



CHALMERS

Chalmers Publication Library

A structured irregular repetition slotted ALOHA scheme with low error floors

This document has been downloaded from Chalmers Publication Library (CPL). It is the author's version of a work that was accepted for publication in:

2017 IEEE International Conference on Communications, ICC 2017, Paris, France, 21-25 May 2017 (ISSN: 1550-3607)

Citation for the published paper:

Paolini, E. ; Liva, G. ; Graell i Amat, A. (2017) "A structured irregular repetition slotted ALOHA scheme with low error floors". 2017 IEEE International Conference on Communications, ICC 2017, Paris, France, 21-25 May 2017

<http://dx.doi.org/10.1109/ICC.2017.7996564>

Downloaded from: <http://publications.lib.chalmers.se/publication/251836>

Notice: Changes introduced as a result of publishing processes such as copy-editing and formatting may not be reflected in this document. For a definitive version of this work, please refer to the published source. Please note that access to the published version might require a subscription.

Chalmers Publication Library (CPL) offers the possibility of retrieving research publications produced at Chalmers University of Technology. It covers all types of publications: articles, dissertations, licentiate theses, masters theses, conference papers, reports etc. Since 2006 it is the official tool for Chalmers official publication statistics. To ensure that Chalmers research results are disseminated as widely as possible, an Open Access Policy has been adopted. The CPL service is administrated and maintained by Chalmers Library.

(article starts on next page)

A Structured Irregular Repetition Slotted ALOHA Scheme with Low Error Floors

Enrico Paolini*, Gianluigi Liva†, and Alexandre Graell i Amat‡

*Department of Electrical, Electronic, and Information Engineering, University of Bologna, Cesena, Italy

†Institute of Communications and Navigation of DLR (German Aerospace Center), Wessling, Germany

‡Department of Signals and Systems, Chalmers University of Technology, Gothenburg, Sweden

Abstract—We propose graph-defined IRSA (G-IRSA), a new approach to design irregular repetition slotted ALOHA (IRSA) uncoordinated multiple access schemes for a controlled-size population of users that become active sporadically. The proposed scheme considers a joint design of the distribution according to which users select their repetition factors and the distribution determining how many packet replicas are transmitted per slot, as well as the *connectivity* of the underlying graph, i.e., to which slots users transmit. This is in sharp contrast to standard IRSA, where only the users degree distribution is optimized, while active users place their packet replicas uniformly at random and thus there is no control on how many replicas are transmitted per slot and in which slots users transmit. The key idea is to establish a link between the IRSA for the considered scenario and low-density parity-check (LDPC) codes for transmission over the binary erasure channel (BEC). Using this parallelism, the design of a G-IRSA scheme can be cast as the design of a high-rate LDPC code over the BEC. We show that the proposed scheme achieves significantly lower error floors than the original IRSA and very good decoding thresholds.

I. INTRODUCTION

Uncoordinated multiple access techniques (e.g., slotted ALOHA and its variants [1]–[3]) are fundamental in scenarios where users transmit over a shared channel and coordination between them is not possible. Recently, research on uncoordinated multiple access techniques based on the classical slotted ALOHA has been revived following the introduction of contention resolution diversity slotted ALOHA (CRDSA) in [4]. In CRDSA transmission is organized into frames, each consisting of the same number of slots, M . Users transmit a fixed number of copies of their packets within a frame, as already proposed in [3] and, crucially, exploit the introduced redundancy at the receiver side to resolve collisions by using successive interference cancellation (SIC). This allows to achieve significantly higher throughputs.

The idea in [4] was further developed in [5] by allowing users to repeat their packets a variable number of times according to a predefined probability distribution. To convey the variable repetition rate principle, the scheme of [5] was named irregular repetition slotted ALOHA (IRSA). In particular, it was shown that the IRSA scheme can equivalently be represented by a bipartite graph and the SIC can be seen as a peeling decoding on the bipartite graph, similar to the

peeling decoding of codes on graphs over the binary erasure channel (BEC). This connection allows borrowing tools from codes on graphs to analyze and optimize IRSA: Its asymptotic performance in the limit of infinitely large frame lengths, as well as infinitely large population size, can be predicted using density evolution (DE) and the typical error floor that it appears in the packet loss rate for finite frame lengths can be analyzed by performing a stopping set analysis [6]–[8]. In [5], the distribution of the repetitions was optimized using DE, leading to significant performance improvements (for large frames) with respect to CRDSA. On the other hand, since in IRSA users transmit their packet replicas in slots of a frame chosen uniformly at random, the probability distribution determining the number of transmitted replicas per slot is not under the control of the system designer, but is fully determined by the average number of repetitions and the load.

