# An Advanced Tree Algorithm with Interference Cancellation in Uplink and Downlink

Quirin Vogel*, Yash Deshpande†, Čedomir Stefanović‡, Wolfgang Kellerer†

*Department of Mathematics, School of Computation, Information and Technology, Technical University of Munich, Germany
†Chair of Communication Networks, School of Computation, Information and Technology, Technical University of Munich, Germany
‡Department of Electronic Systems, Aalborg University, Denmark
Email:{quirin.vogel, yash.deshpande,wolfgang.kellerer}@tum.de, cs@es.aau.dk

***Abstract*—In this paper, we propose Advanced Tree-algorithm with Interference Cancellation (ATIC), a variant of binary tree-algorithm with successive interference cancellation (SICTA) introduced by Yu and Giannakis. ATIC assumes that Interference Cancellation (IC) can be performed both by the access point (AP), as in SICTA, but also by the users. Specifically, after every collision slot, the AP broadcasts the observed collision as feedback. Users who participated in the collision then attempt to perform IC by subtracting their transmissions from the collision signal. This way, the users can resolve collisions of degree 2 and, using a simple distributed arbitration algorithm based on user IDs, ensure that the next slot will contain just a single transmission. We show that ATIC reaches the asymptotic throughput of 0.924 as the number of initially collided users tends to infinity and reduces the number of collisions and packet delay. We also compare ATIC with other tree algorithms and indicate the extra feedback resources it requires.**

***Index Terms*—medium access algorithms, wireless communications, random access, 5G.**

## I. Introduction

Random access (RA) algorithms are crucial for managing a large number of devices that make up the Internet of Things (IoT) ecosystem. IoT scenarios are often characterized by sparse, intermittent packet arrivals and short packet duration [1]. In contrast to scheduling approaches, IoT devices via RA algorithms can transmit their data packets without having to wait for a predetermined time slot, eliminating the signalling overhead required to register devices and schedule packets. In particular, by efficiently managing the transmission of short packets, RA algorithms can help reduce the overall latency of the IoT network, improving the user experience for applications such as real-time monitoring and control [2]. The main challenge for RA algorithms is to avoid collisions that occur when multiple devices try to transmit their data simultaneously, and the major families of RA algorithms differ in the way how the collisions are handled.

The idea behind tree algorithms [3], [4] is to handle collisions by successively partitioning the colliding users into smaller groups. Execution of tree algorithms requires broadcast of the feedback from the access point (AP) after every uplink slot. They exhibit stability until a certain value of the aggregated user arrival rate per uplink slot, which is denoted as the maximum stable throughput (MST). For the basic variant of the algorithm, which is binary tree-algorithm (BTA), the MST is 0.346 packets per slot [3]. Through optimizations of the splitting factor and the channel access policy, the MST of tree algorithms can be increased to 0.4878 packets/slot [5].

Advanced signal processing techniques, such as interference cancellation, further improve the MST of tree-algorithms [6]. Specifically, in binary tree-algorithm with successive interference cancellation (SICTA), after decoding a user packet, the AP successively applies interference cancellation over the previously received and stored collision slots to remove the interference contribution of the decoded user, which potentially enables the decoding of new packets and new rounds of successive interference cancellation (SIC). The MST of SICTA is 0.693 packets/slot.

In this paper, we propose an algorithm called Advanced Tree-algorithm with Interference Cancellation (ATIC), that, leveraging enhanced feedback and Interference Cancellation (IC) capabilities at the users, resolves collisions of degree 2 using just 1 extra uplink slot. This way, the algorithm can achieve MST of 0.924, as shown in the paper, exceeding the MST of SICTA by over 33%. We also show that gated access is the best access method for the proposed algorithm. To compare the extra feedback cost of the proposed algorithm, we assess the amount of resources required in the feedback for the relevant classes of tree algorithms. Finally, we show that the mean packet delay[1] of the proposed algorithm is significantly less than the one of SICTA for significantly higher arrival rates.

The rest of the paper is organized as follows. Section II provides the background and an overview of the related work. Section III states the system model explains the proposed algorithm ATIC. Section IV derives the MST of ATIC. We show some simulation-based evaluation of ATIC in terms of the required resources and mean packet delay in Section V. Finally, we conclude the paper by discussing our findings in Section VI.

[1]The mean number of uplink slots needed to successfully decode the packet after its arrival at a user.

## II. Background and Motivation

### A. Standard Tree Algorithms

The BTA, also known as the Capetanakis-Tsybakov-Mikhailov type collision-resolution protocol (CRP) was proposed in [3]. In BTA, the users decide on which slot to transmit based on the feedback from the AP. The feedback for any slot $t$ is ternary $F[t] \in \{0, 1, e\}$, $\forall t \in \mathbb{Z}^+$, denoting whether the outcome of the slot was an idle $F[t] = 0$, a success $F[t] = 1$, or a collision $F[t] = e$, respectively.

The CRP starts by $n$ users transmitting their packets for the first time in the coming slot, thus marking the beginning of a collision resolution interval (CRI) (for which the slot counter $t$ is set to 0). The moments of the initial transmissions are determined by the channel-access protocol (CAP), which will be elaborated later. If $n \geq 2$, the slot is a collision, i.e., $F[t] = e$, and the $n$ users independently split into two groups, e.g. group 0 and group 1, according to the outcome of a Bernoulli trial. The users who have joined group 0 transmit in the next slot, $t + 1$, while users who have joined group 1 abstain from transmitting and observe the feedback until users from group 0 are resolved. If this slot is also a collision, $F[t+1] = e$, then the process of partitioning the users further is done recursively. Users who have joined group 1 wait until all the users in group 0 have successfully transmitted their packets to the AP, after which they transmit in the next slot and the resolution process continues. The CRI ends when all $n$ users become resolved. The BTA algorithm can be implemented in each user by means of a simple counter [3].

