A Practical Tree Algorithm with Successive Interference Cancellation for Delay Reduction in IEEE 802.16 Networks

Sergey Andreev[1], Eugeny Pustovalov[2], and Andrey Turlikov[2]

[1] Tampere University of Technology (TUT), Finland
sergey.andreev@tut.fi
[2] State University of Aerospace Instrumentation (SUAI), St. Petersburg, Russia
eugeny@vu.spb.ru, turlikov@vu.spb.ru

Abstract. This paper thoroughly studies a modification of tree algorithm with successive interference cancellation. In particular, we focus on the algorithm throughput and account for a single signal memory location, as well as cancellation errors of three types. The resulting scheme is robust to imperfect interference cancellation and is tailored to the uplink bandwidth request collision resolution in an IEEE 802.16 cellular network. The mean packet delay is shown to be considerably reduced when using the proposed approach.

Keywords: tree algorithm, successive interference cancellation, throughput, mean packet delay.

1 Introduction and Background

Contemporary communication networks adopt multi-access techniques to arbitrate the access of the user population to the shared communication link. Multiple access algorithms are specifically designed to effectively control the resource allocation and reside at the Medium Access Control (MAC) layer. They constitute an important component of the widespread wireless protocols, such as IEEE 802.11 (Wi-Fi) and IEEE 802.16 (WiMAX). Random Multiple Access (RMA) algorithms are often used due to their simple implementation and reasonably high performance.

We remind that every MAC algorithm comprises a Channel Access Algorithm (CAA) and a Collision Resolution Algorithm (CRA). Whereas the former arbitrates user access to the shared medium, the latter is responsible for the collision resolution, whenever two or more users transmit their packets simultaneously. The most widespread ALOHA-based family of algorithms includes diversity slotted ALOHA, binary exponential backoff, and other popular mechanisms. The main idea of these approaches is to specify CAA and defer the packet retransmission after a collision took place for some random future time.

By contrast, tree algorithms independently proposed in [1] and [2], focus on CRA and thus demonstrate higher efficiency. The family of conventional tree

K. Al-Begain et al. (Eds.): ASMTA 2011, LNCS 6751, pp. 301–315, 2011.

algorithms is represented by Standard Tree Algorithm (STA) and Modified Tree Algorithm (MTA). During the operation of the conventional STA and MTA, it is implicitly assumed that no meaningful information is extracted at the receiver after a collision. However, recent advances in physical (PHY) layer techniques allow using Successive Interference Cancellation (SIC) techniques [3], [4]. During SIC operation the packets involved into a collision may be restored successively. In [3], it was argued that SIC is naturally applicable to the uplink packet transmission in centralized communication networks. As such, in this paper we consider the prominent IEEE 802.16 [5] wireless cellular protocol.

The pioneering research work [6] proposes a combination of SIC and a tree algorithm (SICTA) to improve the performance of the conventional tree algorithms. Briefly, the acquired collision signals are stored in the signal memory of the receiver, which is assumed to be unbounded and then processed by SIC. Consequently, the performance of SICTA algorithm is shown to double the performance of the conventional STA. In the subsequent years, several modifications of the baseline SICTA algorithm were proposed [7], [8], [9], including the solutions that take advantage of the bounded signal memory. In particular, [10] proposes a SICTA-based algorithm to replace the standard algorithm at the bandwidth requesting stage in IEEE 802.16 networks.

All the existing SICTA-based solutions may be classified into two categories. Firstly, there are algorithms that assume perfect SIC operation and therefore are susceptible to cancellation errors falling into a deadlock. Secondly, there are algorithms that are robust to imperfect SIC operation, but at the same time are unstable when the number of users grows unboundedly. In our previous work [11], we proposed a robust SICTA algorithm that tolerates cancellation errors and demonstrates nonzero performance even when the user population is infinite. In this paper, we extend our algorithm to account for a more realistic SIC operation and conduct its thorough throughput analysis. Finally, we tailor the proposed solution for the uplink bandwidth requesting in the prominent IEEE 802.16 protocol.

2 System Model and Algorithms

2.1 Conventional Tree Algorithms

Consider tree multi-access algorithms proposed independently by [1] and [2] in the framework of classical RMA model with infinite user population and Poisson arrivals. We remind that each tree algorithm defines both CAA and CRA, which arbitrate user channel access and collision resolution process respectively. CRA is conveniently illustrated by a binary tree (see Figure 1, a), where a collided user selects right slot with probability p and selects left slot with probability $(1-p)$. Here the root corresponds to the set of users that collided initially, whereas the remaining nodes correspond to the subsets (possibly, empty) of users that decided to transmit in particular slots of the Collision Resolution Interval (CRI).