In this paper, we consider a radically different approach to design an IRSA scheme. The proposed approach follows from the observation that, in many actual systems, users go through a login procedure to join the network. After this phase, we can leverage on the fact that the population of users in the network is known to the network access point. We hence exploit this information to introduce a finer control on the IRSA access protocol. In particular, we consider the scenario characterized by a large (but controlled in size) population of users with maximum size $N \gg M$. Users become active sporadically, i.e., the number of users that are active in a given frame (in the sense that they have a packet to transmit in that frame), N_a , is a small fraction of the total number of users. Given a maximum number of supported users, N , the access point can construct a bipartite graph with N variable nodes (one per potential user) and M check nodes (one per slot). The graph can be designed by controlling the degree distributions at both node sides, hence enforcing control not only on the repetition distribution, but also on the distribution of the number of transmissions per slot. More importantly, girth optimization techniques [9] can be used to limit the effect of small stopping sets. Each user joining the network is assigned to a different variable node of the graph (e.g., associated with the used ID), which determines the slots in which the user will transmit, whenever it is active.

For the proposed scheme, the SIC process applied at the receiver shares several similarities with the decoding of an $(N, N - M)$ high-rate low-density parity-check (LDPC) code for transmission over the BEC, where the N_a erased variable

The work of E. Paolini was supported by ESA/ESTEC under Contract no. 4000118331/16/UK/ND. The work of A. Graell i Amat was supported by the Swedish Research Council under grant 2016-04253.

nodes (VNs) correspond to the N_a active users. We use this analogy to optimize the VN and the check node (CN) degree distributions of the $(N, N - M)$ LDPC code ensemble to maximize its belief propagation threshold over the BEC and design an IRSA scheme by constructing a particular LDPC code within the ensemble. For each user in the system the designed LDPC code determines its repetition factor as well as the slots in which it will transmit when it becomes active, i.e., the placement of packet replicas is completely deterministic (even if users activate unpredictably). This is in contrast to standard IRSA, where only the repetition distribution is optimized and packet replicas are placed randomly. We show that the proposed IRSA scheme, dubbed graph-defined IRSA (G-IRSA), yields significantly lower error floors than the original IRSA yet achieving excellent decoding thresholds.

II. SYSTEM MODEL

We consider a slotted random access scheme where slots are grouped in medium access control (MAC) frames, simply referred to as *frames* in the following, all with the same length M (in slots). The time duration of each slot is T_{slot} and the frame duration is T_{frame} , thus $M = T_{\text{frame}}/T_{\text{slot}}$. Users contending for the medium access form a population of size N_P , where the maximum value of N_P is $N \gg M$. Users are slot- and frame-synchronous and each user attempts at most one packet transmission per frame. Neglecting guard times, the transmission time duration of a packet is T_{slot} . We say that a user is active in a given frame if it has a packet to be transmitted within the frame. In particular, we consider a scenario where users become active sporadically, with *activation probability* π . Since users become active independently of each other, the number of active users within a frame is modeled by a random variable N_a that is binomially distributed with mean $\mathbb{E}[N_a] = \pi N_P$. The expected channel load (representing the expected number of packet transmissions per slot) is

$$G = \frac{\mathbb{E}[N_a]}{M}.$$

The channel model considered in this paper is the *collision channel*, widely adopted in analyzing random access protocols. Accordingly, in each slot the receiver can always discriminate between a “silence” (no active user has transmitted in that slot), a signal corresponding to a unique packet, or a signal being the result of a collision. In case a unique packet is detected, it is always successfully decoded, while in case of a collision the detected signal provides no information to the receiver about the number and the content of colliding packets. As soon as a packet is decoded, the interference contribution of its copies is removed from the respective slots. We refer to this process as SIC.¹ The process is iterated until no more clean packets can be found in the MAC frame.

III. GRAPH-DEFINED IRSA ACCESS PROTOCOL

In this section, we describe the proposed access scheme. We start with a brief description of the original IRSA to

better highlight the differences with respect to the proposed scheme. The IRSA scheme works as follows. Every time a user becomes active, it selects a repetition degree d randomly by sampling a probability mass function $\{\Lambda_d\}_{d=2}^{d_{\text{max}}}$ known to all users, and generates d copies of its packet, called replicas. The d packet replicas, each of duration T_{slot} , are then transmitted in d slots chosen uniformly at random out of the M slots of the frame. The system designer has full control of the distribution $\Lambda(x) = \sum_{d=2}^{d_{\text{max}}} \Lambda_d x^d$, which can be properly optimized. On the other hand, the distribution $P(x) = \sum_{r=0}^{r_{\text{max}}} P_r x^r$ defining the number replicas per slot is not a design parameter, but is fully determined by $\Lambda(x)$ and G .

The proposed G-IRSA access protocol allows controlling both $\Lambda(x)$ and $P(x)$ and the graph connectivity. In the limiting case where $N_P = N$ (i.e., full system setting), assume that each of the N users of the population is represented by a VN in a bipartite graph and each of the M slots of a frame is represented by a CN. Furthermore, assume a given connectivity between VNs and CNs according to the degree distributions $\Lambda(x)$ and $P(x)$. The resulting bipartite graph is equivalent to that of an $(N, N - M)$ high-rate LDPC code of length N and dimension $(N - M)$ with VN degree distribution $\Lambda(x)$ and CN degree distribution $P(x)$, both from a node perspective. For ease of exposition, we will denote this bipartite graph by $\mathcal{G}_{\text{LDPC}}$.