One can represent the progression of a CRI in terms of full-binary trees, as shown in Fig. 1. Here, we show an example with $n = 4$, where the users are labelled with A, B, C, D. Each node on the tree represents a slot. The labels inside the node show the users that have transmitted in that slot. A node with an empty label inside is idle, while a node with only one label inside is a success. The numbers outside the node represent the slot number in BTA. Hence, the CRI in this example took 9 slots. Leaf nodes are either successes or idle slots while all internal nodes are collisions. The tree structure allows for a recursive analysis of the properties of the CRP.

A full-fledged tree algorithm combines the CRP with a CAP that determines how and when the arriving users will transmit their packets for the first time. The common CAPs are gated, windowed, and free access. In gated access (also called blocked access), when a CRI is in progress, all newly arrived users wait, i.e., they are blocked. The blocked users transmit on the channel in the next slot after the current CRI ends. In windowed access, the time is divided into windows whose length is $\Delta$ slots, $\Delta \in \mathbb{R}^+$. The users arriving in the $k$-th time window join the next CRI after the CRI of the users that arrived in the $(k-1)$-th window is over. In free access, a newly arrived user simply transmits on the channel in the first slot after its arrival. It thus joins an ongoing CRI, if there is one. For all the three CAPs, the main performance parameter is the MST. If the total packet arrival rate in the network $\lambda > 0$ is less than the MST, then the RA scheme will be stable.

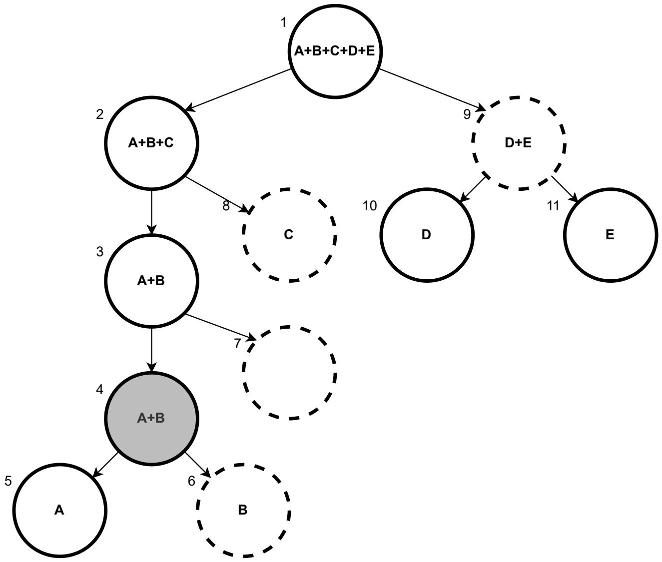


Fig. 1: An example visualization of different tree algorithms. At each collision of more than 2 signals, the users independently split into two groups the users that selected the left group transmit once again. The skipped slots in SICTA are drawn with dotted lines. In ATIC, collisions of degree two (skipped or not) are resolved deterministically. ATIC is able to skip slot 4, as $A$ and $B$ can infer that they are in a collision of degree two. Slot 4 and 5 hence become deterministic and so do 10 and 11.

It was shown in [3] that the MST of BTA with gated access is 0.346 packets per slot. This MST is improved by free access to 0.392 packets/slot [7] and by windowed access to 0.429 packets/slot [4]. An improvement to the BTA called modified tree-algorithm (MTA) was suggested in [4] where a definite collision can be skipped. If there is an idle slot after a collision, the next slot is a definite collision, and the users can avoid transmitting on this slot by randomly splitting once again. For example, slot 4 in Fig. 1 is a definite collision and will be skipped in MTA. Ternary MTA with biased splitting was shown to be the optimum choice [7] among MTA. In biased splitting, the probability of a user joining any of the $d$ groups is not uniform. MTA with clipped access (a version of windowed access) was introduced in [5] and achieves MST of 0.4878 packets/slot.

### B. SICTA

The collisions in BTA are discarded by the AP. One way to improve the MST of tree algorithms was suggested in [6], [8], called SICTA. Here the signals of all collisions are saved by the AP. The AP can then resolve more than one packet per slot by successively subtracting the signals of previously decoded packets (i.e., cancelling their interference) from the previously stored collisions.

To illustrate how certain slots from the BTA will be skipped in SICTA, we use the example in Fig. 1 and use the same slot number (labeled outside the node) as the ones in the figure. Let $Y_t$ be the signal (received by the AP) of slot $t$ and $X_i$ be

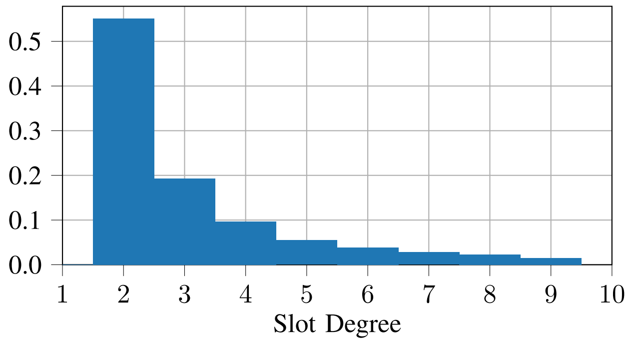


Fig. 2: Simulated normalized collision degree distribution for SICTA with blocked access for 100000 collision slots. The packet arrival rate in the network is 0.693 which is the MST for this scheme. We see that more than half the collisions have a multiplicity of 2.

the signal from user $i$. The algorithm will proceed as MTA until slot 6. The AP will save the signal from the collisions in its memory. Thus the 3 signals in its memory are $Y_1 = X_A+X_B+X_C+X_D$, $Y_2 = X_A+X_B+X_C$, $Y_5 = X_A+X_B$. It can decode the signal from user A since it was the only user to transmit in slot 6, $Y_6 = X_A$. As soon as there is a success in slot 6, the AP will subtract the known signal of user $A$ from all the 3 previous saved collisions. After this step, the AP can also decode $X_B$ since $Y_5 - X_A = X_B$. The AP then proceeds to subtract $X_B$ from its known collisions and is able to decode the signal of user C from the collision in the second slot. Finally, the AP is also able to decode the signal from user D after subtracting $X_C$ from the first collision. Since all the users from the initial collision are decoded, the CRI is over after slot 6. Thus using SIC, one can skip 4 slots out of the 9 from BTA tree. The skipped slots in SICTA are marked with dotted circles.

