ISSN 0005-1179, Automation and Remote Control, 2009, Vol. 70, No. 3, pp. 417–433. © Pleiades Publishing, Ltd., 2009.
Original Russian Text © S.D. Andreev, E.M. Pustovalov, A.M. Tyurlikov, 2009, published in Avtomatika i Telemekhanika, 2009, No. 3, pp. 78–96.

QUEUEING SYSTEMS

Conflict-resolving Tree Algorithm Stable to Incomplete Interference Damping

S. D. Andreev, E. M. Pustovalov, and A. M. Tyurlikov

State University of Aerospace Instrumentation, St. Petersburg, Russia
Received August 7, 2008

Abstract—Consideration was given to an approach uniting the tree algorithm to resolve conflicts at the channel level of a multisite communication system with the procedure for successive cancelation of interference at the physical level. Proposed was a unit-memory algorithm which is stable to incomplete interference cancelation within the framework of the classical model of multiple access with the infinite number of subscribers. A simple procedure of speed calculation which is applicable to the class of the tree algorithms featuring successive interference cancelation was demonstrated by the example of its analysis. The proposed algorithm may be used as an alternative to the scheme of resolution of conflicts for the channel resource defined in the IEEE 802.16 standard.

PACS numbers: 02.50.-r, 02.50.Ey, 02.50.Fz
DOI: 10.1134/S0005117909030084

1. INTRODUCTION

Approaches combining the capabilities of two or more levels of the reference model of interaction of open systems enable one to obtain a higher effectiveness than the traditional interaction of levels. In particular, the joint operation of the *physical* level and the sublevel of *control with access to environment* (CAE) is sufficiently promising because CAE is the "bottleneck" of the current telecommunication systems. For the time being, there are many algorithms of the CAE sublevel [1, 2] which allow the designer seeking a suitable decision to exercise a considerable option. The *random multiple access* (RMA) algorithms that are often used in the communication systems with spike traffic feature sufficiently low data *packet* delay even in the case of an unrestrictedly large number of system subscribers.

Each RMA algorithm includes, generally speaking, a *channel access algorithm* (CAA) and a *conflict resolution algorithm* (CRA). While the first algorithm regulates the procedure of subscriber access to the common transmission environment, the second algorithm defines a rule for resolution of conflicts between the data packets arising at simultaneous transmission from two or more subscribers. In the family of ALOHA-based algorithms [3] such as the *clocked ALOHA with repetition*, *double exponential "backoff,"* and *multiple access with listening of the carrier frequency*, there is no CRA. These algorithms are notable for a sufficiently simple realization, and their main concept lies in that in the case of conflict the repeated packet transmission is backed off for some random time interval.

On the contrary, the tree RMA algorithms proposed independently in [4, 5] define CRA and thereby attain a higher performance as compared with the aforementioned algorithms. It is now the practice to distinguish between the *standard tree algorithm* (STA) and the *modified tree algorithm* (MTA). In what follows, we refer to STA and MTA as the *traditional* tree algorithms.

In the traditional tree RMA algorithms, it is implied that in the case of conflict (interference) between the subscribers at the receiving side no sensible information can be extracted. However, it became possible to use at the physical level a procedure for *successive interference "cancelation"* (SIC) [6–9]. The SIC procedure allows one to restore by turns the conflicting packets (signals) from various subscribers of a wireless communication channel. Nevertheless, the SIC procedure is recommendable only for the ascending channel of the centralized communication networks because it requires the information accessible at the base station [6]. Therefore, this approach may be easily used in the regional wireless communication networks of the IEEE 802.16 standard [10].

An algorithm combining the traditional tree algorithm with the advantages of the SIC procedure, the SIC in a tree algorithm (SICTA) was first proposed in [11]. The same paper described and analyzed a new *protocol* where the SIC procedure processes the received signals with information about the conflict. These *conflict signals* are kept until processing in the receiver memory which is assumed in [11] to be unlimited. Such *base* SICTA algorithm is twice as efficient as the traditional STA.

After that, several modifications of the base algorithm SICTA including those using on the receiving side one memory cell for signal storage were considered [12–16]. In particular, an alternative to the standard algorithm of binary exponential "backoff" was considered in [13]. It regulates resolution of conflicts between the requests for resources of the descending channel in the ascending channel of the networks of the type of IEEE 802.16. All the above algorithms can be conventionally classified with two groups: unstable to incomplete interference cancelation, that is, subject to the effect of *locking* due to the errors of cancelation, and stable to incomplete interference cancelation but inoperable in the model with infinite number of subscriber (unsteady). The present paper fills this gap by a rather readily realizable stable unit-memory algorithm which is steady in the model with infinite number of subscribers.

The following section formulates the assumptions of the *classical* RMA model and proposes its extension with regard to the characteristics of the SIC procedure. Necessary definitions are introduced, and some algorithms are described. The third section demonstrates an approach to the analysis of the SICTA-based algorithms by the example of the proposed algorithm. The fourth section compares the analytical results with the earlier results and discusses the specifics of practical realization of the proposed algorithm.

2. SYSTEM MODEL AND DESCRIPTION OF ALGORITHMS

2.1. Classical Model of the Multiple Access System

We follow numerous works such as [1, 2, 17] to formulate some assumptions about the RMA communication system and the rules for its subscribers. These assumptions may be regarded as a *classical* model of a multiple access system. Despite some simplifying assumptions, this model enables one to compare different algorithms from a the same standpoint. The following assumptions are discussed in detail in [1].

Assumption 1. *Synchronous system.* The time of system operation is divided into equal intervals called the *slots*. All transmitted packets have identical length and are transmitted precisely in one slot. The slots are numerated by nonnegative integers, and the interval $[t, t+1)$ corresponds to the tth slot. For brevity, we call in what follows the tth slot simply the slot t. The boundaries of slots are known to all system subscribers, and each subscriber can start transmission of its packet only at the beginning of the next slot.