Consider now a subset of N_a active users and the resulting bipartite graph consisting of the N_a VNs of $\mathcal{G}_{\text{LDPC}}$ corresponding to the N_a active users and their neighboring CNs. We denote this bipartite graph by \mathcal{G}_{N_a} . Also, consider transmission of the $(N, N - M)$ LDPC code over a BEC that erases the VNs corresponding to the N_a active users and the resulting *residual graph* obtained by removing all nonerased VNs and their adjacent edges from $\mathcal{G}_{\text{LDPC}}$. It is easy to see that this residual graph is identical to \mathcal{G}_{N_a} . Therefore, the IRSA for the scenario with a large population of users of size $N_P = N$ that become active sporadically within a frame may equivalently be seen as using the $(N, N - M)$ high-rate LDPC code for transmission over the BEC.

A. Design of $\Lambda(x)$, $P(x)$, and the graph connectivity

The analogy with LDPC codes for the BEC established above allows us to propose G-IRSA, a novel IRSA scheme based on the design of an LDPC code. In particular, the optimization of the G-IRSA system becomes equivalent to optimizing an LDPC code for transmission over the BEC. Indeed, the performance of the G-IRSA system corresponds to the performance of the underlying LDPC code over the BEC. More precisely, if the decoding threshold of the LDPC code is ϵ^* then the decoding threshold of the resulting IRSA scheme, i.e., the maximum channel load for which all users can be resolved in the limit of infinitely large population size and block length (their ratio remaining constant), is

$$G^* = \frac{\epsilon^* N_P}{M}.$$

To design the G-IRSA scheme, we first optimize the VN degree and CN degree distributions $\Lambda(x)$ and $P(x)$ of the

¹We refer to [4], [5] for details on the accuracy of the model in practical conditions.

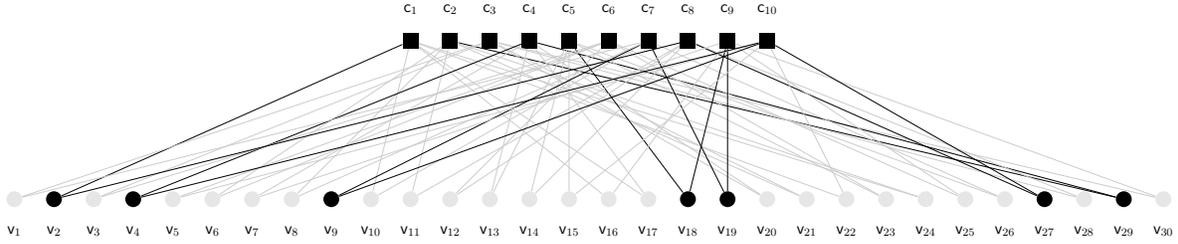


Fig. 1. Example of bipartite graph of a G-IRSA scheme with a population of $N = 30$ users, $M = 10$ slots, and $N_a = 7$ active users corresponding to VNs $v_2, v_4, v_9, v_{18}, v_{19}, v_{27}$, and v_{29} . Effectively, the receiver sees the subgraph highlighted in black.

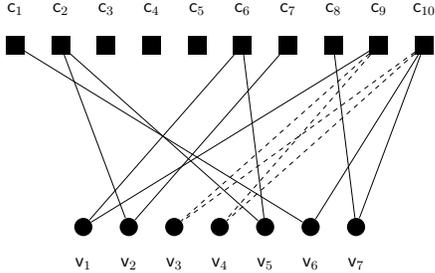


Fig. 2. Example of bipartite graph of a CRDSA scheme with $\Lambda(x) = 2$ and $N_a = 7$ active users that transmit over $M = 10$ slots.

LDPC code ensemble such that, in the limit of infinitely large block length, it achieves a good decoding threshold ϵ^* . For a given N and M a good $(N, N - M)$ LDPC code within the ensemble can then be constructed from the ensemble defined by $\Lambda(x)$ and $P(x)$ using, e.g., the progressive edge growth (PEG) algorithm [9] to avoid small stopping sets, which dominate the performance in the error floor region [6], [8]. Note that, again in the full system setting $N_p = N$, in terms of the IRSA scheme a given LDPC code implies that each of the N users of the population is assigned to a column of the parity-check matrix of the code, \mathbf{H} .