The MST of SICTA with blocked access was shown to be 0.693 [6]. It was also shown that neither free access nor windowed access help in increasing the MST of SICTA above the one of blocked access [9]. Splitting beyond binary, i.e., $d-$ary SICTA reduces the MST for fair splitting. However, if the probability of joining each user follows a special distribution, MST of 0.693 can be achieved for any splitting value of $d$ [10], [11].

The massive improvement of SICTA over BTA in terms of the MST comes at the cost of higher complexity at the AP. The AP must be able to save collision signals in its memory and be able to perform SIC at the end of every slot. The broadcast feedback by the AP is also more complex, $F[t] \in \{0, k, e\}$ where $k$ is an integer indicating the number of slots that should be skipped by all active users in the CRI after every success.

We now turn to the motivation for the scheme proposed in this paper. Fig. 2 shows the distribution of the number of packets in a collision slot, denoted as the collision degree distribution, for the SICTA algorithm with blocked access for the MST simulated over 100000 slots. Collisions of degree two are the most common in SICTA. On the other hand, the throughput[2] of SICTA is the lowest [8, Table II] when resolving collision of degree 2. In contrast, the scheme proposed in this paper has a throughput of 1 packet/slot when resolving collisions of degree 2. This is achieved by sending more information to users in the feedback and assuming that the users are also able to perform IC, as further elaborated in Section III.

## III. System Model

We consider a large user population in the wireless range of an AP. Time in the system is divided into slots of equal duration. Throughout the paper, we express all time-related quantities in terms of slots denoted by an index $t \in \mathbb{Z}^+$.

Data packets arrive randomly at users, and we assume that each arrival happens at a different user and that the total number of arrivals in a slot is Poisson distributed random variable. The users experiencing packet arrivals transmit the packets to the AP over a shared wireless channel. A user is said to be active when it has a packet to send to the AP. Throughout this paper, we also refer to packets as users. All packets are of equal length. Each user has a unique identity, and this value is appended to the data packet. The transmission rate of the users is such that it takes exactly one slot to transmit one packet.

The wireless channel is *interference-limited*, modeled as the standard collision channel in the uplink as well as downlink. Therefore, if only one user transmits in a slot, the data from the packet can be successfully decoded. If more than one user transmits in a slot, their signals interfere, and the AP cannot decode any of them; this scenario is called a collision. If no user transmits in the slot, then we say that the slot is idle. The AP broadcasts immediate and instantaneous feedback to all the users in the network. We discuss the contents of this feedback in the next subsection. A user is said to be active when it has a packet in its queue to send to the AP.

Users are equipped with an internal memory that stores two signals: their own signal and a received signal from the AP.

For every slot $t$, the AP broadcasts a signal indicating the feedback $F[t] \in \{0, k, e\}$ *and* a signal $Z_t$ to all the users in the network. If $F[t] = e$, then $Z_t = Y_t$, i.e., the signal that was just received by the AP. If $F[t] = k$, the AP broadcasts $Z_t = Y_s - Y$, where $s < t$ and $Y_s$ is the most recent signal that is unresolved after performing SIC and $Y$ is the sum of all signals resolved in between slot $s$ and slot $t$. Finally, if $F[t] = 0$, then $Z_t = \emptyset$.

The users keep the last received valid $Z_{t-p} \neq \emptyset, p \in \mathbb{N}^+$ and their own signal $X_i$ in their memory. Upon a collision or success, the active users proceed to cancel their own signal $X_i$ from the broadcasted signal. If $F[t] = e$ and the active user $X_i$ can resolve the signal $Y_t - X_i$, $Y_t$ must have been a collision of degree two. If $F[t] = k$ and the active user $i$ can resolve the signal $Z_t - X_i$, then $Y_s - Y - X_i$ is a decodable

[2]Throughput is the measure of the efficiency and is calculated as the ratio of the number of decoded packets and the number of slots required for their successful decoding.

signal and user $i$ is in a skipped slot of degree two with the signal $Y_s - Y$.

Thus, the algorithm allows the users to identify that only two users are left in the current subtree as well as the other user's signal. Once this information is available to both contending users, it only remains to ensure that they avoid a collision in slot $t+1$. As mentioned previously, each user has a unique ID that is appended to the data packet. A pre-defined hierarchical system (for example, rank users according to the value of their ID) can then be used to decide in a distributed manner which of the two contending users should transmit in slot $t+1$. Both the users will be decoded by the AP in slot $t+1$ as the AP can subtract $Y_{t+1}$ from $Y_t$.

We illustrate ATIC using the example from Fig. 1. In slot 3, the AP broadcasts $X_A + X_B$ and $F[3] = e$. This allows $A$ and $B$ to cancel their own signal from $X_A + X_B$. In the example, $A$ finds its user ID to be higher than that of $B$ and transmits in slot 5. Note that slot 4 is skipped in ATIC, which is not the case for SICTA. The same process is done by user $B$ which finds its user ID less than that of $A$'s and decides to abstain from transmitting in slot 5, fully knowing that $A$ will transmit and its own signal will be resolved along with that of user B's after slot 5. After slot 5, the AP broadcasts the signal $X_D + X_E$, which is the difference between $Y_1 = X_A+X_B+X_C+X_D+X_E$ and the sum of the signal decoded between slot 1 and 5, which is $X_A + X_B + X_C$. Users $D$ and $E$ can then perform SIC, with the result that $D$ finds its user ID to be higher and broadcasts first.

