On the capacity of a random access channel with successive interference cancellation

G.T. Peeters and B. Van Houdt
University of Antwerp - iMinds, Middelheimlaan 1, B-2020 Antwerp, Belgium

***Abstract*—Successive interference cancellation (SIC), which allows one to recover additional information from otherwise lost collision signals, has been proposed in a variety of communication systems. In this paper, we introduce a random access protocol operating on channel supporting SIC that achieves a maximum stable throughput (MST) of 1. To the best of our knowledge, the highest MST achieved on such a channel thus far was 0.693 by the SICTA algorithm of Yu and Giannakis [1]. As opposed to the SICTA algorithm, the newly proposed algorithm has very limited practical value due to its unbounded computational complexity. That is, the computation time needed per time slot is finite, but unbounded over all time slots.**

To bound the computation time, we also propose a hybrid algorithm, which combines the new algorithm with SICTA. We demonstrate that the MST of this hybrid algorithm is $1-\epsilon$, where $\epsilon > 0$ can be chosen arbitrarily small given sufficient computational power.

***Index Terms*—Random-access, maximum stable throughput, successive interference cancellation.**

I. Introduction

The maximum stable throughput (MST) of a random access algorithm is defined as the highest possible (Poisson) input rate λ for which a packet has a finite delay with probability one, i.e., there exists a stationary packet delay distribution (typically with a finite mean). This form of stability is equivalent to stating that the rate of successful transmissions is equal to the input rate [2]. The first algorithms to have a provable MST above zero, known as tree algorithms, were independently developed in the late 1970s by Capetanakis [3] and Tsybakov, Mikhailov and Vvedenskaya [4]. Tree algorithms resolve conflicts by recursively splitting the group of users transmitting simultaneously in one slot, into two separate groups. Each of those groups then retransmits into its designated slot, until eventually every user has its own slot, and thus can be correctly received. Afterwards improved tree algorithms were developed with MSTs as high as 0.4878 under the standard information theoretical multiple access model [5], [2], [6], [7]. Tsybakov and Likhanov [8] showed that no protocol can be constructed which is stable for an input rate λ over 0.568, under the standard model.

The 0.4878 MST has been exceeded in various manners by introducing additional mechanisms not available under the standard model. For example, an additional control field with separate feedback [9] could provide an MST of $0.56/(1+r)$ with r the ratio of the length of the control field over the packet length. Another technique to improve the MST is energy measurement [10]. If such a technique allows collision multiplicity feedback, Pippenger [11] showed that the channel has capacity 1, i.e., a protocol exists that achieves an MST of 1. Later, Ruszinko and Vanroose [12] confirmed this result by constructing such a protocol explicitly.

More recently the successive interference cancellation tree algorithm (SICTA), which combines tree algorithms with successive interference cancellation (SIC) mechanisms, was designed and shown to achieve an MST as high as 0.693 [1]. SIC has the ability to extract additional information from collisions, by "subtracting" one signal from another. As such, whenever a group of users is split into two smaller groups, the interference cancellation (IC) operation only requires the actual transmission of one group. The signal of the second group can be directly determined by subtracting the signal of the first group from the signal of the original, combined group. This original signal could have been transmitted previously, or was also reconstructed by previous SIC operations. As tree algorithms heavily rely on this operation, resolving collisions requires a significantly lower number of slots. Under the standard assumption of Poisson arrivals, the number of new arrivals that can occur in a slot is unbounded. This implies that the maximum number of users involved in a collision under SICTA is also unbounded and therefore SICTA, in its original form, requires an unbounded amount of memory for storing signals. However, a finite number of memory locations suffices to approximate the MST of 0.693 closely, see [13].

The SIC mechanism has also been proposed in the context of satellite communication systems [14]. In such a system, a packet is transmitted in several random positions of a frame possibly causing additional collisions, but the SIC mechanism is subsequently used to retrieve packets from such collisions resulting in an overall higher throughput. In case of a finite user population one can further improve the throughput by relying on combinatorial designs to select the positions within a frame [15], [16].

A frameless approach for distributed random access in a slotted ALOHA framework was presented in [17], [18]. In fact, the algorithm discussed in [17] resembles the one proposed in this paper as users in a contention period also transmit their packet with a fixed probability p_a (the value of which is broadcast by the base station based on an estimate of the number of users contending). The manner in which the packets are subsequently recovered is however very different from our approach.

To the best of our knowledge determining the capacity of a SIC random access channel is still an open problem [19]. In this paper, we describe a random access protocol operating on channel similar to the one used by SICTA (there are some additional assumptions), achieving an MST of 1. The time complexity of this protocol renders it completely useless for practical purposes. More precisely, for any specific time slot t there exists a c_t that bounds the amount of work required in time slot t, however, the set of c_t values is unbounded. However, by combining this theoretical algorithm with SICTA, we can approximate the MST of 1 arbitrarily close even if the computational complexity per time slot is bounded.

