frac{1}{u})\right)^{\mathtt{r}-1} - \left(\frac{1}{u}\right)^{\mathtt{r}-1} (x_1(1-q_{1,u}))^{\mathtt{c}_1} (x_2(1-q_{2,u}))^{\mathtt{c}_2}$$

which yields (8).

In order to complete the proof, we need to provide a proof for (9), which specifies the initial condition of the recursive expression given in (8). By definition, $p_{\mathtt{c}_1,\mathtt{c}_2,\mathtt{r},n}$ is simply the probability that before decoding starts ($u = n$) we have exactly $\mathtt{c}_1$ slots in the first cloud, $\mathtt{c}_2$ slots in the second cloud and $\mathtt{r}$ slots in the ripple. Let us decompose $\mathtt{r}$ as $\mathtt{r} = \mathtt{r}_1 + \mathtt{r}_2$, , with $\mathtt{r}_1 \geq 0$ and $\mathtt{r}_2 \geq 0$, where $\mathtt{r}_1$ and $\mathtt{r}_2$ represent the number of slots of type 1 and type 2 in the ripple, respectively. We have

$$P_n(x_1, x_2, y) = \sum_{\mathcal{S}} p_{\mathtt{c}_1,\mathtt{c}_2,\mathtt{r},n} x_1^{\mathtt{c}_1} x_2^{\mathtt{c}_2} y^{\mathtt{r}-1} = \sum_{\mathcal{S}'} p_{\mathtt{c}_1,\mathtt{c}_2,\mathtt{r}_1+\mathtt{r}_2,n} x_1^{\mathtt{c}_1} x_2^{\mathtt{c}_2} y^{\mathtt{r}_1+\mathtt{r}_2-1} \qquad (30)$$

with

$$\mathcal{S}' = \Big\{ (\mathtt{c}_1, \mathtt{c}_2, \mathtt{r}_1, \mathtt{r}_2) | \mathtt{c}_1 \geq 0, \mathtt{c}_2 \geq 0, \mathtt{r}_1 \geq 0, \mathtt{r}_2 \geq 0, \mathtt{r}_1 + \mathtt{r}_2 \geq 1,$$
$$\mathtt{c}_1 + \mathtt{r}_1 \leq m_1, \mathtt{c}_2 + \mathtt{r}_2 \leq m_2 \Big\}$$

where we remark that though $\mathtt{r}_1$ and $\mathtt{r}_2$ can take value 0, they cannot do so at the same time, since otherwise decoding would not be able to start.

From (4), we have that

$$p_{\mathtt{c}_1,\mathtt{c}_2,\mathtt{r}_1+\mathtt{r}_2,n} = \prod_{h=1}^{2} \frac{m_h!}{\mathtt{c}_h!\, \mathtt{r}_h!\, (m_h - \mathtt{c}_h - \mathtt{r}_h)!} (1 - \Omega_{h,1} - \Omega_{h,0})^{\mathtt{c}_h} \Omega_{h,1}^{\mathtt{r}_n} \Omega_{h,0}^{m_h-\mathtt{c}_h-\mathtt{r}_h}. \qquad (31)$$

If we replace this expression into the right hand side of (30), compute the summation over the set of all valid states $\mathcal{S}'$ given in (31), 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. ∎

Appendix C

Initial Conditions for Dynamic Feedback

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

$$C_h(u) = \sum_{\mathcal{S}} \mathsf{c}_h \, p_{\mathsf{c}_1, \mathsf{c}_2, \dots, \mathsf{c}_k \mathsf{r}, n} = \mathsf{c}_{h,u} \left(1 - \left(1 - \Omega_{k+1,1} \right)^{m_{k+1}} \right)$$

thus, we have

$$\mathring{c}_h = \frac{\mathsf{c}_{h,u}}{m'} \frac{\left(1 - \left(1 - \Omega_{k+1,1} \right)^{m_{k+1}} \right)}{1 - \Omega_{h,1} - \Omega_{h,0}}.$$

For $h = k + 1$ we have

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

which yields

$$\mathring{c}_h = \frac{m_h}{m} \left(1 - \left(1 - \Omega_{h,1} \right)^{m_h} \right).$$

For the ripple we have

$$R(u) = \sum_{\mathcal{S}} (\mathsf{r} - 1) \, p_{\mathsf{c}_1, \mathsf{c}_2, \dots, \mathsf{c}_k \mathsf{r}, n} = \sum_{h=1}^{k} \mathsf{c}_{h,u} + m_{k+1} \Omega_{k+1,1} - 1 + \left(1 - \Omega_{k+1,1} \right)^{m_{k+1}}$$

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

$$\mathring{r} = \sum_{h=1}^{k} \frac{\mathsf{c}_{h,u}}{m'} \left(1 - \frac{\left(1 - \left(1 - \Omega_{k+1,1} \right)^{m_{k+1}} \right)}{1 - \Omega_{h,1} - \Omega_{h,0}} \right) + \frac{m_{k+1}}{m'} \Omega_{k+1,1} - \frac{1}{m'} \left(1 - \left(1 - \Omega_{k+1,1} \right)^{m_{k+1}} \right).$$

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 Global Commun. Conf.*, Singapore, Dec. 2017, pp. 1–6.

[3] 3GPP, "3GPP TS 36.321 V15.4.0: Medium access control 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," *ACM 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 Int. Conf. Commun.*, Kyoto, Japan, Jun. 2011.

[10] G. Liva, E. Paolini, M. Lentmaier, and M. Chiani, "Spatially-coupled random access on graphs," in *Proc. of IEEE Int. Symp. Inf. Theory*, 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. Bajović, 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 Annu. IEEE Symp. on Found. of Comp. 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] S. Ogata and K. Ishibashi, "Application of zigzag decoding in frameless ALOHA," *IEEE Access*, vol. 7, pp. 39 528–39 538, Apr. 2019.

[19] 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.

[20] 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.

[21] M. Fereydounian, X. Chen, H. Hassani, and S. S. Bidokhti, "Non-asymptotic coded slotted ALOHA," in *Proc. of IEEE Int. Symp. Inf. Theory*, Paris, France, Jul. 2019, pp. 111–115.

[22] 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.

[23] 3GPP, "3GPP TR 38.913 V15.0.0: Study on scenarios and requirements for next generation access technologies; (Release 15)," Tech. Rep., Jun. 2018.

[24] R. Karp, M. Luby, and A. Shokrollahi, "Finite length analysis of LT codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Chicago, IL, USA, Jun. 2004.

[25] 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.

[26] A. Shokrollahi, "Theory and applications of Raptor codes," *Mathknow*, vol. 3, pp. 59–89, 2009.

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

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

[29] F. Lázaro and Č. Stefanović, "Finite-length analysis of frameless ALOHA with multi-user detection," *IEEE Commun. Lett.*, vol. 21, no. 4, pp. 769–772, 2016.