In practice, this is possible by envisaging a login procedure through which users join the system and during which an association is established between a new user and a column of \mathbf{H} . In other words, the designed LDPC code determines for each of the N users both the repetition factor and the slots in which it will transmit through its edges. Thus, the way users place their packet replicas in the slots of the frame is completely deterministic (determined by the constructed LDPC code and by the assignment of columns of \mathbf{H} to the users). This is in sharp contrast with the standard IRSA, where the placement is completely random. Yet, the proposed scheme exhibits features typical of *uncoordinated* protocols: After a user is logged in, it activates unpredictably in each frame and, although it must transmit in its assigned slots, the receiver has no control on which users become active in the same frame. Therefore, packet collisions are likely to occur.

Example 1. Consider a population of $N_p = N = 30$ users that transmit over frames of $M = 10$ slots and the representation of this scenario by the bipartite graph in Fig. 1 such that VN i corresponds to user i , $i = 1, \dots, 30$. Disregard for the moment the connections between VNs and CNs. Assume for a given frame that the set of $N_a = 7$ active users is $\{2, 4, 9, 18, 19, 27, 29\}$. In CRDSA and standard IRSA, the 7

active users would select their repetition factors according to a given distribution $\Lambda(x)$ and transmit their copies in slots chosen uniformly at random. An example of the resulting bipartite graph for a CRDSA scheme with $\Lambda(x) = x^2$ is depicted in Fig. 2. In the graph, VNs v_3 and v_4 form a stopping set and thus, users 3 and 4 cannot be resolved. Since users place their copies randomly, small stopping sets may occur. For this example, the girth of the graph is 4 and therefore the size of the smallest stopping set is 2 [10].

Consider now the connections defined by the bipartite graph in Fig. 1 with the same set of $N_a = 7$ active users. In this case, effectively, the receiver will “see” the subgraph defined by VNs $\{v_2, v_4, v_9, v_{18}, v_{19}, v_{27}, v_{29}\}$ and their neighboring CNs (highlighted in black in the figure). It is easy to see that the bipartite graph in Fig. 1 corresponds to that of an $(N, N - M) = (30, 20)$ LDPC code. If this LDPC is used for transmission over the BEC such that the channel erases the symbols corresponding to VNs $\{v_2, v_4, v_9, v_{18}, v_{19}, v_{27}, v_{29}\}$, then the corresponding residual graph on which belief propagation will be run is precisely the one highlighted in black in the figure. This link between IRSA and LDPC coding for transmission over the BEC suggests to optimize the underlying LDPC code, i.e., the degree distributions $\Lambda(x)$ and $P(x)$ can be optimized to yield the best decoding threshold over the BEC, as well as the connectivity of the graph to avoid small stopping sets. For this example, each of the 30 users is then assigned to one of the columns of the parity-check matrix of the designed LDPC code or, equivalently, to one of the VNs of the bipartite graph in Fig. 1. Note that this assignment determines also the slots in which each user will transmit through the connectivity of the graph. This deterministic assignment allows to avoid small stopping sets. For the example, the girth of the graph in Fig. 1 is 6 and hence stopping sets of size 2 are avoided. Therefore, one would expect that G-IRSA yields much lower error floors, as demonstrated in Section V.

IV. SYSTEM ANALYSIS FOR INCOMPLETE GRAPH ASSIGNMENT

In the previous section we explained how the design of an IRSA access scheme for a population of users with maximum size N may be tackled as the design of a high-rate LDPC code. Since users may join the population and leave it, the actual population size at the beginning of each frame is $N_p = N - L$, where the random variable (RV) $L \geq 0$ represents the number of users that have left the population or, equivalently, the number of columns of the LDPC code parity-check matrix

\mathbf{H} that are currently not assigned to any user. Due to the one-to-one correspondence between active users and erasures, from a coding viewpoint the presence of L unassigned resources is equivalent to L “frozen” bits in the LDPC codeword which cannot be punctured by the erasure channel. From another perspective this is equivalent to the removal of L columns from \mathbf{H} , which may then be thought as a matrix of size $M \times (N - L)$.

The dynamic behavior of the users population size N_P legitimately raises some concerns about the effectiveness of the system design proposed in Section III. In fact, since the LDPC code design is tailored to the optimization of the system performance when all N columns of \mathbf{H} have been assigned to users (equivalently, when N_P reaches its maximum value N), it becomes of fundamental importance to assess the system behavior when N_P deviates from N . There are, in particular, two events that deserve attention:

- A number ℓ_1 of new users join the population and available resources are assigned to them;
- A number ℓ_2 of users logged in the system leave the population and release their resources.

Note that while in the former case the resources to be assigned to the new users may be scheduled based on some algorithm, in the latter we have no control of the resources being released. In terms of coding, in the former case we have the freedom to choose which subset of ℓ_1 columns, out of a set of available ones, should be added to the parity-check matrix to obtain a higher-rate code with the best possible performance. In contrast, in the latter case ℓ_2 columns “disappear” from the \mathbf{H} matrix with no possibility to select them. As resource releasing by departing users represents a critical event, potentially able to compromise the system performance, in the following we propose a strategy to mitigate its negative effects.