We note that in the example collision resolution tree (see Figure 1, a) a collision in slot 5 is inevitable, as none of the users which collided in slot 3 select

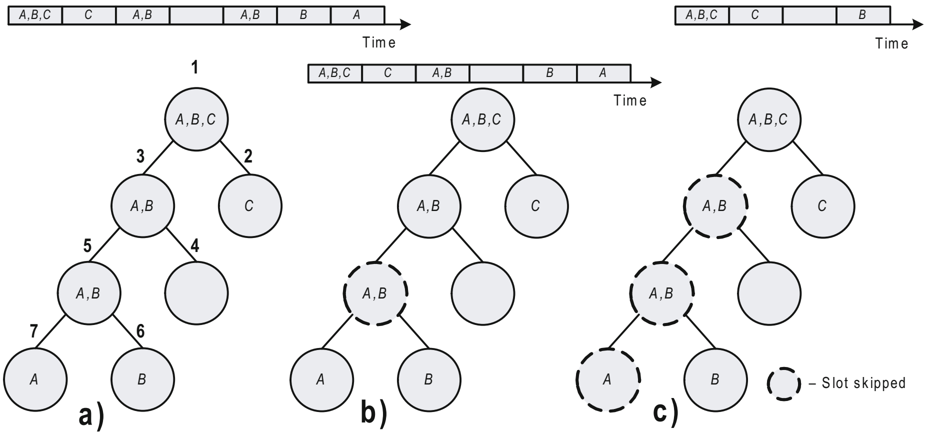

Fig. 1. Tree algorithms operation: a – STA; b – MTA; c – SICTA

empty right slot 4. As such, it is reasonable to skip the collision slot 5 and proceed immediately to the next tree level. This simple modification is adopted by the conventional MTA (see Figure 1, b) to increase the performance of the conventional STA.

Maximum stable throughput is one of the most important performance parameters of tree algorithms. It may be defined as the highest arrival rate, which still results in the bounded value of the mean packet delay. The algorithm is stable for the given arrival rate if its mean packet delay is finite. Conventional STA has the throughput of 0.346, whereas conventional MTA improves throughput up to 0.375 in the framework of the classical model.

2.2 Successive Interference Cancellation

Recent advances in telecommunication equipment allow using SIC at the PHY layer [12], [4]. Generally, SIC is an approach to process a combination of wireless signals having some additional information. Following [7], we show how SIC may improve the performance of a tree algorithm in Figure 2. Assume for simplicity that the channel is error-free. Denote by y_s the signal received by the end of slot s. Similarly, denote by x_A and x_B the signals corresponding to packets A and B respectively. Let two users transmit their packets A and B in the first slot and collide. As such, the receiver acquires the combined signal $y_1 = x_A + x_B$ and decides that a collision occurred. The initial combined signal y_1 is then stored in the signal memory of the receiver.

After acquiring the signal $y_2 = x_A$ at the end of slot 2, the receiver successfully extracts signal x_A and decodes packet A. Further, SIC procedure processes signal y_1 and cancels the extracted signal x_A from the stored combination, that is, $\tilde{y}_1 = y_1 - x_A$. Then it is also possible to extract signal $x_B = \tilde{y}_1$ and to decode

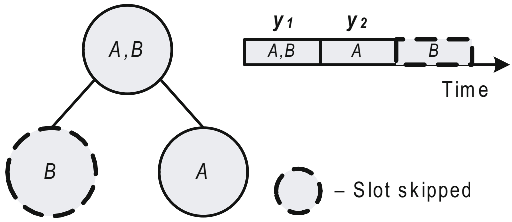

Fig. 2. Simple SIC example

packet B. Therefore, the subsequent collision resolution is not necessary. In the considered example the CRI duration is one slot less for any tree algorithm.

SIC-based algorithms typically exploit extended feedback from the PHY layer. This extra feedback is the result of the SIC operation. The baseline SICTA algorithm proposed in [6] requires the *K–EMPTY–COLLISION* feedback at MAC, where K is the number of successfully restored packets together with the number of left slots of the collision resolution tree labeled as empty after the SIC operation.

Consider example SICTA operation in Figure 1, c, where the CRI duration is only 4 slots. As left subtree may be skipped completely, the throughput is 0.693, that is, twice the STA throughput. More formal analysis is conducted below in subsection 3.2. In Figure 3, we detail the simplified SIC transceiver, which may be used to implement the baseline SICTA algorithm from [6]. Solid lines indicate transmission of data, whereas dashed lines indicate transmission of the control information. As follows from the figure, the unbounded signal memory at the receiver is practically infeasible. To mitigate this limitation, we propose our modification of SICTA that uses only a single signal memory location. As such, the implementation and operation complexity may be considerably reduced.

2.3 Imperfect Interference Cancellation

We note that in practical wireless devices using SIC, the interference cancellation is not perfect [13]. Cancellation errors may occur due to the residual signals after canceling the received signal in the stored combination. For example, after canceling the extracted signal x_A from the combined signal $x_A + x_B$ (see e.g. Figure 2) in slot s, the resulting signal contains $\tilde{y}_s = x_B + n_A$, where n_A is the residual signal x_A. After subsequent cancellation of x_B we analogously obtain $\tilde{\tilde{y}}_s = n_A + n_B$.