The main idea of our approach is to resolve a size N conflict by collecting enough ($\geq N$) linear combinations of these N packets, such that these N individual packets can be extracted by identifying a subset of N linear independent signals. In this regard, there is some resemblance with the wireless multiple access algorithms introduced in [20], [21], [22]. These algorithms are capable of resolving a conflict of K users through source separation techniques. More specifically, each of the K users retransmits its packet in every subsequent slot as long as the base station does not announce the end of the current conflict resolution period. The network diversity multiple access (NDMA) algorithm of [20] resolves the conflict in K slots, by detecting the conflict multiplicity during the very first transmission via orthogonal identification codes. Next, it waits for another $K-1$ retransmissions of the same K packets and retrieves the packets from the K transmissions using source separation techniques. The limiting use of the orthogonal codes of [20] for larger populations is avoided in [21] and [22].

The main difference with our work, is that NDMA focuses on very specific signal processing techniques and relies on induced differences between each user, prohibiting infinite populations. Our work focuses on the infinite user model, where each user is identical, except for the random choices made at each slot. Also, the packets are decoded using the abstract concept of signal subtraction/addition and in our model we assume that the conflict multiplicity cannot be detected.

The extensive, unbounded amount of computational effort per time slot in our approach stems from the unbounded number of slots required to solve a collision. The main idea to bound the amount of work is to truncate the search for linear independent signals after S slots, and fall back to the existing SICTA protocol.

The paper is organized as follows. Section II identifies the channel assumptions under which the result applies. Next, in Section III, we describe the random access protocol having an MST 1. Section IV proves that this algorithm is stable for all input rates $\lambda < 1$. Finally, Section V demonstrates how a more practical algorithm with limited computational complexity can be constructed, which can approximate the MST of 1 arbitrarily close.

II. Slotted Multiaccess Model Assumptions

The proposed algorithm and its analysis are based on the following assumptions, describing a multiaccess model with successive interference cancellation. The first series of assumptions ($S1$ to $S4$) are standard assumptions that have been used by a multitude of authors (for a detailed treatment of these assumptions we refer to [2], [6], [5]).

- $S1$. *Slotted system*: the channel is divided in fixed length time slots; each user is allowed only to start transmitting at the beginning of a time slot; all packets have the same length equal to one time slot.
- $S2$. *Error-free reception* by the receiver: a slot is either received as an idle, success or a collision slot, depending on whether zero, one or more packets are transmitted.
- $S3$. *Infinite population*: there is an infinite set of users, generating packets according to a Poisson process (where each packet corresponds to a new user)
- $S4$. *Error-free feedback*: Immediate SUCCESS/no SUCCESS feedback, where a SUCCESS indicates that all the users participating in the current conflict resolution period (CRP) are successful (see Section III for a definition of the CRP).

Note, as our random access algorithm uses gated access, there is no need to restrict ourselves to default Poisson (or Bernoulli) arrivals (see [23, Theorem 8]). For the hybrid algorithm of Section V we assume Poisson arrivals. For the feedback it suffices that the receiver issues a SUCCESS feedback signal to announce the end of the current CRP. In the absence of this SUCCESS feedback, the current CRP continues and all users behave as such. Also, the SUCCESS feedback is not even required to be immediate; as explained in Section IV, a (bounded) delayed feedback does not influence the MST. SICTA uses immediate $0/k/e$ feedback, with 0 and e representing an idle and collision slot and k represents the number of decoded packets (plus the number of left idles). This is clearly very different from the multiplicity feedback in [12] that identifies the number of users caught in a collision.

The assumptions ($I1$ to $I6$) specific to the interference cancellation mechanism are as follows:

- $I1$. Two signals can be combined by 'adding' them together. Consider two signals a and b, where a consists of the combination of packet signals $A_1 + \ldots + A_m$, and b consists of $B_1 + \ldots + B_n$. We denote $a + b$ as the addition IC operation, which results in the valid signal of $A_1 + \ldots + A_m + B_1 + \ldots + B_n$, where the sum is taken over the reals.
- $I2$. An inverted signal (i.e., $-a$) can also be used as input for an IC operation. If both a signal and its inverted variant are added, they will cancel each other out, resulting in a zero signal.
- $I3$. The output of an IC operation can be used as input for another IC operation; i.e., successive IC is supported.
- $I4$. The receiver can store an unbounded number of signals. For the hybrid algorithm of Section V, a small, finite number of memory locations typically suffice, similar to SICTA (see [13]).
- $I5$. An IC operation resulting in the zero signal (the result of $a + (-a)$) is detected as such.
- $I6$. *Given* a positive integer α and a signal S, IC can detect whether S consists of a single packet amplified by the *given* factor α or not, and if so, can decode the packet. In other words, given α IC can recover packet signal A_1 from the signal $S = \alpha A_1 = A_1 + \ldots + A_1$.