A. Worst-Case Analysis for Released Resources

The approach we follow to prevent released resources from jeopardizing the performance of the G-IRSA system consists of analyzing the *asymptotic worst case*. Assuming that the population size is $N_P = N - L$, with $L > 0$, by asymptotic worst case we mean the choice of the L VNs to be removed from the graph $\mathcal{G}_{\text{LDPC}}$ (together with all edges connected to them) yielding a residual degree distribution pair characterized by the lowest threshold G^* . It represents the worst degree distribution pair the system may inherit when, starting from a state where the population size is N , users keep abandoning the system until L users have logged out. The concept of worst case is defined in terms of degree distribution pair and asymptotic threshold (to be calculated via DE) since a definition based on the finite frame length performance would turn out to be computationally intractable.

The worst case analysis represents a very simple approach to analyze system robustness to uncontrolled resource releasing and to understand what is the maximum tolerable number of departing users after which some action should be taken by a system controller. A possible strategy consists of setting a minimum acceptable threshold G_{\min}^* and of identifying the largest value of L , denoted by \hat{L} , such that the corresponding worst case fulfills $\hat{G}^* > G_{\min}^*$. As long as $L \leq \hat{L}$, no action is

taken when users leave the population. On the other hand, whenever some resources are released yielding $L > \hat{L}$, a scheduler may reassign some of the available resources to some of the users in the population to “rebalance” the VN degree distribution.²

To identify the worst case for a given starting distribution $\Lambda(x)$, a given maximum population size N , and a given number of released resources L , we proceed as follows. We start by calculating all possible VN degree distributions that are compatible with the given input parameters $\Lambda(x)$, N , and L . To this aim, let $\tilde{\mathbf{n}} = (\tilde{n}_2, \tilde{n}_3, \dots, \tilde{n}_{d_{\max}})$, where for $d \in \{2, 3, \dots, d_{\max}\}$ \tilde{n}_d represent the residual number of VNs of degree d after L VNs have been removed from $\mathcal{G}_{\text{LDPC}}$. Any vector $\tilde{\mathbf{n}}$ must be a solution of the Diophantine equation

$$\sum_{d=2}^{d_{\max}} \tilde{n}_d = N - L \quad (1)$$

subject to the constraints

$$0 \leq \tilde{n}_d \leq \Lambda_d N \quad \text{for all } d \in \{2, 3, \dots, d_{\max}\}. \quad (2)$$

A VN degree distribution corresponds to each solution $\tilde{\mathbf{n}}$ of (1) fulfilling all constraints in (2). In the sequel, we will denote this distribution by $\tilde{\Lambda}(x) = \sum_{d=2}^{d_{\max}} \tilde{\Lambda}_d x^d$, where for all $d \in \{2, 3, \dots, d_{\max}\}$ we have

$$\tilde{\Lambda}_d = \frac{\tilde{n}_d}{N - L}. \quad (3)$$

Each polynomial $\tilde{\Lambda}(x)$ represents one of the possible VN degree distributions we may obtain after L users have departed from the population. Associated with $\tilde{\Lambda}(x)$ we have a number $\Delta E(\tilde{\Lambda})$ of edges that have disappeared from the graph $\mathcal{G}_{\text{LDPC}}$, this number being expressed by $\Delta E(\tilde{\Lambda}) = E - (N - L) \sum_{d=2}^{d_{\max}} d \tilde{\Lambda}_d$, where $E = N \sum_{d=2}^{d_{\max}} d \Lambda_d$ is the number of edges in $\mathcal{G}_{\text{LDPC}}$. The search for the worst case may be implemented following the two different approaches described in the following.

1) *Exhaustive CN distribution search*: The first approach consists of performing, for each residual VN distribution $\tilde{\Lambda}(x)$, an exhaustive enumeration of all CN distributions $\tilde{P}(x)$ that are consistent with the initial distribution $P(x)$ and with the number $\Delta E(\tilde{\Lambda})$ of removed edges. Operatively one should proceed as follows. Let $m_r = P_r m$ and \tilde{m}_r be the number of degree- r CNs before and after removal of the $\Delta E(\tilde{\Lambda})$ edges, respectively. The set of all distributions $\tilde{P}(x)$ may be obtained by first finding all solutions $\tilde{\mathbf{m}} = (\tilde{m}_0, \tilde{m}_1, \dots, \tilde{m}_{r_{\max}})$ of the system of simultaneous Diophantine equations

$$\sum_{r=0}^{r_{\max}} \tilde{m}_r = M \quad \text{and} \quad \sum_{r=1}^{r_{\max}} r \tilde{m}_r = E - \Delta E(\tilde{\Lambda}) \quad (4)$$

subject to $\tilde{m}_r \geq 0$ for all r , and then by setting

$$\tilde{P}_r = \frac{\tilde{m}_r}{M} \quad (5)$$

²In this paper we do not address this scheduling algorithm. We point out, however, that a similar problem has been considered in [11] to design rate-compatible LDPC codes.