Note that the throughput of the algorithm described above is equivalent to the following: the AP *always* broadcasts the signal it has received, i.e. collisions and singletons. It hence does not require the AP to calculate *and* broadcast unresolved signals, for example to infer that $(X_A + X_B + X_C) - X_A$ is equal to $X_B + X_C$. Users store all broadcasted signals in their internal memory and perform SIC like the AP with the additional knowledge of their own signals in the SIC process. The version of ATIC introduced previously has the same throughput as this variant. However, it requires users to reserve a large chunk of memory for storing the received signal and the users need have essentially the same computational capacity as the AP to perform SIC. Below we discuss a variant that does not come at the cost of increased storage capacities but uses the same feedback.

*Simpler variant with a lower throughput*

We can still achieve a significant increase in throughput over SICTA if we apply ATIC to only non-skipped collisions i.e., on the left subtree. In this variant, the AP only broadcasts the received signal *if* it is a collision, i.e., $F[t] = e$. In Fig. 1, this still causes slot 4 to be skipped, as here we have a collision of degree two which is in the left group. However, slots 10 and 11 are no longer deterministic in this version of ATIC, since the signal $X_D + X_E$ is actually never broadcasted. Therefore, we only improve throughput in collisions of degree 2, that occur on the left nodes of branches. Analogous to the calculations performed in Section IV, we can show that the throughput of this variant with fair splitting is $6\log(2)/5 \approx 0.832$, which is a 20% gain over SICTA.

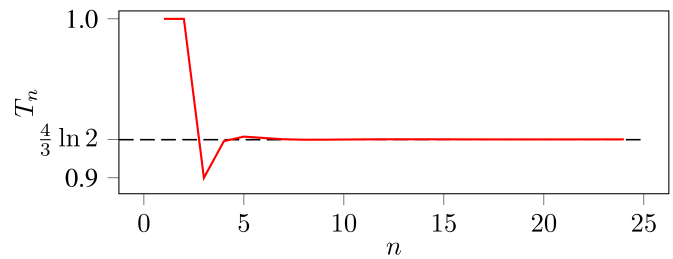


Fig. 3: The throughput conditioned on a collision of $n$ users. The conditional throughput is 1 when $0 < n \leq 2$ and tends to $\frac{4}{3}\ln 2$ as $n \longrightarrow \infty$. The lowest conditional throughput of 0.9 is achieved for $n = 3$.

## IV. Analysis

Let $l_n$ be the conditional CRI length, i.e., the CRI length given $n$ users have collided in the first slot. The evolution $l_n$ can be expressed recursively as,

$$l_n = \begin{cases} 1 & \text{if } n = 0, 1 \\ 2 & \text{if } n = 2 \\ l_i + l_{n-i} & \text{if } n \geq 3 \end{cases} \tag{1}$$

where $i$ is the (random) number of users out of $n$ which have chosen group 0. We assume that each packet chooses group 0 independently of one another and with probability $p \in (0, 1)$. We are interested in finding the expected CRI length conditioned on the fact that $n$ users have collided in the first slot, $L_n = \mathrm{E}[l_n]$. The expected CRI length allows us to express the conditional throughput $T_n = \frac{n}{L_n}$, which measures the average efficiency of resource utilization of the CRP.

### A. Closed-form equation of $L_n$

Using (1) one can derive a recursive equation for $L_n$. However, we skip this derivation since a closed-form expression exists for $L_n$.

**Theorem 1.** *Let $q = 1-p$ and set $r = 2 - 4pq - 3\left(p^2 + q^2\right)$. We then have that for every $n \geq 0$*

$$L_n = 1 + \sum_{i=2}^{n} \binom{n}{i} \frac{(-1)^i\,(i - 1 + ri(i-1)/2)}{1 - p^i - q^i}. \tag{2}$$

*Proof.* The proof is given in Appendix A. □

Fig. 3 shows the conditional throughput $T_n$ for different values of $n$ according to (2) in the case $p = q = 1/2$. In the worst case, $T_n = 0.9$ for when $n = 3$. In comparison, in the best case, $T_n$ of SICTA is 0.693 or $\ln 2$. Moreover, $T_n$ of SICTA for $n = 2$ is 0.66, while $T_n = 1$ for $n = 2$ for the proposed algorithm.

### B. Asymptotic Behaviour

The closed-form expression (2) is useful for calculating the behavior of $L_n$ and $T_n$ when $n$ is small. However, as $n$ grows, the number of summands in (2) increases and the computation becomes numerically challenging due to cancellation effects.

On the other hand, Fig. 3 suggests that the value of $T_n$ seems to settle at $\frac{4}{3}\ln 2$ as $n$ grows. Here we formally prove that, asymptotically, $T_n$ tends to $\frac{4}{3}\ln 2$, save an oscillatory component of rather small amplitude.

**Theorem 2.** *The throughput $n/L_n$ of ATIC is asymptotically maximized for $p = q = 1/2$. For this value, we have*

$$T_n = \frac{n}{L_n} = \frac{4}{3}\ln(2) + g(n) + o(1)\,, \tag{3}$$

*where $g(n)$ is a small sine-like perturbation, as in [7], usually between $10^{-3}$ and $10^{-6}$.*

The proof is given in Appendix B, also providing an asymptotic expansion for all values of $p$, see (29).

The asymptotics of $T_n$ can be used to derive the MST for the case of the blocked access using the techniques from [7] or [8], where MST $= \lim_{n\to\infty} T_n$. Comparing the MST of the proposed tree algorithm with the one of SICTA $(\log(2))$, there is a gain of one-third.

### C. Windowed Access

In windowed access, the packets that start the $k$-th CRI have arrived during the windowed interval $(k\Delta, (k+1)\Delta)$, where $\Delta$ is the window size that is optimized for the arrival rate. Using windowed access makes sense only if it supports a higher arrival rate than gated access.