Assumption 2. *Informativeness of the communication channel.* One and only one of the following *events* can occur in each slot:

- packet is transmitted only by one subscriber ("SUCCESS");

- none of the subscribers transmits packet ("EMPTY");
- packets are transmitted simultaneously by two or more subscribers ("CONFLICT").

Assumption 3. *Type of feedback.* The information about any of the three events in the slot (see Assumption 2) arrives to the subscriber ACE sublevel by the *end* of the slot.

Assumption 4. *Feedback confidence.* The *feedback* information (see Assumption 3) arriving to the subscriber CAE sublevel is *confident.*

Assumption 5. *Infinite number of subscribers.* The system has *infinite* number of subscribers each with a buffer to store one data packet. The packet is stored from the instant of its arrival to the subscriber until the end of its successful transmission. The time intervals between two successive instants of data packet occurrences are statistically independent random variables distributed exponentially with the parameter $\frac{1}{\lambda}$, where λ is the *intensity* of the inflow of new packets to the system. Each new packet arrives to a new subscriber, that is, the data packets and destination subscribers are identical.

2.2. Traditional Tree Algorithms

We recall that each tree algorithm defines CAA and CRA regulating, respectively, the order of packet transmission to the channel and the process of resolution of the arising conflicts. We describe them beginning from the latter. At occurrence of a conflict in the tth slot, the conflicting subscribers are decomposed into two subsets. For example, each of the conflicting subscribers "throws an (incorrect) coin" and selects the first subset with the probability p or the second with the probability $(1-p)$. The subscribers of the first subset transmit their packets in the slot $t+1$; if this slot was empty or had a successful transmission, then the subscribers of the second subset transmit in the slot $t+2$. Otherwise, if one more conflict arises in the slot $t+1$, the first subset again is decomposed into two subsets, and the above actions are repeated for them, whereas the second subset "waits" until the conflict between the subscribers of the first subset is resolved.

The above CRA may be conveniently represented as a binary tree (see Fig. 1a) where the root vertex corresponds to the set of subscribers entering into the initial conflict. The rest of the vertices

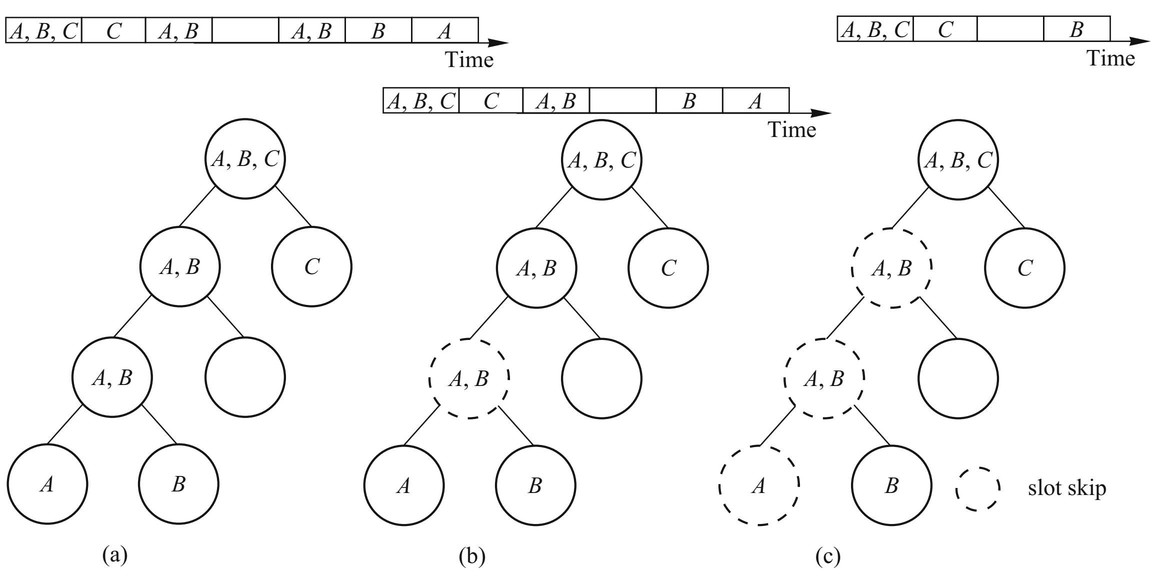

Fig. 1. Examples of operation: (a) STA, (b) MTA, (c) SICTA.

correspond to the (possibly, empty) subsets of the subscribers transmitting their packets in each slot of the *conflict resolution period* (CRP), that is, until the final success or empty window. The tree edges reflect the process of decomposition, that is, two branches "grow" from the vertices with two or more subscribers. At that, the *right* vertex of this subtree corresponds to the first subset of the decomposition, and the *left*, to the second. Since each vertex of the tree defines uniquely the slot within which its subscriber subset is transmitting, we use for brevity the terms left and right slot of the *conflict resolution tree* (CRT).

We note that in the example of CRT (see Fig. 1a) a conflict in slot 5 is inevitable because an empty slot 4 follows the conflict in slot 3, which means that all conflicting subscribers choose the left subtree. Consequently, it is advisable to skip the slot with an inevitable conflict (slot 5) and go directly to the next CRT level. The collection of the aforementioned rules defines the CRA for the traditional MTA (see Fig. 1b) within the framework of the classical model of the RMA system. To formalize CAA, it is necessary to define the rules for packet transmission by the subscribers not involved in conflict resolution.

Each of the tree algorithms can use any of the three alternative CAA's: *gateway, window*, or *nonblocked*. With the gateway CAA, transmission of new packets arriving during some CRP is postponed until the beginning of the next one. As soon as the current conflicts are resolved, all the delayed packets are transmitted simultaneously. If there are two or more such packets, then inevitably a new conflict occurs and gives rise to a corresponding CRP. The window CAA generalizes the scheme with gateway access for the case where a new CRP is formed with participation not of all packets delayed at the preceding CRP but only of their part arriving during some time window. Efficiency of the algorithm may be improved by an appropriate choice of the window size control. The gateway and window access algorithms are often called the *blocked* algorithms or the algorithms with blocked access. Finally, in the nonblocked algorithms a new data packet is transmitted immediately upon its arrival. At that, the notion of "CRP" does not arise, and realization of the algorithm becomes simpler because addition of a new subscriber to the system requires no preliminary observation of the events in the communication channel. The nonblocked algorithms are sometimes called the free-access algorithms.