If the power of the residual signal $n_A + n_B$ is sufficiently high, the receiver mistakenly decides that the corresponding slot is not empty, that is, detects a non-existent collision between the users. For simplicity, we assume that this event occurs with some constant probability that depends on the receiver implementation. As such, due to the imperfect SIC operation, the PHY-MAC feedback is error-prone.

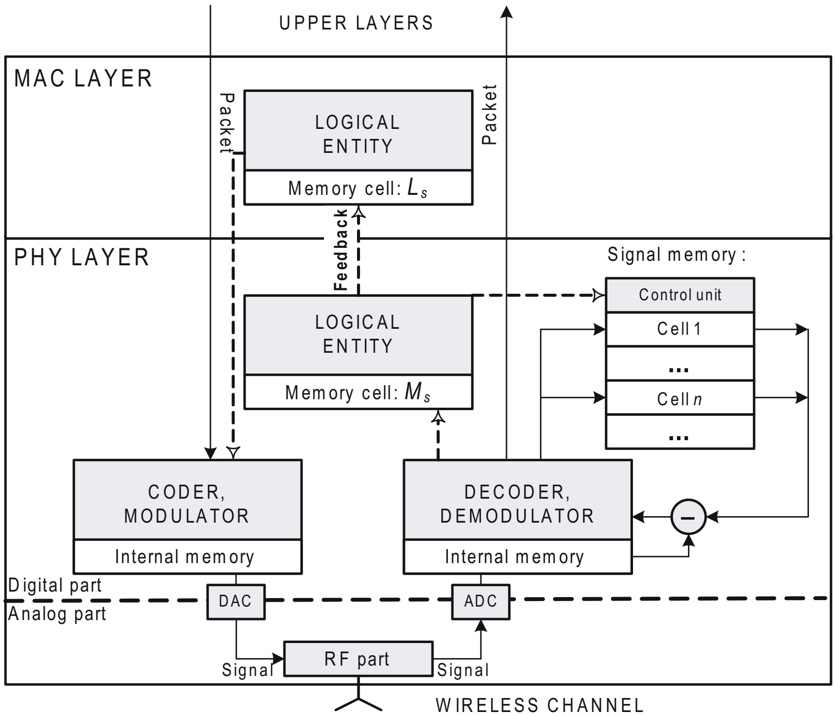

Fig. 3. SIC transceiver structure

Despite its high throughput, the baseline SICTA algorithm is vulnerable to cancellation errors. Indeed, assume that in the considered example in Figure 1, c the last cancellation operation of the signal x_B in the initial stored signal implies the residual signal n_B, that is, $\tilde{\tilde{y}}_1 = y_1 - x_C - x_B + n_B$. If now the power of n_B is high enough, the signal x_A may be extracted unsuccessfully and the collision resolution process would continue. Sooner or later, when packet A is decoded successfully, the residual signal power may be too high and the receiver would detect another non-existent collision in the left slot. According to the SICTA rules, the collision resolution process would then continue indefinitely unless it is aborted externally. As such, SICTA falls into a deadlock.

Our overview of SICTA-based algorithms indicates that currently there is no algorithm that is robust to imperfect interference cancellation and, at the same time, stable in the framework of the classical RMA model. The proposed robust SICTA (R-SICTA) algorithm tolerates cancellation errors for the cost of some reduction in its throughput. Additionally, it takes advantage of the single signal memory location at the receiver side. The increase in the amount of the available memory will result in growing throughput (still below the throughput of SICTA) and complexity [9].

The main idea of the proposed algorithm (see Figure 4, a) is to refrain from skipping particular collision slots (such as slot 3), which otherwise might result

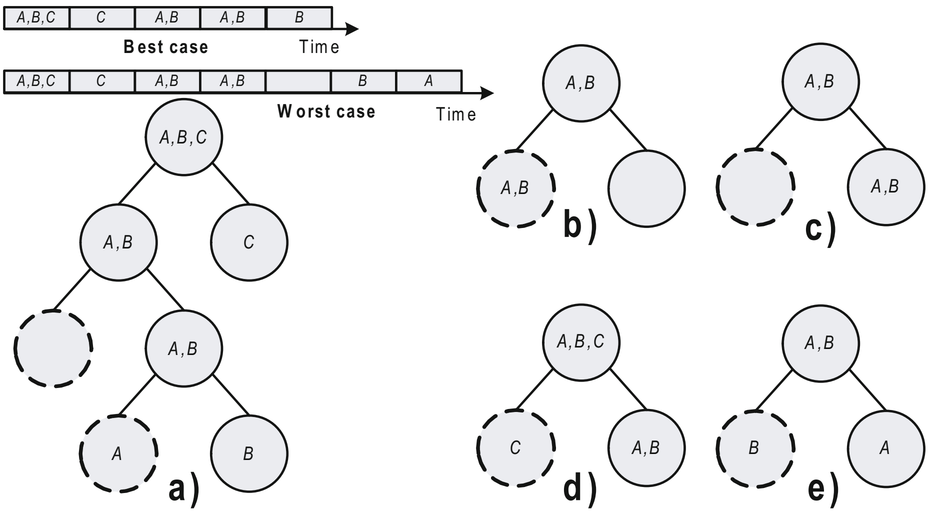

Fig. 4. Proposed R-SICTA algorithm: a – example operation of R-SICTA; b, c, d, e – some subtrees of R-SICTA

in a deadlock. In Figure 4, a, the time diagram for the best case corresponds to two successful cancellation operations, whereas the time diagram for the worst case corresponds to two unsuccessful operations. The formal description of our R-SICTA algorithm is given below in subsection 3.3. We note that due to cancellation errors the rules 3, 5, and 7 (see Table 1) must not allow skipping a slot of the collision resolution tree. Otherwise, deadlock effect may not be controlled.