From Fig. 3 we see that the conditional throughput is 1 for $0 < n \le 2$. This high efficiency for a small number of users hints to the possibility that if we can restrict most CRIs to start with $n < 3$ by using an optimized window size, our algorithm might perform better. Indeed, this is the case for BTA [12] and tree algorithms with multi-packet reception (MPR) [13], [14], where the of the CRP is higher for smaller $n$, such that windowed access pushes the MST to be higher than that of gated access. For SICTA, the efficiency is in fact lower for smaller $n$ and hence windowed access does not improve the MST over blocked access [8].

We use the method in [12] to numerically find the optimal window size and the corresponding MST for our algorithm. The probability that $n$ packets arrive in window $\Delta$ when the arrival rate is Poisson distributed with mean $\lambda$ packets per slot is then

$$\Pr\{N = n\} = \frac{(\lambda\Delta)^n}{n!} e^{-\lambda\Delta}. \tag{4}$$

The expected CRI length conditioned on the window size and packet arrival rate is,

$$L(\lambda\Delta) = \mathrm{E}\{L_n|\lambda\Delta\} = \sum_{n=0}^{\infty} L_n \frac{(\lambda\Delta)^n}{n!} e^{-\lambda\Delta}. \tag{5}$$

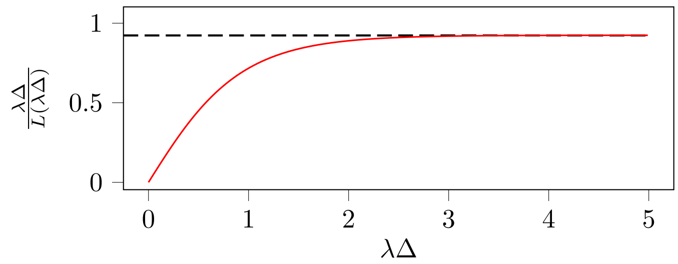


Fig. 4: Throughput for windowed access with ATIC as function of the expected number of arrivals per window $\lambda\Delta$. We see that gated access is the best CAP to be used along with ATIC.

For the RA scheme to be stable, the mean CRI length has to be less than the window size $\Delta$.[3] Thus,

$$L(\lambda\Delta) < \Delta. \tag{6}$$

Rewriting the above equation, we get the mean arrival rate for which the RA scheme will be stable, if $\lambda < \frac{\lambda\Delta}{L(\lambda\Delta)}$. Fig. 4 plots the function $\frac{\lambda\Delta}{L(\lambda\Delta)}$ for different values of $\lambda\Delta$. The function increases to $\frac{4}{3}\ln 2$ as $\lambda\Delta$ grows, implying that the optimal window size (and thus the product $\lambda\Delta$) tend to infinity. Thus, using windowed access does not provide benefits in terms of the MST in comparison to gated access.

### D. Delay Analysis

Similar to [8], [15], one can use the moment generating function from Equation (10) to approximate the average delay experienced by each packet. As the delay does not exhibit a closed-form solution (see [8, Section V.C]), we numerically simulate the delay using the method provided in [15]. The mean packet delay for BTA, SICTA and ATIC for different mean packet arrival rates $\lambda$ is shown in Fig. 5. We see that ATIC significantly reduces the delay for arrival rates higher than 0.5 packets per slot. For example, at an arrival rate of $\lambda = 0.5$, SICTA provides a mean packet delay of 1.7 slots while ATIC gives a mean packet delay of 1 slot only. This difference gets more and more pronounced as the packet arrival rate approaches the MST of SICTA. At $\lambda = 0.693$ packets/slot, the delay of the SICTA scheme becomes unbounded. For $\lambda$ at 95 % of their respective MST, SICTA gives a mean packet delay of 12.3 slots while ATIC gives a mean packet delay of 10.2 slots.

## V. Feedback and Memory Requirements

### A. Feedback Requirements

The throughput gain in the proposed algorithm comes at the cost of requiring more resources for the downlink broadcast feedback. Here we assess the number of resources required for the feedback for BTA, SICTA, and ATIC. We note that this type of analysis is by default neglected in the available literature. We also note that the required feedback resources

[3] $\mathrm{E}\{L_n^2\} < \infty$ also holds, but we omit the proof which can be done following [12].

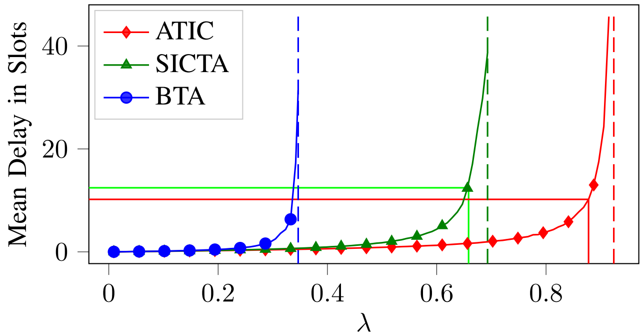


Fig. 5: The mean packet delay of BTA, SICTA, and ATIC. The packets arrive in the network independently with a Poisson distribution with mean $\lambda$. Even for arrival rates theoretically supported by SICTA in the interval (0.6, 0.693] packets per slot, the delay becomes very large. In such cases, ATIC provides a much lower packet delay.

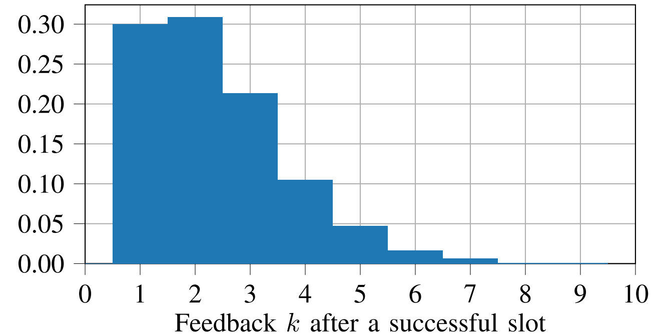


Fig. 6: Simulated probability mass distribution of the feedback value $k$ conditioned that the slot was a success, obtained for SICTA with blocked access for 100000 slots. The packet arrival rate $\lambda$ is 0.693, which is the MST of the scheme. It can be observed that the value of $k$ did not exceed 9.