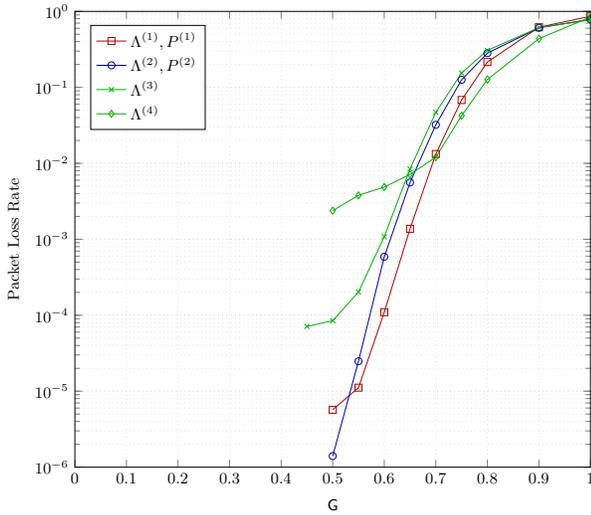


Fig. 3. PLR versus the load G for a G-IRSA scheme with degree distribution pair $(\Lambda^{(1)}, P^{(1)})$, a regular G-IRSA scheme with degree distribution pair $(\Lambda^{(2)}, P^{(2)})$, a standard CRDSA scheme with packet repetition factor 3, and a standard IRSA scheme with packet replicas distribution $\Lambda^{(4)}$. Population size $N = 2000$, frame length $M = 200$.

for each solution \tilde{m} .³ Out of all obtained pairs $(\tilde{\Lambda}(x), \tilde{P}(x))$, where $\tilde{\Lambda}(x)$ is obtained by (3) after solving (1) and $\tilde{P}(x)$ by (5) after solving (4), the pair characterized by the smallest threshold G^* represents the sought worst case.

2) *Average CN distribution*: The second approach consists of developing, for each residual VN distribution $\tilde{\Lambda}(x)$, a unique average residual CN distribution $\tilde{P}(x)$. This approach consists of first calculating an approximated average probability $\varrho(k)$ that the generic CN loses k edges when $\Delta E(\tilde{\Lambda})$ edges are removed from the graph, and then combining this probability with the initial distribution $P(x)$ to obtain the residual CN distribution $\tilde{P}(x)$. The calculation of $\varrho(k)$ relies on two simplifying assumptions. The first assumption is that, when a user leaves the population, the corresponding edges are detached from CNs chosen uniformly at random. The second one is that up to L edges may be detached from each CN.

Under these assumptions, for a given CN let X_i , $i \in \{1, 2, \dots, L\}$, be a Bernoulli RV which takes value 1 if the CN degree is decreased by one when the i -th departing user logs out and takes the value 0 if it does not. The L RVs are independent and identically distributed with $\Pr\{X_i = 1\} = \Delta E(\tilde{\Lambda})/L/M$. This is because the average number of edges removed per departing user is $\Delta E(\tilde{\Lambda})/L$ and there are M CNs. For each $k \in \{0, 1, \dots, L\}$, the probability $\varrho(k)$ is given by

$$\varrho(k) = \binom{L}{k} \left(\frac{\Delta E(\tilde{\Lambda})/L}{M} \right)^k \left(1 - \frac{\Delta E(\tilde{\Lambda})/L}{M} \right)^{L-k}.$$

The probability distribution ϱ , obtained under the simplifying hypotheses mentioned above, is now exploited to develop the residual CN distribution $\tilde{P}(x)$. To calculate the residual fraction \tilde{P}_r of CNs of degree r , we proceed as follows. Let $\mathcal{R} = \{r : P_r > 0\}$ be the set of nonzero CN degrees in the original distribution $P(x)$. Moreover, let the elements of \mathcal{R} be $r_1 = r_{\max}, r_2, \dots, r_{|\mathcal{R}|}$ such that $r_1 > r_2 > \dots > r_{|\mathcal{R}|}$. We initialize a variable c as $c = r_1$, the largest degree in \mathcal{R} .

³All integers \tilde{m}_r depend on $\tilde{\Lambda}(x)$, although not explicitly indicated.

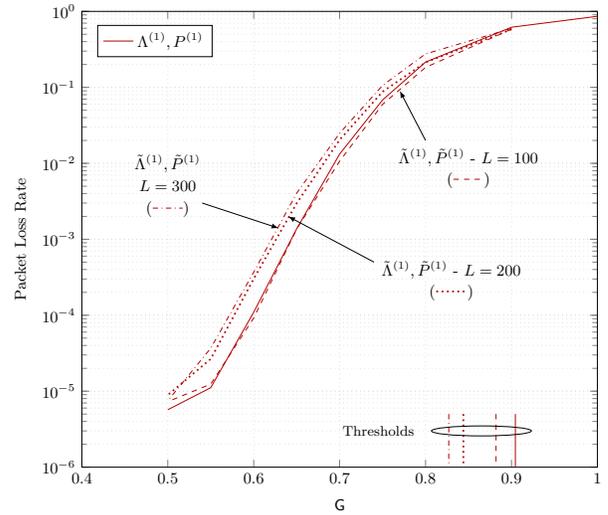


Fig. 4. PLR versus the load G for regular and irregular G-IRSA schemes in an incomplete resource assignment regime. Population size $N = 2000$, frame length $M = 200$.