3 Proposed Throughput Calculation Technique

3.1 General Procedure

Here we describe an approach to obtain the throughput of a tree algorithm with successive interference cancellation. As such, we develop the re-calculation method for the mean collision resolution time from [14] and apply it to SICTA-based algorithms. Denote by v the duration of CRI in slots (time to resolve a collision of size k, or the number of nodes in the respective collision resolution tree), which is a discrete random variable. The conditional mean $E[v|\text{collision of size } k \text{ is being resolved}]$ determines the CRI duration for a collision between k users. Our approach to the throughput calculation of tree algorithms using SIC is based on the following auxiliary assumptions.

Proposition 1. *Consider tree algorithm A using SIC. Denote by T_k^A the mean time to resolve a collision of size k for this algorithm. Taking $\frac{k}{T_k^A}$ into account, the following bounds for the throughput of A denoted by R_A may be established:*

$$\liminf_{k\to\infty} \frac{k}{T_k^A} < R_A < \limsup_{k\to\infty} \frac{k}{T_k^A}. \tag{1}$$

The proof of this proposition follows immediately from [1]. Consider now STA denoting the respective mean collision resolution time by T_k. Analogously to (1), we may derive the bounds for the throughput of STA omitting the lower index and denoting it simply by R:

$$\liminf_{k\to\infty} \frac{k}{T_k} < R < \limsup_{k\to\infty} \frac{k}{T_k}. \tag{2}$$

We note that the bounds for R were established in [15] and are equal to:

$$0.34657320 < R < 0.34657397. \tag{3}$$

Proposition 2. *Upper and lower bounds for the throughput R of STA may be obtained as:*

$$\left(\frac{2}{\ln 2} + c\right)^{-1} < R < \left(\frac{2}{\ln 2} - c\right)^{-1}, \tag{4}$$

where $c = 3.127 \cdot 10^{-6}$.

The interrelation between the bounds from (3) and expression (4) was established in [14].

Proposition 3. *Denoting the number of nodes in the collision resolution tree of STA by v, we readily obtain $T_k = E[v]$. The number of success, collision, and empty slots within CRI is denoted by v_s, v_c and v_e respectively. Then $v_s+v_c+v_e = v$ and at the same time:*

$$v_s = k; \quad v_c = \frac{v-1}{2}; \quad v_e = \frac{v+1}{2} - k. \tag{5}$$

The proof of the above relations may be found in [14].

Proposition 4. *Consider the collision resolution tree of algorithm A as the collision resolution tree of STA, where the time to pass some tree nodes is zero due to the SIC operation. We establish the number of thus skipped nodes r denoting by u the number of remaining nodes. Coming to the expected values, we may write:*

$$E[u] = E[v] - E[r] \quad \textit{or} \quad T_k^A = T_k - E[r]. \tag{6}$$

The proof of this proposition follows from Proposition 1 and equation (2). Substituting (6) into (1) and after some transformations, we derive the bounds for the throughput R_A as functions of known bounds for the throughput of STA (4). Below we use Proposition 3 to obtain $E[r]$ for the baseline SICTA algorithm, as well as for the proposed R-SICTA algorithm. Also we account for Proposition 4 to establish bounds for their throughputs.

3.2 Baseline SICTA Algorithm

Below we calculate the throughput of the baseline SICTA algorithm from [6] as an example. The following feedback must be available from the SIC receiver for the proper operation of SICTA:

1. *COLLISION.*
2. *EMPTY.*
3. K – the number of successfully restored packets together with the number of left slots of the collision resolution tree labeled as empty after SIC operation ($K \geq 1$).

The operation of the respective PHY layer is detailed in Figure 3.

Proposition 5. *The mean number of non-skipped nodes in the collision resolution tree of SICTA (T_k^S) is established as:*

$$T_k^S = T_k - \frac{1}{2}E[v_s] - \frac{1}{2}(E[v_c] - 1) - \frac{1}{2}E[v_e] = T_k - \frac{1}{2}T_k + \frac{1}{2} = \frac{T_k + 1}{2}, \quad (7)$$

where T_k is the mean number of nodes in the collision resolution tree of STA.

The thorough proof of this proposition may be conducted using the approach from [14] and describing the performance of SICTA in the framework of the graph theory.