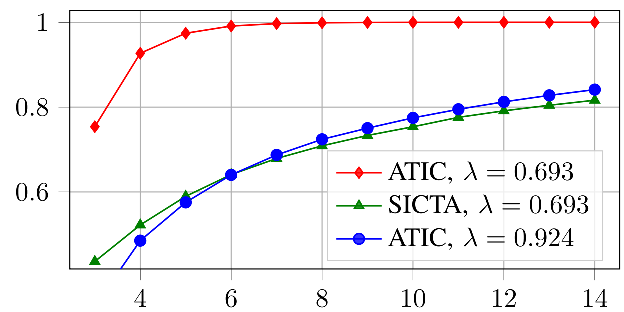


Fig. 7: The empirical cumulative distribution function of collisions per CRI for SICTA and ATIC obtained by simulation with gated access over 100000 slots. Since the AP has to keep every collision in a CRI in its memory, this value is indicative of the number of slots the AP needs to hold in its memory. When the arrival rate $\lambda = 0.693$, ATIC needs at the most 7 slots which is much less than the 50 needed by SICTA. When the arrival rate $\lambda = 0.924$, the required memory for ATIC is comparable to SICTA with $\lambda = 0.693$.

are by default not included in the throughput calculation, which is the case in this paper as well.

For BTA, the ternary feedback would need 2 bits/slot. Theoretically, for SICTA, the feedback message upon a success that contains information about how many slots (i.e., $k$) should be skipped can be any positive integer. To estimate the number of bits one would need in practice for SICTA in the broadcast feedback, we simulated SICTA for the maximum supported arrival rate (at MST). Fig. 6 shows the simulated probability mass distribution for $k$. It can be observed that the value of $k$ did not exceed 9, leading us to conclude that in practice 4 bits for the broadcast feedback in case of SICTA can be used to represent all required feedback messages with a high probability.

In the case of ATIC, the entire received signal must be broadcast in the feedback by the AP in the case of a collision. It is reasonable to assume that the number of bits required for the feedback can be approximated by the packet size in bits $B$, and in practice, it holds that $B \gg 4$. Moreover, in ATIC, the feedback requirements increase with the packet size. Nevertheless, uplink and downlink channels are of comparable capacity in a multitude of wireless cellular technologies (e.g., LTE), so this type of requirement could effectively be supported in practice.

### B. Memory Requirements

In both SICTA and ATIC, all collisions need to be stored in the APs memory. Following the methods given in [11], one can show that for gated access, the $C_n/L_n \sim \frac{1}{2}$, where $C_n$ is the expected number of collisions given $n$ packets in the initial collision. Hence, we have for ATIC

$$\frac{C_n}{n} \sim \frac{3}{8}\frac{1}{\ln(2)}\,, \tag{7}$$

which is significantly less than the $\frac{C_n}{n} \sim (2\log(2))^{-1}$ for SICTA. During a CRI, the AP needs to hold all the collisions in its memory. Every resolved packet is then subtracted from every saved collision to attempt IC. Hence the number, of collisions in a CRI determines the memory requirements at the AP in slots. Thus on average, the required memory capacity of ATIC is smaller than that of SICTA for the same number of users starting the contention.

Figure 7 shows the CDF of the collisions per CRI obtained for ATIC and SICTA by simulating gated access over 100000 slots. When the packet arrival rate, $\lambda$ is 0.693 packets/slot (MST of SICTA), ATIC needed a memory capacity of 7 slots. In comparison, SICTA needed a memory capacity of 50 slots. On the other hand, when the arrival rate is 0.924 packets/slot (MST of ATIC), the required memory capacity is comparable to SICTA with $\lambda$ at 0.693 packets/slot. Thus, ATIC does not require more memory capacity than SICTA when the arrival rates are close to their respective MSTs. When the arrival rates are the same, ATIC needs a much smaller memory capacity.

Note that, ATIC also requires the users to hold 2 signals in their memory, namely their own and the received collision feedback. In the case of SICTA and BTA they need to hold just 1 signal, i.e., their own.

## VI. Conclusion

In this paper, we have shown that by broadcasting the received composite signal and leveraging IC at the user side, we can increase the MST to 0.924. This is an improvement over SICTA by one-third and rather close to the absolute limit of 1 for the collision channel model. Our proposed scheme also lowers the average packet delay and required memory capacity at the AP.

The requirement of enhanced feedback limits the practical applicability of the scheme to systems with frequent and adept feedback channels and/or with short packet sizes. An example of the latter is mobile cellular systems, such as LTE, where the capabilities and scheduling of the downlink and uplink channels are balanced by default.


## Acknowledgements

The work of Č. Stefanović was supported by the European Union's Horizon 2020 research and innovation programme under Grant Agreement number 883315. Q. Vogel would like to thank Silke Rolles for funding part of this research. Y. Deshpande's work was supported by the Bavarian State Ministry for Economic Affairs, Regional Development and Energy (StMWi) project KI.FABRIK under grant no. DIK0249.


## Appendix

### A. Proof of Theorem 1

We prove the statement by deriving a functional equation for the moment generating function and calculate the mean by taking the derivative.