The combination of $I1$, $I2$, $I3$ and $I4$ enables IC to obtain any linear combination of the form $\alpha a + \beta b$, with α and β integers, a and b signals.

These assumption are somewhat stronger than what is required by the SICTA algorithm [1]. The IC mechanism used by SICTA only requires the ability to repeatedly subtract successfully received packets from earlier received collision signals. In other words, in SICTA the resolution of one packet may propel the resolution of additional packets via the IC mechanism, while our approach combines several signals in a brute-force approach to resolve all the packets at once.

At this point we should stress that we are considering a noiseless channel and perfect interference cancellation. If either of these assumptions is violated, significant changes may be required as for SICTA (see [24]).

We must note that the operation of our algorithm does not allow us to upper bound the required processing speed (ops per time slot) as the finite amount of work associated with a length m conflict resolution period is not a linear function of m. However, for the hybrid algorithm of Section V there exists a c that bounds the amount of work required in any time slot. Finally, we assume that the packets participating in the same conflict resolution period are unique, thus if two users were to transmit exactly the same packet, we do not regard this collision as a success. For the hybrid algorithm in Section V there are typically very few participants in a CRP, meaning the probability of having identical packets can be neglected.

III. An algorithm achieving MST 1

The channel access protocol we use is commonly known as blocked or gated access. Here, an initial collision of N stations causes all subsequent new arrivals to postpone their first transmission attempt until the N initial stations have resolved their collision. The time elapsed from the initial collision until the point where the N stations have transmitted successfully is called the conflict resolution period (CRP). In short, when the blocked access mode is used, new arrivals are blocked until the CRP during which they arrived has ended. They will participate in the next CRP.

The following simple rule is the core of our algorithm. Once a CRP has started, each user participating in the CRP, transmits its packet with probability $1/2$. This behavior is repeated for each of the subsequent slots, until a SUCCESS feedback is issued by the receiver. This feedback indicates that all packets have been successfully decoded, which terminates the CRP. For N large, this rule often results in a sequence of collisions, as the probability of having 0 or 1 transmissions in a single slot is small for N large. Nonetheless, as we will see, a brute-force-like search through all possible IC combinations on the signals received thus far may provide the receiver with enough information to decode all packets belonging to a CRP.

There is one exception to this transmission rule: at the start of each CRP we require that all users transmit their packet. The most important reason for this rule is that we need to be able to detect whether all the packets participating in the current CRP have been decoded from the transmitted signals. Otherwise, for instance if some user, having a packet ready at the start of a CRP, does not transmit its packet while enough signals have been transmitted to decode all the remaining packets, it is impossible (without any additional mechanism) to detect its presence in the current CRP, resulting in a preliminary termination of the CRP. A second, additional advantage of this rule is that the first signal in the CRP is guaranteed to be nonempty unless the CRP contains no users, meaning after the first slot, we already have one linearly independent signal available. The case where the current CRP contains no users can be easily detected by the occurrence of the empty signal in this first slot. In this case, the current CRP can be terminated and the SUCCESS feedback can be issued immediately.

Once enough signals have been transmitted in a nonempty CRP, the job of the receiver consists of decoding the individual packets out of the transmitted signals, and based on this, issue a SUCCESS feedback to the terminals. We now describe the decision and decoding process. The proposed procedure is by no means an optimal one and its time complexity clearly forbids a direct practical implementation. As we focus on the theoretical bound, it suffices that the amount of work required in each step is finite.

Consider a CRP consisting of N users, with N obviously unknown. With the i-th transmitted signal, we associate a signal column vector $\mathbf{s_i} \in \mathbb{R}^N$. The j-th entry of $\mathbf{s_i}$ is either 1 or 0, depending on the fact whether the j-th user has transmitted its packet in slot i or not. If a signal vector $\mathbf{s_i}$ contains exactly one entry equal to 1, we can associate it with a single, successfully received packet.

To describe the decoding process, suppose $m \geq N$ slots have occurred thus far in the current CRP. Further, suppose N of these m signals correspond to a set of N linearly independent signal vectors $\mathbf{s_{i_1}}, \ldots, \mathbf{s_{i_N}}$, with $1 = i_1 < i_2 < \ldots < i_N = m$. Now, define $\mathbf{M}$ as the $N \times N$ matrix having $\mathbf{s_{i_k}}$ as its k-th column. As $\mathbf{M}$ consists of linearly independent columns, $\mathbf{M}$ is invertible.