Consequently, accounting for (1) and (4), we establish the following bounds for the throughput R_S of SICTA:

$$\left(\frac{1}{\ln 2} + c\right)^{-1} < R_S < \left(\frac{1}{\ln 2} - c\right)^{-1}. \quad (8)$$

Concluding our analysis, we note that upper and lower bounds for the throughput of SICTA are sufficiently close to each other due to small value of c. Therefore, SICTA throughput may be written as:

$$R_S \approx \ln 2 \approx 0.693, \quad (9)$$

which gives the previously known result from [6]. However, our approach if far simpler and has reduced computational burden.

3.3 Proposed R-SICTA Algorithm

Consider now the proposed R-SICTA algorithm, which is robust to imperfect interference cancellation. This algorithm stores a single collision signal and cancels both success and collision signals (see Figure 4, a). It is also designed to tolerate cancellation errors (see subsection 2.3).

The following feedback must be available from the SIC receiver for the proper operation of R-SICTA:

1. *COLLISION* and slot skip (contents of the left slot extracted) (C/skip).
2. *COLLISION* and no slot skip (C/–).
3. *SUCCESS/EMPTY* and no slot skip (SE/–).
4. *SUCCESS* and slot skip (contents of the left slot extracted) (S/skip).
5. *EMPTY* and slot skip (inevitable collision in the following slot) (E/skip).

Ensure: During algorithm operation, account for a particular CAA.
1: Reset position L of a user in collision resolution tree.
2: Generate new arrivals to the user according to a particular arrival flow.
3: **if** user has pending packets **then**
4:     **if** position of user $L = 0$ **then**
5:         Transmit a packet.
6: Wait for end of the current slot.
7: Receive feedback from PHY.
8: **if** C/skip feedback is received **then**
9:     **if** $L = 1$, **then**
10:         Delete the pending packet.
11:     **else if** $L = 0$, **then**
12:         $L = \begin{cases} 0 & \text{with probability } p, \\ 1 & \text{with probability } 1 - p. \end{cases}$
13: **else if** C/– feedback is received **then**
14:     **if** $L > 0$, **then**
15:         $L = L + 1$.
16:     **else**
17:         $L = \begin{cases} 0 & \text{with probability } p, \\ 1 & \text{with probability } 1 - p. \end{cases}$
18: **else if** SE/– feedback is received **then**
19:     **if** $L > 0$, **then**
20:         $L = L - 1$.
21:     **else**
22:         Delete the pending packet.
23: **else if** S/skip feedback is received **then**
24:     **if** $L \geq 2$, **then**
25:         $L = L - 2$.
26:     **else**
27:         Delete the pending packet.
28: **else if** E/skip feedback is received **then**
29:     **if** $L = 1$, **then**
30:         $L = \begin{cases} 0 & \text{with probability } p, \\ 1 & \text{with probability } 1 - p. \end{cases}$
31: Go to step 2.

Algorithm 1. R-SICTA MAC operation

Algorithm 1 describes the operation of R-SICTA MAC. Denote the received signal by cs, the stored signal by ss, and the extracted signal by ms. The PHY operation is summarized in Table 1.

Below we express T_k^{RS} according to Proposition 4. As such, the following tree nodes are subtracted from T_k as they are skipped due to SIC operation:

- Figure 4, b, when a collision slot is followed by an empty slot. The MTA rules allow to skip a collision slot with probability 1.
- Figure 4, c, when a collision slot is followed by a collision slot. The SIC operation allows to skip an empty slot with probability $1 - q_{ce}$.

Table 1. R-SICTA PHY operation

Rule	Channel – PHY	Memory contents	PHY – MAC	Store
1	*COLLISION*	$ss - cs = 0$	C/skip	cs
2	*COLLISION*	$ss - cs = ms$	C/skip	cs
3	*COLLISION*	otherwise	C/-	cs
4	*SUCCESS*	$ss - cs = ms$	S/skip	0
5	*SUCCESS*	otherwise	SE/-	0
6	*EMPTY*	$ss \neq 0$	E/skip	ss
7	*EMPTY*	$ss = 0$	SE/-	0

– Figure 4, d, when a collision slot is again followed by a collision slot. The SIC operation allows to skip a success slot with probability $1 - q_{cs}$.
– Figure 4, e, when a collision slot is followed by a success slot. The SIC operation allows to skip a success slot with probability $1 - q_{ss}$.

We note that q_{ce}, q_{cs}, and q_{ss} are the parameters of SIC and depend on its implementation. The following proposition may thus be formulated.

Proposition 6. *The mean number of non-skipped nodes in the collision resolution tree of R-SICTA (T_k^{RS}) is established as:*

$$T_k^{RS} = \left(\frac{1}{2} + \frac{q_{ce}}{4}\right) T_k - \frac{1}{2} + \frac{q_{ce}}{4} + \frac{k}{2}(1 + q_{cs} - q_{ce}) + \frac{N_k}{2}(q_{ss} - q_{cs}), \qquad (10)$$

where T_k is the mean number of nodes in the collision resolution tree of STA and N_k is the number of collisions of size 2 in the STA collision resolution tree of initial size k, whereas probabilities q_{ce}, q_{ss}, and q_{cs} were discussed above.