Furthermore, for each $r \in \{0, 1, \dots, c\}$ we initialize \tilde{m}_r as $\tilde{m}_r = 0$. Next, for all $r \in \{0, 1, \dots, c\}$ we increase \tilde{m}_r by a quantity given by

$$\Delta \tilde{m}_r = m_c \frac{\varrho(c-r)}{\sum_{j=0}^c \varrho(j)} \quad (6)$$

i.e., the original number of CNs of degree c multiplied by the probability that a CN of degree c loses $c - r$ edges. This latter probability is obtained by truncating and renormalizing the probability distribution ϱ to account for the degree c of the CNs being processed. After all \tilde{m}_r variables have been updated, c is updated as $c = r_2$, the increments $\Delta \tilde{m}_r$ are calculated again based on (6) and, based on them, the variables \tilde{m}_r are updated again. The procedure terminates after the case $c = r_{|\mathcal{R}|}$ has been processed. At the end of the procedure the residual CN distribution is constructed according to (5). Out of all obtained pairs $(\tilde{\Lambda}(x), \tilde{P}(x))$, the one characterized by the smallest threshold G^* is returned as the worst case.

Although the approach based on the exhaustive search is more effective in identifying the degree distribution pair characterized by the lowest asymptotic threshold G^* , the second approach is much more efficient to implement as it avoids the enumeration of all possible CN distributions $\tilde{P}(x)$ for a given VN distribution $\tilde{\Lambda}(x)$. Moreover, although it relies on approximations, the second approach returns reliable results, as illustrated in the following example.

Example 2. Consider the case $\Lambda(x) = x^3$ and $P(x) = x^{30}$, corresponding to an LDPC code with rate $R = 9/10$ and whose VNs are all of degree 3. In the case $L = 0$ (i.e., when all resources have been assigned) the system threshold is $G^* = 0.82833$. Since all users have the same repetition factor, even if $L > 0$ users leave the system, we expect very small G^* variations. Next we show how the results obtained by applying the average CN distribution approach are coherent with this expectation. To develop the example, we choose $N = 2000$ and $M = 200$. In this case there is only one solution to (1) for any L , namely, $\tilde{n}_3 = N - L$, and we always have $\tilde{\Lambda}(x) =$

x^3 . Let $L = 100$, so $\Delta E(\tilde{\Lambda}) = 300$ and $\Delta E(\tilde{\Lambda})/L/M = 0.015$. We have $\mathcal{R} = \{30\}$, hence (6) is computed only for $c = r_{\max} = 30$. We obtain $\tilde{P}(x) = 0.001x^{23} + 0.003x^{24} + 0.014x^{25} + 0.047x^{26} + 0.126x^{27} + 0.253x^{28} + 0.336x^{29} + 0.221x^{30}$, where all fractions are rounded to three decimals. The corresponding threshold is $G^* = 0.82831$. For $L = 200$ and $L = 300$ we obtain $G^* = 0.82822$ and $G^* = 0.82803$, respectively.

V. NUMERICAL RESULTS

To assess the improvement in terms of PLR (especially in the error floor region) provided by G-IRSA with respect to standard IRSA, we designed the degree distribution

$$\begin{aligned}\Lambda^{(1)}(x) &= 0.100x^2 + 0.781x^3 + 0.119x^{12} \\ P^{(1)}(x) &= 0.126x^{38} + 0.076x^{39} + 0.797x^{40}\end{aligned}$$

characterized by a threshold $G^* = 0.904$. The distribution pair was designed by placing a constraint on the rate ($R = 9/10$), on the maximum VN degree ($d_{\max} = 12$), and on the fraction of degree-2 VNs to not exceed the ratio M/N , this latter constraint being often adopted in irregular LDPC code design. The graph $\mathcal{G}_{\text{LDPC}}$ was constructed for a maximum user population size $N = 2000$, and for a MAC frame length $M = 200$. The graph hence possesses 2000 VNs and 200 CNs. The PEG algorithm was used to draw the edge connections in order to limit the number of short cycles. The performance of the G-IRSA scheme based on the so-obtained graph was compared with those achieved by other graphs having the same number of VNs and CNs. In particular, the following degree distributions were considered:

- a G-IRSA scheme with regular distribution $\Lambda^{(2)}(x) = x^3$ and $P^{(2)}(x) = x^{30}$, with threshold $G^* = 0.828$;
- a standard CRDSA scheme with repetition-3, i.e., $\Lambda^{(3)}(x) = x^3$, with threshold $G^* = 0.818$;
- a standard IRSA scheme from [5] with $\Lambda^{(4)}(x) = 0.50x^2 + 0.28x^3 + 0.22x^8$, with threshold $G^* = 0.938$.