Speed is one of the main characteristics of any tree algorithm. The maximum *speed* of the steady transmission is defined as the lowest upper boundary of the intensities (see Assumption 5) of the inflow to the system of new packets under which the system is steady. The multiple-access system is regarded as *steady* under a given intensity of the input flow if the mean (limit, or time-mean, or ensemble-mean when time tends to infinity) packet delay is finite [1]. The traditional MTA with gateway access [4, 5] reaches the speed of 0.375 under "fair" decomposition of the subscribers with $p = \frac{1}{2}$. For the given algorithm, however, the "unfair" decomposition with $p = 0.582$ is optimal, which enables the speed of 0.381 [18]. Despite an increase in speed at optimization of the parameter p, the present paper confines itself for simplicity to analyzing only the algorithms with the gateway access and fair decomposition. Some remarks about the unblocked access will be made in the conclusion.

2.3. Successive Cancelation of Interference

The progress of the telecommunication equipment enabled one to use the procedure of successive interference cancelation (SIC) at the physical level of the communication networks [6–9]. We follow [12] in considering a simplistic example of Fig. 2 to gain an insight into how the SIC procedure enables one to increase effectiveness of the tree algorithm. Let in some slot t two subscribers transmit conflicting packets A and B. We assume also that there is no noise in the communication channel. We denote by y_t the signal received by the end of slot t and by x_A, the signal corresponding

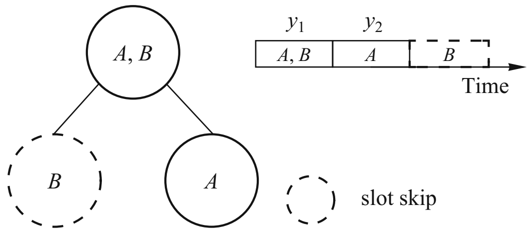

Fig. 2. Example of the interference cancelation procedure.

to the packet A. Having received the signal $y_2 = x_A$ at the end of the second slot, the receiver with SIC successfully decodes the packet A. Then, the SIC procedure cancels the interference of the signal x_A with the signal x_B in the first slot, that is, $x_B = \widetilde{y}_1 = y_1 - x_A$. Therefore, the packet B also is successfully restored and no further conflict resolution is required. In this example the length of CRP is one slot less than with any traditional tree algorithm.

We note that the SIC procedure is used most frequently in iterative decoding of a conflicting (composite) signal [6, 9]. In particular, from the received composite signal the receiver with SIC first extracts and decodes the most powerful signal. Then, the signal is again coded and "subtracted" from the original one. Therefore, a new composite signal is obtained, and the procedure of interference cancelation is iterated.

Applicability of this SIC procedure is substantially limited by the need for coding the signal at the receiving side. The algorithms discussed below do without such a strong constraint. In particular, the SIC-based transceiver as described in Item B of the Appendix suffices for the base SICTA algorithm [11]. As follows from this description, in the physical communication systems the information about an event occurring in the slot does not arrive directly to the CAE sublevel of the subscriber. In virtue of the specifics of the transceiving equipment, this information is modified at the physical level. Generally speaking, the volume of the physical level feedback to the CAE sublevel may be retained, expanded, or narrowed.

In the SIC-based algorithms, the volume of feedback accessible at the subscriber CAE sublevel is extended as compared with the triple one because additional information obtained at interference cancelation is used. We modify Assumption 3 about the type of feedback with regard to this fact.

Assumption 3′. *Feedback at interference cancelation.* The procedure of successive interference cancelation operates at the physical level of the subscriber. The feedback volume accessible at the subscriber CAE sublevel is extended with regard to the capabilities of the given procedure. The feedback information is still accessible by the *end* of the current slot.

The base SICTA algorithm [11] requires from the physical level an extended feedback like K–"EMPTY"–"CONFLICT," where K is the number of successfully decoded packets (signals) summed with the left CRT slots labeled as empty after executing the SIC procedure (see for more detail the Appendix, Item A.1). We recall that for appropriate operation of SICTA it is assumed that the receiving side has an unlimited memory to store the conflicting signals received from the communication channel. We extend the system model by Assumption 6 about the volume of signal memory.

Assumption 6. *Volume of the signal memory.* The subscriber physical level has an unlimited number of memory cells to store the signals received from the communication channel.

Let us consider the base SICTA algorithm (Fig. 1c) where the CRP span is only 4 slot. After the successful reception of the signal $y_2 = x_C$ in the second CRP slot, the content of the corresponding left slot is determined as the result of interference cancelation. This operation of interference cancelation was denoted above by $\widetilde{y}_1 = y_1 - x_C$. At the end of the second slot, the feedback $K = 1$

arrives from the physical level to the CAE sublevel. Since slot 3 is empty, according to the MTA rules its corresponding left slot is skipped. Finally, successful reception of the signal x_B in the slot 4 allows one to extract the signal x_A as $x_A = \widetilde{\widetilde{y}}_1 = \widetilde{y}_1 - x_B = y_1 - x_C - x_B$. Since the left subtree of the STA algorithm is skipped entirely by SICTA, its speed is 0.693, that is, is equal exactly to the double speed of STA. See Section 3 for a more rigorous analysis of the SICTA algorithm.

2.4. Memory Limitation at the Receiving Side.

At the receiving side, an unlimited memory area is actually unrealizable. Taking this fact into consideration, [15] proposed for the first time a unit-memory modification of the SICTA algorithm. Since, owing to the simplicity of realization of the SIC procedure, this case is of great practical interest, we modify Assumption 6 about the volume of signal memory.