We omit the proof of this proposition due to the space limitations.

Finalizing our analysis, it is important to evaluate the relation $\frac{N_k}{k}$ for the increasingly large values of k. Clearly, $N_0 = N_1 = 0$ as there are no collision nodes in the respective collision resolution trees. Also it is easy to show that $N_2 = 2$. Generally, accounting for the properties of collision resolution trees, for $k > 2$ it follows that:

$$N_k = \frac{\sum_{i=1}^{k-1} \binom{k}{i} N_i}{2^{k-1} - 1}. \qquad (11)$$

Equation (11) may be calculated recursively for any chosen value of k.

Consider the Poisson transform [16] of the sequence $N_0, N_1, \ldots, N_i, \ldots$, which is denoted by:

$$N(s) \triangleq \sum_{k \geq 0} N_k \cdot \frac{s^k}{k!} e^{-s}, \;\; s \in \mathcal{R}. \qquad (12)$$

After some derivations, we may obtain the following recursive expression to establish $N(s)$:

$$N(s) = 2N\left(\frac{s}{2}\right) + \frac{s^2}{2}e^{-s}. \tag{13}$$

Consider also the normalized Poisson transform analogously to [15], which is denoted by:

$$M(s) \triangleq \frac{N(s)}{s}. \tag{14}$$

Using (13), we may rewrite (14) as:

$$M(s) = M\left(\frac{s}{2}\right) + \frac{s}{2}e^{-s}. \tag{15}$$

The function $M(s)$ is periodic for large values of its argument, which may be used to evaluate it. We consider the normalized Poisson transform for sufficiently large values of its argument $2^n r$, where $n \in \mathcal{Z}$ and $r \in \mathcal{R}$. Formally substituting $2^n r$ into (15), we obtain:

$$M(2^n r) = M(2^{n-1} r) + \frac{2^n r}{2}e^{-2^n r}. \tag{16}$$

For considerably large n, the variation of $M(2^n r)$ from 2^n to 2^{n+1} corresponds to one period of function $M(2^n r)$. Therefore, for $1 \leq r \leq 2$ and some value of n we obtain the highest and the lowest values of the function for all the subsequent values of its argument. Consider the equality (16) in more detail and execute the recursion for $n - 1$ times:

$$M(2^n r) = M(r) + \sum_{i=1}^{n} \frac{2^i r}{2} e^{-2^i r} = M(r) + H_n(r). \tag{17}$$

The series $H_n(r)$ converge fast and easy to evaluate with any required precision. The values of $M(r)$ for low r are easy to obtain accounting for (12) and (14). Further, using (17), we study the behavior of the initial function $M(2^n r)$ over one period, when $n \geq 20$. It may be shown that the precision of the obtained values is then at least 10^{-8}. We give these values in the integer points k, that is, $M(2^n r) = M(k)$:

$$\max_{2^{20} \leq k \leq 2^{21}} M(k) = \limsup_{k \to \infty} M(k) < 0.72135464 + 1 \cdot 10^{-8} \tag{18}$$

and

$$\min_{2^{20} \leq k \leq 2^{21}} M(k) = \liminf_{k \to \infty} M(k) > 0.72134039 - 1 \cdot 10^{-8}.$$

Following the approach from [17] (Theorem 1), it may be shown that the lower and the upper bounds for $M(k)$ (18) also hold for the relation $\frac{N_k}{k}$. We note that as the limit of $\frac{N_k}{k}$ does not exist, we inevitably have an interval, where the behavior of $\frac{N_k}{k}$ is not determined. The length of this interval, however, is only 0.00001425.

Simplifying the representation of the final result, we note that $\limsup_{k\to\infty} \frac{N_k}{k} = \liminf_{k\to\infty} \frac{N_k}{k} = \gamma$ with the precision of at least three decimal digits and $\gamma = 0.721$. Also for the sake of simplicity, we avoid finding the upper and the lower bounds for the throughput of R-SICTA (R_{RS}) as they are sufficiently close to each other. Then the resulting approximation for the throughput of the proposed algorithm may be obtained as:

$$R_{RS} \approx \frac{2\ln 2}{2 + q_{ce} + 2\ln 2(1 + q_{cs} - q_{ce} + (q_{ss} - q_{cs})\gamma)}. \tag{19}$$

In particular, when $q_{ce} = q_{ss} = q_{cs} = 0$, that is, when there are no cancellation errors, $R_{RS} \approx 0.515$.

4 IEEE 802.16 Performance Improvement and Conclusions

The combination of successive interference cancellation at PHY and tree algorithms at MAC constitutes a promising direction toward the improvement of the contemporary communication protocols. In particular, it allows for the considerable throughput increase for the moderate cost of implementation and operation complexity. Currently, the family of SIC-based algorithms is known, where the baseline SICTA demonstrates the highest throughput of 0.693 under the classical set of assumptions. However, SICTA requires unbounded signal memory at the receiver side, which is practically infeasible. Moreover, its performance degrades significantly due to the imperfect interference cancellation.