Assume for now that we somehow know N, the indices i_2 to i_{N-1} and their corresponding matrix $\mathbf{M}$ (clearly, we do not know any of these when running the algorithm) and that $\mathbf{M}^{-1}$ contains only integer numbers. In this case we can directly decode each of the N packets from the signals $\mathbf{s_{i_1}}, \ldots, \mathbf{s_{i_N}}$. For example, consider a CRP with $N = 3$ users and denote their packets as A, B and C. Suppose the following signals $\mathbf{s_{i_1}} = \mathbf{s_1} = ABC, \mathbf{s_{i_2}} = \mathbf{s_2} = AB$ and $\mathbf{s_{i_3}} = \mathbf{s_4} = AC$ are transmitted in slot $i_1 = 1$, $i_2 = 2$ and $i_3 = 4$, respectively. The corresponding matrix $\mathbf{M}$ becomes:

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

with

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

Thus, to recover the individual packets, we can combine the signals as follows: $(\mathbf{M}^{-1})_{1,1}\mathbf{s_{i_1}} + (\mathbf{M}^{-1})_{2,1}\mathbf{s_{i_2}} + (\mathbf{M}^{-1})_{3,1}\mathbf{s_{i_3}} = -ABC + AB + AC = A$, while B and C can be obtained by $ABC - AC$ and $ABC - AB$ respectively. In other words, the columns of $\mathbf{M}^{-1}$ provide us with the required coefficients of the linear combinations needed by the IC mechanism.

If the inverted matrix contains rational numbers (it cannot hold irrational ones as $\mathbf{M}$ is a binary matrix), this procedure cannot be applied directly, as taking fractions of signals is not allowed by our assumptions. However, if we multiply each element of the inverted matrix $\mathbf{M}^{-1}$ by the determinant $|\mathbf{M}|$, we see that all fractions become integers. Hence, if we use the columns of the matrix $|\mathbf{M}|\mathbf{M}^{-1}$ as the coefficients of the linear combinations used by the IC mechanism, we end up with a set of N different packets each amplified by a factor $|\mathbf{M}|$ as $\mathbf{M}(|\mathbf{M}|\mathbf{M}^{-1})$ equals $|\mathbf{M}|$ times the identity matrix. This causes no problems for our decoding process, as we assumed that a packet can be decoded if it is amplified by known factor ($I6$), by taking $|\mathbf{M}|$, a known number, as the known factor α.

To resolve the problem of the unknowns N, $i_2, \ldots, i_{N-1}$ and $\mathbf{M}$, we propose a brute-force solution. After every transmission in a CRP, we try out all possible values for each of these unknowns. As only a finite number of possibilities exists, this results in a finite amount of work required at the end of each time slot. Suppose, the m-th slot of a CRP just occurred. The N values that we need to explore are $N = 2, \ldots, m$ (as $N = 1$ would be detected in the very first slot). To determine the indices $i_2, \ldots, i_{N-1}$, for each $N = 2$ to m, we try all possible subsets of size $N-2$ out of $m-2$. For each N value and set of indices i_1 to i_N, we need to construct the matrix $|\mathbf{M}| \cdot \mathbf{M}^{-1}$ in some manner. As it seems impossible to retrieve $\mathbf{M}$ from the signals $\mathbf{s_{i_1}}, \ldots, \mathbf{s_{i_N}}$, we simply try all possible invertible, binary, size N matrices $\mathbf{M}$ as well (their number is also finite in m). Note, the computational effort to generate all possible N, $i_2, \ldots, i_{N-1}$ indices and $\mathbf{M}$ matrices is therefore finite in m. By forming the N linear combinations that correspond to the N columns of $|\mathbf{M}| \cdot \mathbf{M}^{-1}$, we can check whether a set of N packets can be decoded. If this never occurs for all possible $N = 2, \ldots, m$ values, $i_2, \ldots, i_{N-1} \subseteq \{2, \ldots, m-1\}$ indices and $\mathbf{M}$ matrices, we do not issue the SUCCESS feedback.