Assumption 6′. *Unit signal memory.* At the physical subscriber level there is one memory cell to store the signals from the communication channel.

Importantly, it was suggested in [15] to add to each transmitted packet a bit (additional control field) indicating whether it is the first or repeated transmission of the packet. This modification of the system of assumptions makes the value of speed of [15] incomparable with the speed of SICTA. Finally, [16] considered the currently best RMA *decomposition* algorithm featuring *transmission in the order of arrivals* (TOA or First Come First Served (FCFS)) [19] and attaining for the window CAA the speed of 0.4871. Proposed were two SIC-based modifications of this algorithm having within the framework of the model under consideration the speeds 0.6048 and 0.6173, respectively.

The first modification with *success cancelation* (FCFS/SC) enables one to subtract from the stored conflicting signal only the successfully received signal. Therefore, the SIC procedure is made much simpler as compared with that described in the Appendix, Item B. The second modification with *success and collision cancelation* (FCFS/SCC) enables one to subtract from the stored signal both the successful and conflict signals, which accounts for an increase in the algorithm speed at the expense of some complication of the receiver physical level because in practice subtraction of two conflicting signals might prove to be more problematical. Nevertheless, we confine ourselves to the second-type modifications because namely they enable one to realize to the full the potentialities of the SIC procedure.

Unfortunately, the main disadvantage of the algorithms based on the TOA decomposition algorithms lies in the need for extremely precise time service in the communication system, which is extremely difficult to realize. The time resolution should be such that any pair of the arriving packets has different time labels. Otherwise, the TOA decomposition algorithm becomes inoperable [1]. Additionally, all the unit-memory algorithms mentioned in this subsection [15, 16] are unstable to incomplete cancelation of interference to be discussed in what follows.

2.5. Allowance for Incomplete Interference Cancelation

We note that in the physical receivers with the SIC procedure [7] errors of interference cancelation, that is, the residue signals after subtraction of the received signal from the original composite signal, may occur. For example, after the subtraction in some slot t of the signal x_A from the composite signal $x_A + x_B$ the resulting signal contains, $\widetilde{y}_t = x_B + n_A$, where n_A is the residue signal x_A. Similarly, after the subtraction of the signal x_B we obtain $\widetilde{\widetilde{y}}_t = n_A + n_B$.

If the signal n_A+n_B has a sufficient power, then the receiver makes an erroneous decision that the slot is not empty, that is, assumes that an there was a nonexistent conflict between the subscribers. We assume for simplicity that in virtue of the interference cancelation errors this event occurs with a constant probability depending on the characteristics of the receiver. Stated differently, after the operation of interference cancelation the receiver does not decode successfully the signal with the

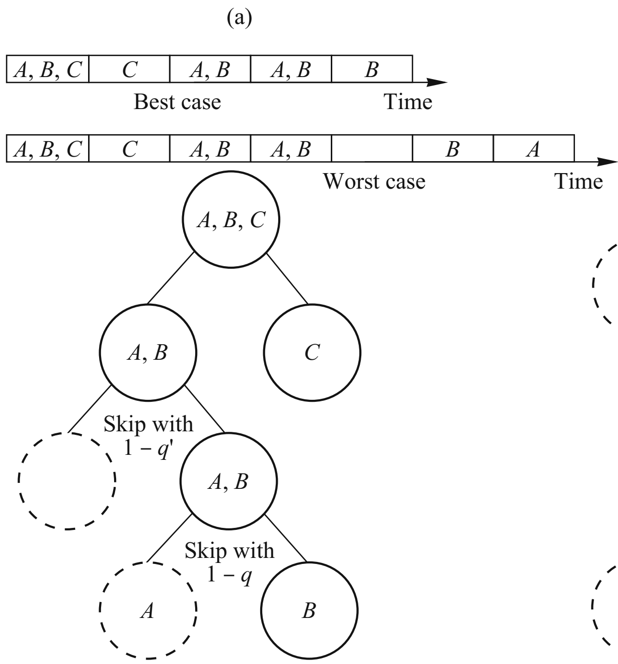

Fig. 3. (a) Example of operation of R-SICTA/SCC; (b)–(e) some subtrees of CRT R-SICTA/SCC.

given probability. To take these errors into consideration, we modify Assumption 4 about feedback confidence.

Assumption 4′. *Imprecise feedback.* In virtue of incomplete interference cancelation, the feedback information arriving to the subscriber CAE sublevel is *imprecise*. This is due to the errors at signal decoding. In particular, after the subtraction of the successfully received signal from the original one we get the error of signal decoding with the probability q. Similarly, subtraction of the conflict signal from the original one provides the error of signal decoding with the probability q'. In practice one may expect that $q' \geq q$, that is, the signal decoding errors are more frequent at subtraction of the conflict signal.

Despite its high speed, the base SICTA algorithm described in the Appendix, Item A.1 is vulnerable to the interference cancelation errors. Indeed, let us assume that in the example of Fig. 1b the last operation of subtraction of the signal x_B from the first saved signal gave some noise level n_B, that is, $\tilde{\tilde{\tilde{y}}}_1 = y_1 - x_C - x_B + n_B$. For a sufficiently high level n_B, the signal x_A cannot be restored successfully, and the process of conflict resolution will go on. Sooner or later when the packet A will be transmitted successfully the residual noise level may be so high that the receiver determines one more conflict in the left slot. According to the rules for operation of the algorithm, the process of conflict resolution between the nonexistent subscribers will continue until it is aborted artificially from outside. Therefore, the *locking* effect arises.