Write for $n \geq 0$ and $Q_n(z) = \mathrm{E}\left[z^{l_n}\right]$, where $z \in \mathbb{C}$. Equation (1) gives

$$Q_0(z) = Q_1(z) = z \quad \text{and} \quad Q_2(z) = z^2\,. \tag{8}$$

Note that $i$, the number of packets choosing the left slot, follows a binomial distribution with parameter $p$. By conditioning on $i$ and using Equation (1), we get for $n \geq 3$

$$Q_n(z) = \mathrm{E}\left[z^{l_n}\right] = \sum_{i=0}^{n} \binom{n}{i} p^i \left(1-p\right)^{n-i} Q_i(z) Q_{n-i}(z)\,. \tag{9}$$

Write now $q = 1-p$, for brevity. Using the two equations above, we get for the Poisson moment generating function $Q(x,z) = \sum_{n\geq 0} x^n Q_n(z)/n!$ that

$$Q(x,z) = (1+x)z + \frac{x^2 z^2}{2} + \sum_{n\geq 3}\sum_{i=0}^{n} \left(\frac{Q_i(z)(px)^i}{i!}\right)\left(\frac{Q_{n-i}(z)(xq)^{n-i}}{(n-i)!}\right)\,. \tag{10}$$

Note that the final term can be rewritten as

$$\sum_{n\geq 0}\sum_{i=0}^{n} \left(\frac{Q_i(z)(px)^i}{i!}\right)\left(\frac{Q_{n-i}(z)(xq)^{n-i}}{(n-i)!}\right) - z^2 - z^2 x - x^2\left(z^3 p^2/2 + z^2 pq + z^3 q^2/2\right)\,, \tag{11}$$

by adding and subtracting the terms for $n = 0, 1, 2$. The previous two equations give

$$Q(x,z) = Q(px,z)Q(qx,z) + \left(z - z^2\right) + x\left(z - z^2\right) + \frac{x^2}{2}\left(z^2 - 2z^2 pq - z^3\left(p^2 + q^2\right)\right)\,. \tag{12}$$

The moment generating function of $L_n$ is given by $L(x) = \mathrm{e}^{-x}\sum_{n\geq 0} x^n L_n/n!$. By differentiation, one obtains that $L(x) = \mathrm{e}^{-x}\frac{\mathrm{d}Q}{\mathrm{d}z}(x,1)$. Hence, Equation (12) gives

$$L(x) = L(px) + L(qx) - \mathrm{e}^{-x}\left(1 + x - \frac{x^2}{2}\left(2 - 4pq - 3\left(p^2+q^2\right)\right)\right)\,. \tag{13}$$

Set $r = \left(2 - 4pq - 3\left(p^2+q^2\right)\right)$. We then have that for $L(x) = \sum_{n\geq 0}\alpha_n x^n$ and $n \geq 3$

$$\alpha_n = \frac{1}{n!}\frac{\left(-1\right)^n\left(n - 1 + rn(n-1)/2\right)}{1 - p^n - q^n}\,, \tag{14}$$

by coefficient comparison. Note that by the definition of $l_n$,$\alpha_0 = 1$, $\alpha_1 = 0$ and $\alpha_2 = 1/2$. Using that $L_n = \sum_{i=0}^{n}\alpha_i n!/(n-i)!$ immediately gives

$$L_n = 1 + \frac{n(n-1)}{2} + \sum_{i=3}^{n}\binom{n}{i}\frac{\left(-1\right)^i\left(i - 1 + ri(i-1)/2\right)}{1 - p^i - q^i}\,. \tag{15}$$

We rewrite the right-hand side as

$$1 + \frac{n(n-1)}{2}\left(1 - \frac{1+r}{1-p^2-q^2}\right) + \sum_{i=2}^{n}\binom{n}{i}\frac{\left(-1\right)^i\left(i - 1 + ri(i-1)/2\right)}{1 - p^i - q^i}\,. \tag{16}$$

Note that $1 - \frac{1+r}{1-p^2-q^2} = 2\frac{-1+(p+q)^2}{1-p^2-q^2} = 0$. Hence

$$L_n = 1 + \sum_{i=2}^{n}\binom{n}{i}\frac{\left(-1\right)^i\left(i - 1 + ri(i-1)/2\right)}{1 - p^i - q^i}\,. \tag{17}$$

This concludes the proof of Theorem 1. ☐

### B. Proof of Theorem 2

The starting point of our asymptotic analysis of $L_n$ is Equation (17). Note that by [8, Equation 34]

$$\sum_{i=2}^{n}\binom{n}{i}\frac{\left(-1\right)^i\left(i-1\right)}{1 - p^i - q^i} \sim n\frac{1}{-p\log(p) - q\log(q)} + n g_1(n)\,, \tag{18}$$

where $g_1(n)$ is a small fluctuation as described in Theorem 2.

We now analyse the remaining part of the sum:

$$\frac{r}{2}\sum_{i=2}^{n}\binom{n}{i}\frac{\left(-1\right)^i i(i-1)}{1 - p^i - q^i}\,. \tag{19}$$

As $r/2$ is a linear factor, we neglect it for now and multiply it back on later.

By differentiation for the binomial theorem, one obtains

$$\sum_{i=2}^{n} \binom{n}{i} i(i-1)x^i = x^2 n(n-1)\left[1+x\right]^{n-2} . \tag{20}$$

We write $\mathfrak{P}_m^{(2)} = \{\mu_1, \mu_2 \in \mathbb{N} \cup \{0\} : \mu_1 + \mu_2 = m\}$. We also abbreviate $p(\mu) = p^{\mu_1} q^{\mu_2}$. Using the geometric series, we get that

$$\begin{aligned} &\sum_{i=2}^{n} \binom{n}{i} \frac{(-1)^i \left(i(i-1)\right)}{1-p^i-q^i} \\ &= \sum_{m\geq 0} \sum_{\mu \in \mathfrak{P}_m^{(2)}} \binom{m}{\mu} p(\mu)^2 n(n-1)\left(1-p(\mu)\right)^{n-2} . \end{aligned} \tag{21}$$

Using a similar reasoning to [7], we extract the leading term

$$\sum_{m\geq 0} \sum_{\mu \in \mathfrak{P}_m^{(2)}} \binom{m}{\mu} p(\mu)^2 n^2 \left(1-p(\mu)\right)^{n} . \tag{22}$$

Using the same reference again, we write this as

$$(1+o(1)) \sum_{m\geq 0} \sum_{\mu \in \mathfrak{P}_m^{(2)}} \binom{m}{\mu} p(\mu)^2 n^2 \mathrm{e}^{-np(\mu)} . \tag{23}$$