Remark, decoding the signals $\mathbf{s_{i_1}}, \ldots, \mathbf{s_{i_N}}$ into a series of N different packets is sufficient to issue a SUCCESS message and end the CRP, if the combination of all the decoded signals matches the signal $\mathbf{s_{i_1}} = \mathbf{s_1}$ (which contains all the user packets). Indeed, multiple solutions for $i_2, \ldots, i_{N-1}$ and $\mathbf{M}$ can exist, but still lead to the same N distinguishable packet signals. For instance, if $\mathbf{P}$ is any permutation matrix, then $\mathbf{PMP}^t$ (with $\mathbf{P}^t$ the transposed of $\mathbf{P}$) results in the same set of N signals (also amplified by $|\mathbf{M}|$ as $|\mathbf{P}| = |\mathbf{P}^t| = \pm 1$), but the signals are in a different order.

IV. Performance analysis

To analyze the MST of a blocked access random access algorithm, we first need to determine the mean time L_N to resolve a collision of N initial colliders. As stated earlier the i-th slot can be associated with a signal vector $\mathbf{s_i}$. Therefore, we only require that N linearly independent signals vectors $\mathbf{s_{i_1}}, \ldots, \mathbf{s_{i_N}}$ have been transmitted. After this, the corresponding matrix $\mathbf{M}$ is found by our brute-force procedure and the current CRP ends.

First, we determine the probability p_k of having a new, linearly independent signal vector, given that $0 < k < N$ linearly independent signal vectors $\mathbf{s_{i_1}}, \ldots, \mathbf{s_{i_k}}$ were already transmitted in the current CRP:

$$p_k \geq \frac{2^N - 2^k}{2^N} = 1 - 2^{k-N}, \tag{1}$$

as out of the 2^N possible signal vectors, we can construct at most 2^k signal vectors as linear combinations of the k linearly independent signal vectors $\mathbf{s_{i_1}}, \ldots, \mathbf{s_{i_k}}$ (as any k-dimensional subspace of $\mathbb{R}^N$ intersects the corner points of the N-dimensional cube formed by the 2^N binary vectors in $\mathbb{R}^N$ in at most 2^k points). Since each station transmits in every slot with probability $1/2$, each vector is equally likely to occur, yielding the formula above.

It is worth noting that over the Galois field GF(2) one can construct exactly 2^k binary vectors as linear combinations of $\mathbf{s_{i_1}}, \ldots, \mathbf{s_{i_k}}$, meaning $1-2^{k-N}$ would be the exact probability to find another independent signal vector (allowing one to determine the exact probability that a binary matrix is of full rank [25]). Over the field of the real numbers fewer binary vectors may exist, e.g., if $\mathbf{s_{i_1}} = (1,1,0)$ and $\mathbf{s_{i_2}} = (0,1,1)$, then $(1,0,1)$ is a linear combination of $\mathbf{s_{i_1}}$ and $\mathbf{s_{i_2}}$ over GF(2), but not over the real numbers. For our purpose independence over the real numbers suffices as we only require that $\mathbf{M}$ has a real inverse.

It takes $1/p_k$ slots on average to go from $k > 0$ linearly independent vectors to $k+1$; the first linearly independent vector is already found in the first slot, as we require all stations to transmit in this first slot. As such, we find that L_N can be bounded as follows:

$$L_N \leq 1 + \sum_{k=1}^{N-1} \frac{1}{1 - 2^{k-N}} = N + \sum_{j=1}^{N-1} \frac{1}{2^j - 1}. \tag{2}$$

To demonstrate the impact of the upper bound on L_N, we have plotted both N/L_N and the resulting lower bound on N/L_N in Fig. 1. The actual values for N/L_N were obtained by simulations; for each $0 < N \leq 25$, 10^6 CRPs with randomly chosen signal vectors were simulated. We observe that our algorithm performs even significantly better than the proposed lower bound, except for $N < 3$, where both values coincide.

To determine the MST for our blocked access protocol, we need to analyze N/L_N, as N tends to infinity. We first observe that

$$\lim_{N \to +\infty} \sum_{j=1}^{N-1} \frac{1}{2^j - 1} = E_b = 1.606695\ldots, \tag{3}$$

with E_b the Erdős-Borwein constant [26]. Notice, this is a surprising result, as it indicates that the average length of a CRP containing N users is upper bounded by $N + 1.606695$ slots. As such,

$$1 \geq \liminf_{N \to +\infty} \frac{N}{L_N} \geq \lim_{N \to +\infty} \frac{N}{N + \sum_{j=1}^{N-1} \frac{1}{2^j - 1}} = 1. \tag{4}$$

This proves that the algorithm is stable for any $\lambda < 1$, as $\lambda < 1 = \liminf_{N \to +\infty} \frac{N}{L_N}$ [23]. Furthermore, an additional (bounded) delay on the SUCCESS feedback does not influence the MST, as it becomes negligible compared to N when N becomes large.