Analysis of the SIC-based algorithms suggests that for the time being there exists no algorithm which is stable to the incomplete interference cancelation, that is, steady within the framework of the classical model of multiple access with infinite number of subscribers (Assumption 5). The algorithm proposed in the present paper (see the Appendix, Item A.2) is stable to the errors of interference cancelation which is reached at the expense of certain reduction in its speed. Additionally, at the receiving side it makes use of the unit memory similar to the approaches of [15, 16]. An increase in the volume of the accessible memory leads to a certain increase in the speed of algorithm which is bounded from above by the speed of SICTA equal to 0,693, but hinders its

Operation of the physical level of R-SICTA/SCC

Rule	Channel–U1	Memory content	U1–U2	Saving
1	“CONFLICT”	$ss - cs = 0$	C/skip	cs
2	“CONFLICT”	$ss - cs = ms$	C/skip	cs
3	“CONFLICT”	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

practical realization. Analysis of the algorithms with more than one memory cell is an independent research problem going out of the scope of the present paper.

The main idea of the algorithm (see Fig. 3a) lies in renunciation to skip some conflicting slots such as slot 3 which might lead to locking if missed. In what follows, this algorithm will be called the *robust SICTA with success and collision cancelation* (R-SICTA/SCC). In Fig. 3a, the time diagram for the best case corresponds to two successful interference cancelations, whereas the diagram for the worst case corresponds to two unsuccessful operations. The algorithm is described in formal terms in the Appendix, Item A.2. We note that in view of the interference cancelation errors, rules 3, 5, and 7 (see the table) should not admit slot missing in CRT which could lead to unstable operation of the algorithm.

3. ANALYSIS OF THE TREE ALGORITHMS WITH INTERFERENCE CANCELATION

3.1. Speed Calculation: General procedure

We describe an approach to calculation of the speed of the tree algorithms with successive interference cancelation by modifying recalculation of the mean time of conflict resolution for the MTA of [17] and then demonstrate its application to analysis of the algorithm derived from the SICTA. We denote by v the duration of CRP in the slots (the conflict resolution time of multiplicity k, the number of vertices in the corresponding CRT) which is a discrete random variable. The conditional expectation $E[v|$ conflict of multiplicity k is permitted] defines the mean CRP length for the conflict with participation of k subscribers. The steps of the proposed procedure for calculation of the speed of the algorithm with the SIC procedure may be formulated as follows:

(1) Let us consider a tree algorithm A with the SIC procedure and denote for it by T_k^A the mean time of conflict resolution of multiplicity k. As follows from [4], the relation $\dfrac{k}{T_k^A}$ enables one to establish the following limits to the speed of the tree algorithm A denoted by R_A:

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

(2) Let us consider the STA whose mean time of conflict resolution is denoted by T_k. By analogy with (1), we set down the limits of the STA speed omitting the subscript and denoting it simply by R:

$$\liminf_{k\to\infty} \frac{k}{T_k} < R < \limsup_{k\to\infty} \frac{k}{T_k}. \qquad (2)$$

We note that the limits of R were calculated in [20] as

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

The relation between the values of limits from (3) and the following expressions

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

where $c = 3.127 \times 10^{-6}$, was determined in [17].

(3) We assume that the number of vertices in the STA CRT is v and obtain $T_k = E[v]$. The number of successful conflict and empty slots during CRP is denoted by v_s, v_c, and v_e, respectively. Since $v_s + v_c + v_e = v$, we use [17] to establish

$$\begin{aligned} v_s &= k, \\ v_c &= \frac{v-1}{2}, \\ v_e &= \frac{v+1}{2} - k. \end{aligned} \tag{5}$$

We consider the CRT of the algorithm A as the STA CRT where the time of looking through some vertices is zero in virtue of the SIC procedure. We calculate the expectation $E[r]$ of the number of such vertices, denote by u the number of vertices with the nonzero look time of CRT of the algorithm A, and put down:

$$E[u] = E[v] - E[r] \tag{6}$$

or

$$T_k^A = T_k - E[r]. \tag{7}$$

By substituting (7) in (1) and transforming it, one can express the limits for the speed R_A in terms of the certain limits for the STA speed (3).

We note that the first two steps of the above procedure are identical for any algorithm A. Therefore, in what follows we confine ourselves only to the third step, that is, determination of $E[r]$ and the final limits of the speed R_A.

3.2. Base SICTA Algorithm

We consider by way of example calculation of the speed of the base algorithm SICTA [11] described in detail in the Appendix, Item A.1. A strict proof may be carried out by following the approach of [17] and describing the SICTA algorithm in the graph-theoretical terms. In what follows, we confine ourselves for brevity to describing the gist of the proof. We note that at operation of the SICTA algorithm for any CRT vertex the content of the first slot of the left subtree is always set up using the SIC procedure without look of the corresponding vertex. To determine the mean number of vertices with a nonzero look time in the SICTA CRT (T_k^S), one needs, therefore, to subtract from the mean number of vertices in the STA CRT (T_k) half of the successful, conflicting (except for the initial CRT slot), and empty slots:

$$T_k^S = T_k - \frac{1}{2}E[v_s] - \frac{1}{2}(E[v_c] - 1) - \frac{1}{2}E[v_e] = T_k - \frac{1}{2}T_k + \frac{1}{2} = \frac{T_k + 1}{2}. \tag{8}$$

Therefore, we obtain with regard to (1) and (4) the following limits for the speed of the SICTA algorithm (R_S):

$$\left(\frac{1}{\ln 2}+c\right)^{-1} < R_S < \left(\frac{1}{\ln 2}-c\right)^{-1}. \tag{9}$$

We conclude analysis by noting that the upper and lower speed limits are very close to each other due to the small value of the constant c. Therefore, the speed of the SICTA algorithm can be conveniently set down as follows:

$$R_S \approx \ln 2 \approx 0.693, \tag{10}$$

which coincides with the well-known result of [11], but this result was established without bulky calculations.

3.3. Proposed R-SICTA/SCC Algorithm

Now we consider the proposed R-SICTA/SCC algorithm which is stable to incomplete interference cancelation. It subtracts from the stored conflict signal both the successful and conflict signals (see Section 2.4) and is stable to the interference cancelation errors (see Section 2.5).