Recall that for a function $f$, its Mellin transform (see [16]) is given by

$$\mathcal{M}\left[f(x); s\right] = \int_0^\infty x^{s-1} f(x) \mathrm{d}x \tag{24}$$

with inverse transform

$$f(x) = \frac{1}{2\pi i} \int_{c-i\infty}^{c+i\infty} x^{-s} \mathcal{M}\left[f(x); s\right] \mathrm{d}s \, , \tag{25}$$

for some $c \in \mathbb{R}$. For $f(x) = x^2 \mathrm{e}^{-x}$, we have that it is

$$\mathcal{M}\left[f(x); s\right] = \Gamma(s+1) \quad \text{if} \quad \Re(s) > -2 \, . \tag{26}$$

Hence, we get that the sum in Equation (23) can be rewritten as

$$\sum_{m\geq 0} \sum_{\mu \in \mathfrak{P}_m^{(2)}} \binom{m}{\mu} \frac{1}{2\pi i} \int_{-3/2-i\infty}^{3/2+i\infty} n^{-s} p(\mu)^{-s} \Gamma(s+1) \mathrm{d}s \, . \tag{27}$$

We use the geometric sum again to rewrite the above as

$$\frac{1}{2\pi i} \int_{-3/2-i\infty}^{3/2+i\infty} \frac{n^{-s}\Gamma(s+1)}{1-p^{-s}-q^{-s}} \mathrm{d}s \, . \tag{28}$$

Using the reside theorem again as in [7], we get the the above integral is given by

$$n\left(\frac{1}{-p\log(p)-q\log(q)} + g_2(n)\right) , \tag{29}$$

where $g_2(n)$ is again small and fluctuating. Combining Equation (18) and Equation (29), we get that

$$\frac{L_n}{n} = \frac{1+r/2}{-p\log(p)-q\log(q)} + g_3(n) \, . \tag{30}$$

In the case of fair splitting, we get that $r = -1/2$ and hence

$$\frac{n}{L_n} \quad \text{has leading term} \quad \frac{4}{3}\log(2) \approx 0.9242 \, . \tag{31}$$

Using [17], we note that $g_3(n)$ is zero if and only if $\log(p/q)$ is irrational. Note that by expanding $r$, the leading term in Equation (30) can be written as

$$\frac{p^2-p-0.5}{(1-p)\log(1-p)+p\log(p)} \, , \tag{32}$$

and is hence minimized at $p = 1/2$. This concludes the proof of Theorem 2. □

## References


[1] J. Navarro-Ortiz, P. Romero-Diaz, S. Sendra, P. Ameigeiras, J. J. Ramos-Munoz, and J. M. Lopez-Soler, "A survey on 5G Usage Scenarios and Traffic Models," *IEEE Communications Surveys Tutorials*, 2020.

[2] M. Vilgelm, H. Murat Gürsu, and W. Kellerer, "Random Access Protocols for Industrial Internet of Things: Enablers, Challenges, and Research Directions," in *Wirel. Networks Ind. IoT Appl. Challenges Enablers*. Springer International Publishing, 2021.

[3] J. Capetanakis, "Tree algorithms for packet broadcast channels," *IEEE Trans. Info. Theory*, vol. 25, no. 5, pp. 505–515, Sep. 1979.

[4] J. L. Massey, "Collision-resolution algorithms and random-access communications," in *Multi-user communication systems*. Springer, 1981, pp. 73–137.

[5] S. Verdu, "Computation of the efficiency of the mosely-humblet contention resolution algorithm: A simple method," *Proceedings of the IEEE*, vol. 74, no. 4, pp. 613–614, 1986.

[6] Y. Yu and G. Giannakis, "SICTA: a 0.693 contention tree algorithm using successive interference cancellation," in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, vol. 3, 2005, pp. 1908–1916 vol. 3.

[7] P. Mathys and P. Flajolet, "Q-ary collision resolution algorithms in random-access systems with free or blocked channel access," *IEEE Transactions on Information Theory*, vol. 31, no. 2, pp. 217–243, 1985.

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

[9] G. T. Peeters and B. Van Houdt, "On the maximum stable throughput of tree algorithms with free access," *IEEE Transactions on Information Theory*, vol. 55, no. 11, pp. 5087–5099, 2009.

[10] Y. Deshpande, C. Stefanović, H. Murat Gürsu, and W. Kellerer, "Correction to: High-throughput random access using successive interference cancellation in a tree algorithm," *IEEE Transactions on Information Theory*, pp. 1–1, 2022.

[11] Q. Vogel, Y. Deshpande, C. Stefanović, and W. Kellerer, "Analysis of d-ary Tree Algorithms with Successive Interference Cancellation," *arXiv:2302.08145*, 2023.

[12] R. Rom and M. Sidi, *Multiple access protocols: performance and analysis*. Springer Science & Business Media, 2012.

[13] C. Stefanović, H. M. Gürsu, Y. Deshpande, and W. Kellerer, "Analysis of Tree-Algorithms with Multi-Packet Reception," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020, pp. 1–6.

[14] C. Stefanović, Y. Deshpande, H. Murat Gürsu, and W. Kellerer, "Tree-algorithms with multi-packet reception and successive interference cancellation," *IEEE Transactions on Communications*, 2023.

[15] M. Molle and A. Shih, "Computation of the packet delay in massey's standard and modified tree conflict resolution algorithms with gated access," *Tech. Rep. CSRI-264.*, 1992.

[16] P. Flajolet, X. Gourdon, and P. Dumas, "Mellin transforms and asymptotics: Harmonic sums," *Theoretical Computer Science*, vol. 144, no. 1, pp. 3–58, jun 1995. [Online]. Available: https://www.sciencedirect.com/science/article/pii/030439759500002E

[17] G. Fayolle, P. Flajolet, and M. Hofri, "On a functional equation arising in the analysis of a protocol for a multi-access broadcast channel," *Advances in Applied Probability*, vol. 18, no. 2, pp. 441–472, 1986.