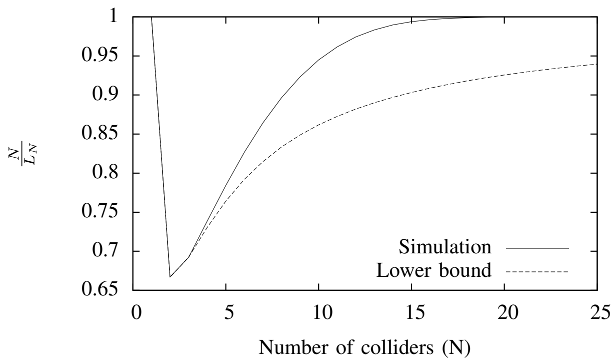

Fig. 1. Illustration of the efficiency of resolving CRPs, as a function of the number of initial colliders N.

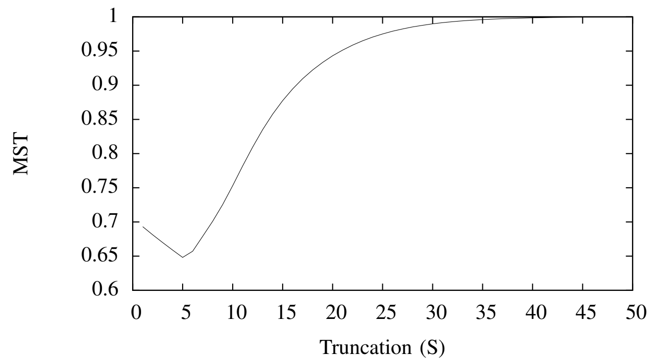

Fig. 2. The MST of the hybrid tree algorithm as a function of S.

V. Hybrid algorithm with limited time complexity

In this section we introduce a hybrid tree algorithm, by combining the algorithm of Section III with the SICTA algorithm [1]. Using this hybrid algorithm, we demonstrate how an of MST of $1-\epsilon$ can be achieved if the time complexity per time slot is bounded by some c, where $\epsilon > 0$ can be chosen arbitrarily small for c large.

Consider a CRP with N users. The idea is to first run the algorithm as described in Section III for at most a fixed number of slots S. If after $s \leq S$ slots N linearly independent combinations were transmitted, the CRP can be ended after s slots, as discussed in Section III. Otherwise, we switch to SICTA after S time slots and solve the collision of the N users using SICTA. During this second phase we also make use of the $0/k/e$ feedback of SICTA. Note that we can skip the first slot of the SICTA operation, as it is identical to the first slot of the entire CRP: under both algorithms all the users involved in the current CRP transmit in the first slot.

By using this approach, we limit the computational complexity, as the number of possible linear combinations is limited to the finite set formed by these S slots, thus a c that bounds the amount of work for any time slot can be determined (though its value depends on S). We note that the number of signal memory locations required during the first phase is bounded by S (see assumption $I4$). However, as the number of users in a CRP is unbounded, the number of signal memory locations needed for the possible second phase of the CRP (i.e., SICTA) is in principle still unbounded, but as remarked before a limited number of memory locations suffice to approximate its MST arbitrarily close [13].

We propose to run this hybrid tree algorithm in windowed access, as opposed to Section III that relied on gated access. Running this algorithm in gated access would rely on its efficiency for solving very large groups of colliding users. Such large groups would require more than S slots, which implies that a switch to SICTA occurs in such a CRP and the hybrid algorithm simply wastes $S-1$ slots. Windowed access typically produces smaller conflicts, because only users that arrived during a specific size α_0 interval, called the allocation window, may take part in a CRP. Once this CRP is resolved, the next size α_0 interval is resolved. A more detailed description of windowed access can be found in [5].

If we define $\bar{\bar{L}}_N$ as the mean number of slots SICTA requires to resolve a collision of N users [19], we can define $\bar{L}_N$ as the mean number of slots required by our hybrid tree algorithm to resolve N users:

$$\bar{L}_N = \sum_{i=1}^{S} i F_{N,i} + (S + \bar{\bar{L}}_N - 1)(1 - \sum_{i=1}^{S} F_{N,i}),$$

where $F_{N,i}$ is defined as the probability that a collision of N users can be solved in exactly i slots, given that all users transmit in the first slot. Or in other words, $F_{N,i}$ holds the probability that i vectors of N randomly chosen binary numbers (except for the first vector, which contains all ones) form an N dimensional vector space, whereas removing the last vector spans only an $N-1$ dimensional vector space. We obtain the probabilities $F_{N,i}$ by simulating 10^6 CRPs. In case more than S slots are required to transmit N linearly independant signals, which occurs with probability $1-\sum_{i=1}^{S} F_{N,i}$, we rely on SICTA, which requires $\bar{\bar{L}}_N$ additional slots, minus one because we can reuse the signal in the first of the S slots for SICTA.