We recall (Assumption 6) that after subtracting the successfully received signal from the original one the resulting signal cannot be decoded successfully with the probability q, and after subtraction of the conflict signal the resulting signal cannot be decoded successfully with the probability q' ($q' \geq q$). According to the procedure for determination of the speed (see Section 3.1), we express T_k^{RS}, the mean number of vertices with nonzero time of look in the R-SICTA/SCC CRT and for that subtract from the mean number T_k of vertices in the STA CRT the following vertices whose look time is zero owing to the SIC procedure:

- Fig. 3b with the probability 1;
- Figs. 3c and 3d with the probability $1-q'$;
- Fig. 3e with the probability $1-q$.

After an elementary rearrangement, with regard to (5) we obtain

$$T_k^{RMS} = \left(\frac{1}{2}+\frac{1}{4}q'\right)T_k - \frac{1}{2} + \frac{1}{4}q' + \frac{k}{2} - \frac{1}{2}(q'-q)N_k, \tag{11}$$

where N_k is the mean number of conflicts of multiplicity two in the STA CRT of the initial multiplicity k. To conclude the analysis, we need to estimate the ratio $\dfrac{N_k}{k}$ for unrestrictedly large k. Obviously, $N_0 = N_1 = 0$ because no "conflicting" vertices can exist in the given CRT's. It is easy to demonstrate that $N_2 = 2$. In the general case, for $k > 2$ with allowance for the properties of the CRT we obtain that

$$N_k = \frac{\sum\limits_{i=1}^{k-1} C_k^i N_i}{2^{k-1}-1}. \tag{12}$$

The recurrent expression (12) can be easily calculated for any finite number k.

We consider the Poissonian transformation [21] of the sequence $N_0, N_1, \ldots, N_i, \ldots$ denoted as follows:

$$N(s) \triangleq \sum_{k\geq 0} N_k \times \frac{s^k}{k!}e^{-s}, \quad s \in \mathcal{R}. \tag{13}$$

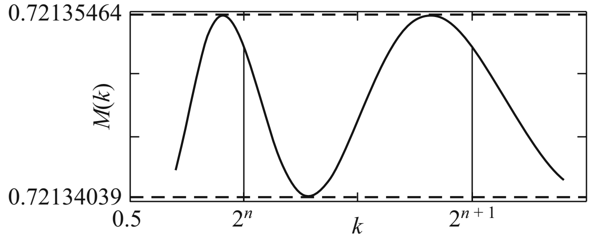

Fig. 4. "Periodic" nature of $M(k)$.

By taking after the approach of [22], it is possible to establish the following recurrent expression for $N(s)$:

$$N(s) = 2N\left(\frac{s}{2}\right) + \frac{s^2}{2}e^{-s}. \tag{14}$$

By analogy with [20], we consider the normalized Poissonian transformation denoted as follows:

$$M(s) \triangleq \frac{N(s)}{s}. \tag{15}$$

Using (14), we can set down (15) as follows:

$$M(s) = M\left(\frac{s}{2}\right) + \frac{s}{2}e^{-s}. \tag{16}$$

For larger values of its argument, the function $M(s)$ is "periodic," which can be used to calculate it. We calculate the normalized Poissonian transformation for sufficiently large values of the argument $2^n r$, where $n \in \mathcal{Z}$ and $r \in \mathcal{R}$. We formally substitute $2^n r$ in (16):

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

For sufficiently large n, variations of the argument of the function $M(2^n r)$ from 2^n to 2^{n+1} correspond to one "period" of $M(2^n r)$. Consequently, for $1 \le r \le 2$ and some n we obtain the largest and smallest values of the function for all subsequent values of its argument. We consider in more detail equality (17) and perform $n-1$ time the recurrent passage:

$$M(2^n r) = M(r) + \sum_{i=1}^{n} \frac{2^i r}{2}e^{-2^i r} = M(r) + H_n(r). \tag{18}$$

The series $H_n(r)$ converges rapidly and can be easily calculated to within the given accuracy. For smaller r, the values of $M(r)$ may be readily calculated with regard to (13) and (15). We now use (18) to examine behavior of the original function $M(2^n r)$ over one "period" where $n \ge 20$. It may be proved that at that the accuracy of the function values is not smaller than 10^{-8}. These values are calculated at the integer points k, that is, $M(2^n r) = M(k)$:

$$\max_{2^{20} \le k \le 2^{21}} M(k) = \limsup_{k\to\infty} M(k) < 0.72135464 + 1 \times 10^{-8} \tag{19}$$

and

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

For sufficiently large k, the values of $M(k)$ over one "period" are shown on the logarithmic scale in Fig. 4.

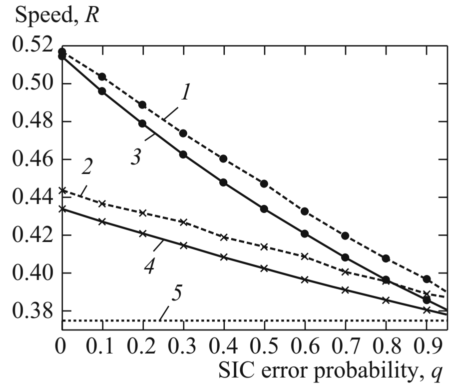

Fig. 5. Speed of R-SICTA/SCC for incomplete interference cancelation: (*1*) R-SICTA/FA, $q' = q$; (*2*) R-SICTA/FA, $q' = 1$; (*3*) R-SICTA/SCC, $q' = q$; (*4*) R-SICTA/SCC, $q' = 1$; (*5*) MTA.

It is possible to prove by following the approach of [23, Theorem 1] that the upper and lower limits of the function $M(k)$ (19) are valid also for $\frac{N_k}{k}$. We note that since the limit of $\frac{N_k}{k}$ does not exist, an interval over which no conclusion can be drawn about behavior of $\frac{N_k}{k}$ arises inevitably. Its length does not exceed 0.00001425, that is, is equal to the difference between the upper and lower limits of the ratio $\frac{N_k}{k}$.