We proposed a practical SICTA-based algorithm that mitigates the limitations of the baseline SICTA and has the throughput of 0.515 in case of no cancellation errors. Moreover, our R-SICTA is robust even in case of high cancellation error probability and demonstrates graceful performance degradation. In the worst case, when SIC operation is not possible, R-SICTA converges to MTA with the throughput of 0.375. In order to conclude upon the feasibility of the proposed algorithm, we consider its usage for the uplink bandwidth requesting in the prominent IEEE 802.16 protocol.

Below we evaluate the gain after the replacement of the standardized Binary Exponential Backoff (BEB) algorithm [5] with the proposed R-SICTA. We use our own IEEE 802.16 simulator extensively verified over the recent years and referenced by our previous works, e.g. [18]. The simulation parameters are summarized in Table 2. In Figure 5, we plot the reservation delay as the function of the overall arrival rate. Here M stands for the total number of users and N is the number of bandwidth request slots per frame. As expected, the baseline SICTA algorithm with unbounded signal memory demonstrates the lowest delay. However, the proposed R-SICTA algorithm with single signal memory performs closely to SICTA in case of no cancellation errors.

Table 2. Basic simulation parameters

IEEE 802.16 network parameter	Value
DL:UL	60:40
PHY type	OFDM
Frame duration	5 ms
Sub-channel bandwidth	7 MHz
Contention slot length	170 μs
Data packet size	4096 bits

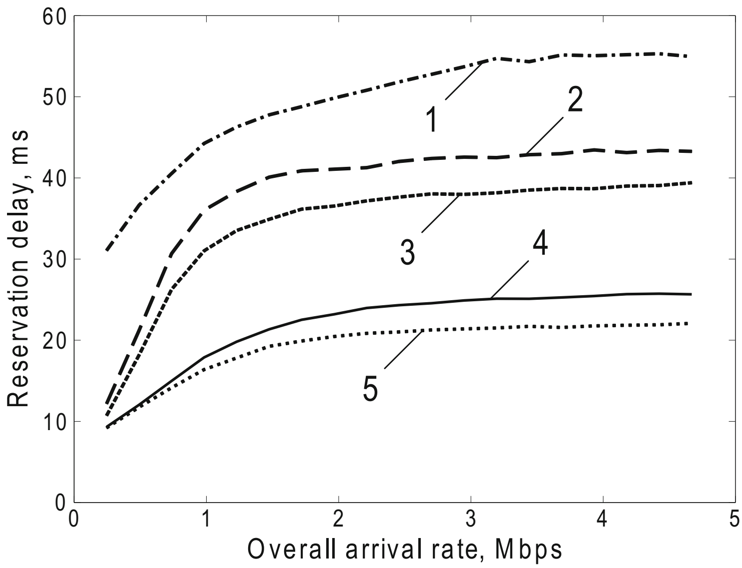

Fig. 5. IEEE 802.16 reservation delay for $M = 6$ and $N = 1$: 1 – BEB, $W = W_0$, $m = 0$; 2 – STA; 3 – MTA; 4 – R-SICTA, $q_{ce} = q_{ss} = q_{cs} = 0$; 5 – SICTA

In the worst case of imperfect interference cancellation, when $q_{ce} = q_{ss} = q_{cs} = 1$, the R-SICTA reservation delay approaches that of MTA. As such, the gap between curves 3 and 4 in Figure 5 demonstrates the range of possible gains from the proposed solution depending on the cancellation error probability. Finally, STA has higher reservation delay than MTA, whereas the standardized BEB algorithm is clearly the worst case even with the optimal operation parameters W and m [18].

Figure 6 plots the overall packet delay in IEEE 802.16. The results generally follow the respective trends as in Figure 5, but show different delay values. In particular, for the case without cancellation errors, the proposed R-SICTA has the gain of 50-60% comparatively to the standardized BEB algortihm and depending on the arrival rate. In the worst case of imperfect interference

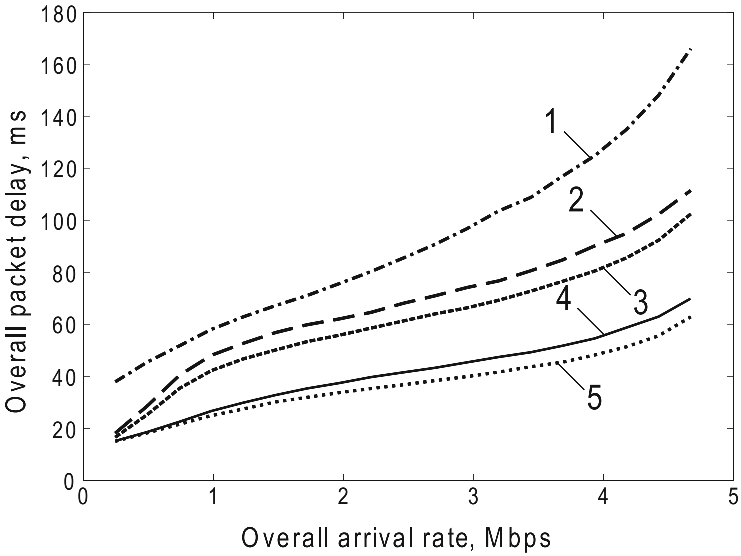

Fig. 6. IEEE 802.16 overall delay for $M = 6$ and $N = 1$: 1 – BEB, $W = W_0$, $m = 0$; 2 – STA; 3 – MTA; 4 – R-SICTA, $q_{ce} = q_{ss} = q_{cs} = 0$; 5 – SICTA