In order to have a stable system, the average length $\bar{L}$ of a CRP must be less than the average distance that the starting point of the allocation window advances between two windows, which equals α_0 (this is due to the Lemma of Pakes [5]). By multiplying both sides with λ, we can rewrite this as

$$\lambda < \frac{\lambda \alpha_0}{\bar{L}},$$

where the right hand side of this equation is a function f of $\lambda\alpha_0$. By numerically maximizing this function, denoting x_{max} as the point in which the maximum is reached and $f(x_{max})$ as its maximum value, we obtain the highest possible maximum stable throughput $\lambda_{max} = f(x_{max})$ by setting $\alpha_0 = x_{max}/f(x_{max})$. Notice, $\bar{L} = \sum_{N \geq 0} \bar{L}_N b_N$, where b_N is the probability that N users participate in a CRP. Hence, for Poisson arrivals $b_N = (\lambda\alpha_0)^N / N! \exp(-\lambda\alpha_0)$.

Figure 2 shows the MST for various truncation points S. $S = 1$ implies an identical operation as SICTA and thus the MST is equal to 0.693, whereas slightly higher truncation points S have a negative effect on the MST. The reason for this is that small $S > 1$ values have a very low probability of solving the collision in the first phase of the CRP. For instance when $S = 2$, the CRP only ends after $S = 2$ slots if $N = 2$

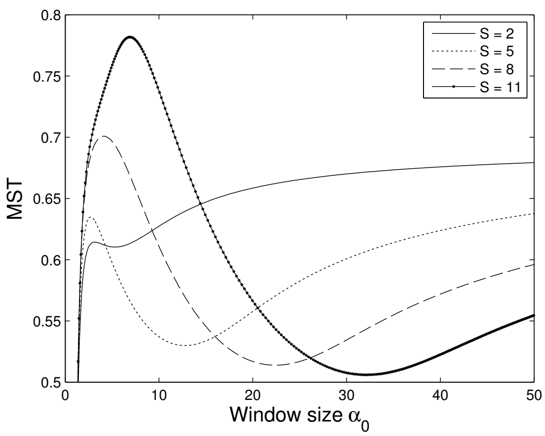

Fig. 3. MST of the hybrid algorithm as a function of the window size α_0 for various S values.

and exactly one user retransmits in the second slot. When the collision cannot be solved in the first S slots, $S-1$ slots are wasted (the first is used by SICTA), resulting in a lower MST. When $S \geq 8$, the gain of the first phase outweighs its cost and the resulting MST exceeds 0.693; at $S = 8$ the MST is close to 0.7. For larger S values, the MST approaches 1 rather quickly. In fact, the optimal window size α_0 for $S \leq 7$ is ∞ (see Figure 3), which corresponds to gated access, and the resulting MST is equal to the one of SICTA (as the $S-1$ wasted slots do not affect the MST under gated access). The reason why we observe an MST below 0.693 for $S = 2$ to 7 in Figure 2 is simply because we limited the search for the optimal x_{max} to the interval $[0, 40]$.

To conclude, we remark that this algorithm can be further improved, by reusing the second slot of the CRP as the second slot of SICTA. To understand this, we note that all users operate in a similar manner during the second slot: all the users transmit with probability $1/2$, which corresponds in SICTA to choosing either the left or right branch (if $p = 1/2$). Determining the MST of this improvement turns out to be less obvious and the S value for which SICTA is outperformed does not decrease below 8.

VI. Conclusion

In this paper, we described a novel random access algorithm, which is shown to achieve an MST of 1 on a successive interference cancellation channel. Because of its high complexity a practical implementation is unlikely. We also demonstrated how under bounded time and signal memory requirements, a hybrid algorithm can be constructed which approximates an MST of 1 arbitrarily close.

References

[1] Y. Yu and G. B. Giannakis, "SICTA: a 0.693 contention tree algorithm using successive interference cancellation." in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies, Miami (USA)*, March 2005, pp. 1908–1916.

[2] A. Ephremides and B. Hajek, "Information theory and communication networks: an unconsummated union," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2416–2434, October 1998.

[3] J. Capetanakis, "Tree algorithms for packet broadcast channels," *IEEE Trans. Inf. Theory*, vol. 25, no. 5, pp. 319–329, 1979.

[4] B. S. Tsybakov and V. Mikhailov, "Free synchronous packet access in a broadcast channel with feedback," *Problemy Peredachi Informatsii*, vol. 14, no. 4, pp. 32–59, 1978.

[5] D. Bertsekas and R. Gallager, *Data Networks*. Prentice-Hall Int., Inc., 1992.