The PLR of the four random access schemes under SIC-based decoding is shown in Fig. 3 as a function of the system load G . For both the irregular and the regular G-IRSA schemes, the PLR reported in Fig. 3 is the one corresponding to $L = 0$, i.e., to the case where all columns of the 200×2000 matrix \mathbf{H} are assigned to users. We observe an impressive improvement of the error floor of the PLR both in the irregular case and in the regular one. Note that the slightly better performance of the standard IRSA scheme with respect to the irregular G-IRSA for PLRs above 10^{-2} (a region of limited interest) is due to its slightly better threshold. For the regular G-IRSA scheme based on the pair $(\Lambda^{(2)}, P^{(2)})$, the application of the PEG algorithm allows enforcing a girth equal to 6 in the graph $\mathcal{G}_{\text{LDPC}}$, resulting in a very low error floor and no loss in waterfall performance.

While Fig. 3 addresses the case in which all system resources have been assigned to users, Fig. 4 illustrates PLR curves corresponding to worst cases identified with the average CN distribution method described in Section IV. The analysis was conducted for the G-IRSA scheme with the degree distribution pair $(\Lambda^{(1)}, P^{(1)})$, for $L \in \{100, 200, 300\}$. The relevant worst case distributions and thresholds are reported

TABLE I
WORST-CASE THRESHOLD DEGRADATION DUE TO INCOMPLETE GRAPH ASSIGNMENT, FOR THE DEGREE DISTRIBUTION PAIR $\Lambda^{(1)}, P^{(1)}$

| L | Worst-case VN d.d. | |
|-----|-------------------------------------|-------|
| | $\tilde{\Lambda}^{(1)}(x)$ | G^* |
| 100 | $0.105x^2 + 0.823x^3 + 0.072x^{12}$ | 0.882 |
| 200 | $0.111x^2 + 0.869x^3 + 0.020x^{12}$ | 0.845 |
| 300 | $0.080x^2 + 0.920x^3$ | 0.827 |

in Table I. Coherently with the asymptotic analysis, a gap arises between the PLR curve corresponding to a complete resource assignment and the ones corresponding to worst cases in incomplete resource assignment conditions. Notably, even for $L = 300$ (corresponding to 15% of the users leaving the system with no action taken by a scheduler) the penalty is very limited both in the waterfall and error floor regions.

VI. CONCLUSIONS

We have presented G-IRSA, an uncoordinated access protocol based on user packet repetitions and SIC decoding, which exhibits a better performance than standard IRSA in terms of PLR, especially in the error floor region. The main features of G-IRSA are represented by a full control not only of the distribution according to which users select their repetition factors (as for standard IRSA) but also of the distribution determining the number of packet replicas transmitted per slot and of the connectivity of the underlying graph. This is made possible by the introduction of a login procedure enabling a finer control of the users access activity.

REFERENCES

- [1] N. Abramson, "The ALOHA system - Another alternative for computer communications," in *Proc. of the 1970 Fall Joint Comput. Conf., AFIPS Conf.*, vol. 37. AFIPS Press, 1970, pp. 281–285.
- [2] D. Raychaudhuri, "ALOHA with multipacket messages and ARQ-type retransmission protocols – Throughput analysis," *IEEE Trans. Commun.*, vol. 32, no. 2, pp. 148–154, Feb. 1984.
- [3] G. Choudhury and S. Rappaport, "Diversity ALOHA - A random access scheme for satellite communications," *IEEE Trans. Commun.*, vol. 31, no. 3, pp. 450–457, Mar. 1983.
- [4] E. Casini, R. De 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.
- [5] 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.
- [6] 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.
- [7] E. Paolini, "Finite length analysis of irregular repetition slotted ALOHA (IRSA) access protocols," in *Proc. IEEE Int. Conf. Commun. Workshop*, London, UK, Jun. 2015.
- [8] M. Ivanov, F. Brännström, A. Graell i Amat, and P. Popovski, "Broadcast coded slotted ALOHA: A finite frame length analysis," *IEEE Trans. Commun.*, vol. 65, no. 2, pp. 651–662, Feb. 2017.
- [9] X. Hu, E. Eleftheriou, and D. Arnold, "Regular and Irregular Progressive Edge-Growth Tanner Graphs," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 386–398, Jan. 2005.
- [10] A. Orłitsky, R. Urbanke, K. Viswanathan, and J. Zhang, "Stopping sets and the girth of Tanner graphs," in *Proc. IEEE Int. Symp. Inf. Theory*, Lausanne, Switzerland, Jun. 2002.
- [11] T. Tian and C. Jones, "Construction of rate-compatible LDPC codes utilizing information shortening and parity puncturing," *EURASIP J. Wireless Commun. Netw.*, vol. 2005, no. 5, pp. 789–795, 2005.