To simplify the representation of the final result, we note that $\limsup\limits_{k\to\infty} \frac{N_k}{k} = \liminf\limits_{k\to\infty} \frac{N_k}{k} = \gamma$ to within at least three decimal positions and $\gamma = 0.721$. For simplicity, we also refuse to determine the upper and lower limits of the speed of R-SICTA/SCC (R_{RS}) assuming that they are sufficiently close. Then, the final approximation for the speed of the proposed algorithm can be established as follows:

$$R_{RS} \approx \frac{4R}{2 + q' + 2R(1 - (q' - q)\gamma)}. \tag{20}$$

In particular, $R_{RS} \approx 0.5147$ if $q' = q = 0$, that is, in the absence of the interference cancelation errors. We focus on two important particular values of the probability q' (see Fig. 5). The first for $q' = q$, that is, subtraction of the successful and conflicting signals is probabilistically indistinguishable. Then, the algorithm speed is $\frac{4R}{2 + q + 2R}$. The second for $q' = 1$, that is, subtraction of the conflicting signals is impossible. Then, we obtain a *stable* SICTA algorithm *with success cancelation* (R-SICTA/SC) having the speed $\frac{4R}{3 + 2R(1 - (1 - q)\gamma)}$.

4. SUMMARY. COMPARISON OF THE SYSTEMS

Combination of the interference cancelation procedure at the physical level and the tree algorithms at the CAE sublevel is a promising line of research enabling one to increase substantially the data transmission rate at unessential complication of the hardware. There exists today a whole family of algorithms based on this approach of which the base SICTA algorithm has the highest rate of steady transmission 0.693 in the model with infinite number of subscribers (see Fig. 6). However, the SICTA algorithm requires an unlimited memory area at the receiving side, which is actually impossible. On the contrary, the aforementioned publications considered modifications of this algorithm for the unit-memory receiver. One can see in Fig. 6 that all steady blocked algo-

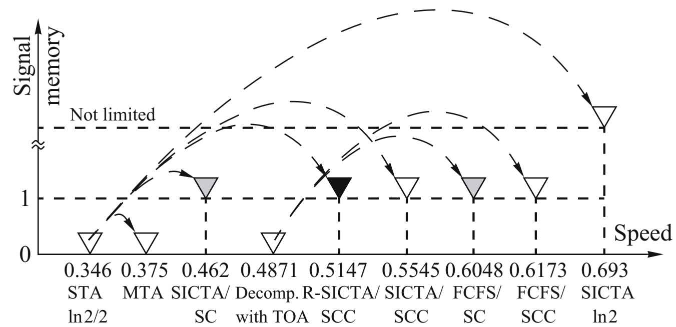

Fig. 6. Comparison of the speed of algorithms with blocked access.

rithms with successive interference cancelation are based either on STA (0.346) or the TOA-based algorithm (0.4871).

In practice, operation of all designs based on the interference cancelation procedure worsens due to incomplete cancelation. In the present paper, the model of the multiple access system was extended with regard to this fact. We recall that the base SICTA algorithm, as well as the modifications of the algorithm of decomposition with TOA having nonzero speed in the model with infinite number of subscribers, are subject to locking in the presence of interference cancelation errors. This gave rise to the need for an algorithm stable to incomplete interference cancelation steady within the framework of the classical model of multiple access with infinite number of subscribers.

The proposed modification of the SICTA algorithm (R-SICTA/SCC) which is stable to the interference cancelation errors has speed 0.5147 in the case of error-free operation of the cancelation procedure (see Fig. 6). Yet the algorithm works satisfactorily even under a high error probability (Fig. 5) and exhibits *graceful* degradation of its characteristics. In the worst case where errors make interference cancelation impossible, it has guaranteed speed 0.375 coinciding with that of the MTA.

Two algorithms that are *unstable* under incomplete interference cancelation [24] can be obtained by modifying the rules of the proposed algorithm (the Appendix, Item A.2). Here, their consideration is of theoretical interest. The first algorithm makes use of the interference cancelation procedure only in the case of successful signal reception. The second algorithm subtracts both the successful and conflicting signals, which speeds it up at the expense of complicated physical level. Using the above approach to analysis of the given algorithms, one can determine the speeds 0.462 and 0.5545, respectively, for the first (SICTA/SC) and second (SICTA/SCC) algorithms (see Fig. 6). Detailed description of the algorithms and determination of their speeds lies outside the scope of the present paper.

It should be noted that a similar idea of using the interference cancelation procedure can be used for direct modification of the decomposition. The corresponding modifications of this algorithm [16] with the unit-memory receiver are known to have speeds 0.6048 (FCFS/SC) and 0.6173 (FCFS/SCC) (see Fig. 6). Unfortunately, practical realization of the decomposition algorithm is hindered by the need for an unlimitedly precise time service. On the contrary, the proposed above algorithm is well balanced in complexity and performance.

5. CONCLUSIONS

We note that the aforementioned blocked R-SICTA/SCC algorithm can be readily modified for the *nonblocked* CAA [25]. Free access does not change the rules described in the Appendix, Item A.2, and just modifies the order of channel occupation by new subscribers which now manifests some advantages. First, the new subscribers can enter the system without initial delay caused by waiting for resolution of the current conflict. Second, the nonblocked algorithm usually are faster.

The speed of the nonblocked modification of the proposed R-SICTA algorithm with free access (R-SICTA/FA) which is stable to incomplete interference cancelation in fact does exceed that of its blocked counterpart (see Fig. 5). Therefore, this modification is recommendable for the physical communication systems. Calculation of the speed of this algorithm is an individual problem also going outside the scope of the given text. Therefore, in Fig. 5 the speed of R-SICTA/FA was obtained by simulation.

APPENDIX

A. DESCRIPTION OF THE TREE ALGORITHMS WITH THE INTERFERENCE CANCELATION PROCEDURE

A.1. Base SICTA Algorithm

The SICTA algorithm needs the following feedback from the physical level to the receiver CAE sublevel:

(1) “CONFLICT’;’

(2) “EMPTY”;

(3) K is the number of successfully decoded packets (signals) plus the number of the left slots in CRT marked as empty after executing the interference cancelation procedure ($K \geq 1$).