[6] G. Polyzos and M. Molle, "Performance analysis of finite nonhomogeneous population tree conflict resolution algorithms using constant size window access," *IEEE Trans. Commun.*, vol. 35, no. 11, pp. 1124–1138, 1987.

[7] N. D. Vvedenskaya and M. S. Pinsker, "Non-optimality of the part-and-try algorithm," in *Abstracts of the International Workshop on Convolutional Codes and Multiuser Communications*, Sochi, USSR, 1983, pp. 141–148.

[8] B. S. Tsybakov and N. B. Likhanov, "Upper bound on the capacity of a random multiple-access system," *Problemy Peredachi Informatsii*, vol. 23, no. 3, pp. 64–78, 1987.

[9] D. Kazakos, L. Merakos, and H. Deliç, "Random multiple access algorithms using a control mini-slot." *IEEE Trans. Comput.*, vol. 46, no. 4, pp. 473–476, 1997.

[10] S. Khanna, S. Sarkar, and I. Shin, "An energy measurement based collision resolution protocol," in *Proceedings of the 18-th ITC conference*, Berlin Germany, 2003.

[11] N. Pippenger, "Bounds on the performance of protocols for a multiple-access broadcast channel," *IEEE Trans. Inf. Theory*, vol. 27, no. 2, pp. 145–151, 1981.

[12] M. Ruszinko and P. Vanroose, "How an Erdös-Rényi-type search approach gives an explicit code construction of rate 1 for random access with multiplicity feedback," *IEEE Trans. Inf. Theory*, vol. 43, no. 1, pp. 368–372, 1997.

[13] G. T. Peeters and B. Van Houdt, "Interference cancellation tree algorithms with k-signal memory locations," *IEEE Transactions on Communications*, vol. 58, no. 11, pp. 3056–3061, Nov. 2010.

[14] R. De Gaudenzi and O. del Rio Herrero, "Advances in random access protocols for satellite networks," in *International Workshop on Satellite and Space Communications (IWSSC)*. IEEE, 2009, pp. 331–336.

[15] G. T. Peeters, R. Bocklandt, and B. Van Houdt, "Multiple access algorithms without feedback using combinatorial designs," *IEEE Transactions on Communications*, vol. 57, no. 9, Sep. 2009.

[16] G. T. Peeters and B. Van Houdt, "Design and analysis of multi-carrier multiple access systems without feedback," in *22nd International Teletraffic Congress (ITC)*. IEEE, 2010, pp. 1–8.

[17] C. Stefanovic and P. Popovski, "Aloha random access that operates as a rateless code," *IEEE Transactions on Communications*, vol. 61, no. 11, pp. 4365–4662, 2013.

[18] E. Paolini, C. Stefanovic, G. Liva, and P. Popovski, "Coded random access: Applying codes on graphs to design random access protocols," *IEEE Communications Magazine, to appear*, 2015.

[19] Y. Yu and G. B. Giannakis, "High-throughput random access using successive interference cancellation in a tree algorithm," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4628–4639, Dec. 2007.

[20] M. Tsatsanis, R. Zhang, and S. Banerjee, "Network-assisted diversity for random access wireless networks," *IEEE Trans. Signal Process.*, vol. 48, no. 3, pp. 702–711, Mar 2000.

[21] R. Zhang, N. Sidiropoulos, and M. Tsatsanis, "Collision resolution in packet radio networks using rotational invariance techniques," *IEEE Trans. Commun.*, vol. 50, no. 1, pp. 146–155, Jan 2002.

[22] B. Ozgul and H. Delic, "Wireless access with blind collision-multiplicity detection and retransmission diversity for quasi-static channels," *IEEE Trans. Commun.*, vol. 54, no. 5, pp. 858–867, May 2006.

[23] I. Cidon and M. Sidi, "Conflict multiplicity estimation and batch resolution algorithms," *IEEE Trans. Inf. Theory*, vol. IT-34, no. 1, pp. 101–110, Jan 1988.

[24] X. Wang, Y. Yu, and G. B. Giannakis, "A robust high-throughput tree algorithm using successive interference cancellation," *IEEE Transactions on Communications*, vol. 55, no. 12, pp. 2253–2256, 2007.

[25] P. J. Ferreira, B. Jesus, J. Vieira, and A. J. Pinho, "The rank of random binary matrices and distributed storage applications," *IEEE Communications Letters*, vol. 17, no. 1, pp. 151–154, 2013.

[26] D. E. Knuth, *Sorting and Searching*, 2nd ed., ser. The Art of Computer Programming. Reading, Massachusetts: Addison-Wesley, 1998, vol. 3, pp. 155–156.