cancellation, when $q_{ce} = q_{ss} = q_{cs} = 1$, the gain reduces to 25-56%, but still remains considerable.

Summarizing, we have thoroughly analyzed the proposed practical R-SICTA algorithm with successive interference cancellation. We established its throughput, as well as tailored it to the uplink bandwidth requesting in the contemporary IEEE 802.16 protocol. Our results indicate significant delay gains after the replacement of the standardized BEB algorithm with the proposed R-SICTA. As such, the proposed solution is attractive to improve the performance of modern cellular networks.

Acknowledgments

This work is supported by the Russian Foundation for Basic Research (project # 10-08-01071-a).

References

1. Tsybakov, B.S., Mikhailov, V.A.: Free synchronous packet access in a broadcast channel with feedback. Problems of Information Transmission 14(4), 32–59 (1978)
2. Capetanakis, J.I.: Tree algorithms for packet broadcast channels. IEEE Transactions on Information Theory 25(4), 505–515 (1979)
3. Pedersen, K., Kolding, T., Seskar, I., Holtzman, J.: Practical implementation of successive interference cancellation in DS/CDMA systems. In: Proceedings of IEEE ICUPC, vol. 1, pp. 321–325 (1996)

4. Weber, S., Andrews, J., Yang, X., Veciana, G.: Transmission capacity of wireless ad hoc networks with successive interference cancellation. IEEE Transactions on Information Theory 53(8), 2799–2814 (2007)
5. IEEE Std. 802.16m (D9), Amendment to IEEE Standard for Local and metropolitan area networks. Advanced Air Interface
6. Yu, Y., Giannakis, G.B.: SICTA: A 0.693 contention tree algorithm using successive interference cancellation. In: Proceedings of IEEE INFOCOM, vol. 3, pp. 1908–1916 (2005)
7. Yu, Y., Giannakis, G.B.: High-throughput random access using successive interference cancellation in a tree algorithm. IEEE Transactions on Information Theory 53(12), 4628–4639 (2007)
8. Wang, X., Yu, Y., Giannakis, G.B.: A robust high-throughput tree algorithm using successive interference cancellation. IEEE Transactions on Communications 55(12), 2253–2256 (2007)
9. Peeters, G.T., Houdt, B.V.: Interference cancellation tree algorithms with k-signal memory locations. IEEE Transactions on Communications 58(11), 3056–3061 (2010)
10. Wang, X., Yu, Y., Giannakis, G.B.: Combining random backoff with a cross-layer tree algorithm for random access in IEEE 802.16. IEEE Wireless Communications and Networking Conference 2, 972–977 (2006)
11. Andreev, S., Pustovalov, E., Turlikov, A.: SICTA modifications with single memory location and resistant to cancellation errors. In: Balandin, S., Moltchanov, D., Koucheryavy, Y. (eds.) NEW2AN 2008. LNCS, vol. 5174, pp. 13–24. Springer, Heidelberg (2008)
12. Agrawal, A., Andrews, J., Cioffi, J., Meng, T.: Iterative power pontrol for imperfect successive interference cancellation. IEEE Transactions on Wireless Communications 4(3), 878–884 (2005)
13. Andrews, J., Hasan, A.: Analysis of cancellation error for successive interference cancellation with imperfect channel estimation. Tech. Rep. EE–381K: Multiuser Wireless Communications (2002)
14. Evseev, G.S., Turlikov, A.M.: A connection between characteristics of blocked stack algorithms for random multiple access system. Problems of Information Transmission 43(4), 345–379 (2007)
15. Gyorfi, L., Gyori, S., Massey, J.L.: Principles of stability analysis for random accessing with feedback. NATO Security through Science Series: Information and Communication Security 10, 214–250 (2007)
16. Szpankowski: Average Case Analysis of Algorithms on Sequences. Wiley, Chichester (2001)
17. Gyorfi, L., Gyori, S.: Analysis of tree algorithm for collision resolution. In: Proceedings of AofA, pp. 357–364 (2005)
18. Andreev, S.D., Turlikov, A.M., Vinel, A.V.: Contention-based polling efficiency in broadband wireless networks. In: Al-Begain, K., Heindl, A., Telek, M. (eds.) ASMTA 2008. LNCS, vol. 5055, pp. 295–309. Springer, Heidelberg (2008)