The operation rules of the physical level of transceiver are described in Item B.

At the CAE sublevel, each subscriber participating in conflict resolution must follow the variable L_t corresponding to its level in CRT in the slot t. The subscriber transmits its data packet only if $L_t = 0$ by the beginning of the slot. At the start of the algorithm, $L_0 = 0$. At the end of some slot t, the subscriber refreshes the variable L_t with regard to the probability p:

(1) “CONFLICT”: if $L_t > 0$, then $L_{t+1} = L_t + 1$,

otherwise, if $L_t = 0$, then $L_{t+1} = \begin{cases} 0, & p \\ 1, & 1-p. \end{cases}$

(2) “EMPTY”: if $L_t > 1$, then $L_{t+1} = L_t$,

otherwise, if $L_t = 1$, then $L_{t+1} = \begin{cases} 0, & p \\ 1, & 1-p. \end{cases}$

(3) K: $L_{t+1} = L_t - K$.

If $L_t = 0$, then $L_{t+1} = \begin{cases} 0, & p \\ 1, & 1-p, \end{cases}$

otherwise, if $L_t < 0$, then packet received successfully.

A.2. R-SICTA/SCC Algorithm

The R-SICTA/SCC algorithm needs the following feedback from the physical level to the receiver CAE sublevel:

(1) “CONFLICT” and omission of level (content of left slot extracted) (C/skip);

(2) “CONFLICT” and no omission of level (C/–);

(3) “SUCCESS”/“EMPTY” and no omission of level (SE/–);

(4) “SUCCESS” and omission of level (content of left slot extracted) (C/skip);

(5) “EMPTY” and omission of level (inevitable conflict in the next slot) (E/skip).

We denote the received signal, stored in the signal memory, and some successfully decoded signal by cs, ss, and ms, respectively. Operation of the physical level is described in the table. We denote for brevity the physical level by U1 and the CAE sublevel by U2.

Each subscriber participating in conflict resolution at the CAE sublevel should follow the variable L_t corresponding to its level in CRT in the slot t. The subscriber transmits its data packet only if by the beginning of the slot $L_t = 0$. At the beginning of operation of the algorithm, $L_0 = 0$. At the end of some slot t, the subscriber refreshes the value of the variable L_t with regard to the probability p as follows:

(1) C/skip: if $L_t \geq 2$, then $L_{t+1} = L_t$,
otherwise, if $L_t = 1$, then packet received successfully,
otherwise, $L_{t+1} = \begin{cases} 0, & p \\ 1, & 1-p. \end{cases}$

(2) C/–: if $L_t > 0$, then $L_{t+1} = L_t + 1$,
otherwise, $L_{t+1} = \begin{cases} 0, & p \\ 1, & 1-p. \end{cases}$

(3) SE/–: if $L_t > 0$, then $L_{t+1} = L_t - 1$,
otherwise, packet received successfully.

(4) S/skip: if $L_t \geq 2$, then $L_{t+1} = L_t - 2$,
otherwise, packet received successfully.

(5) E/skip: if $L_t \geq 2$, then $L_{t+1} = L_t$,
otherwise, if $L_t = 1$, then $L_{t+1} = \begin{cases} 0, & p \\ 1, & 1-p. \end{cases}$

B. DESCRIPTION OF THE INTERFERENCE CANCELATION PROCEDURE

A simplified diagram of the transceiver with interference cancelation which can be used to realize the base SICTA algorithm [11] described in the Appendix, Item A.1 is shown in Fig. 7. The bold arrows show the directions data transmission, and the dashed arrows, the directions of the control information.

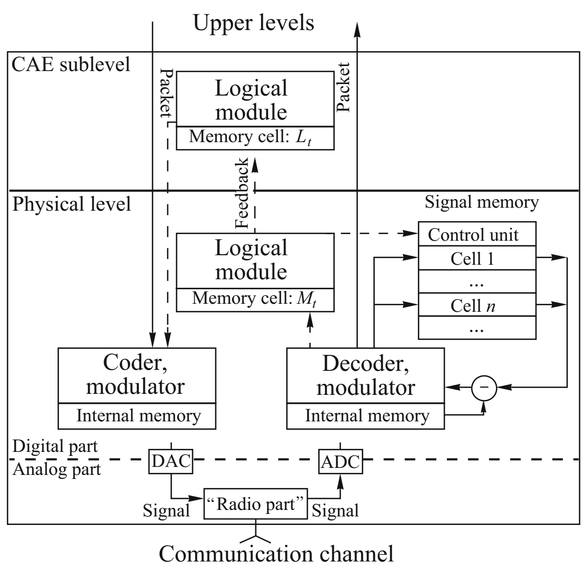

Fig. 7. Simplified diagram of the transceiver with interference cancelation.

Let us consider operation of the transceiver shown in Fig. 7 at the end of a current slot. The transmitting part is implemented as a downward path of the data arriving as packets from the upper levels of the communication system. Upon passing the *coder* and *modulator*, the digital data are converted into the analog form and sent to the communication channel. At that, the *"radio part"* performs conversion of the sequence of readings in a radio signal and back. We note that the *logic module of the CAE sublevel* realizing some tree algorithm controls the coder by indicating the time instants when it must start further conversion of data for transmission. For that, the logic module follows the variable L_t having the sense of the current position of the subscriber in CRT in the slot t (see Section 2.2 and Appendix, Item A). The subscriber transmits its data packet only if $L_t = 0$.

Upon conversion of the received analog signal in the digital form, operation of the receiving part is as follows. An attempt is made to demodulate and decode the received signal which is temporarily stored for that in the internal (register) memory of the *decoder*. In the case of failure (a conflict signal received), the decoder informs the *logic module of the physical level* and keeps the digitized signal readings in the *signal memory*. At that, the logic module controls the process of writing by acting upon the *control unit* and indicating the current free cell. For that, the given module must follow the number of stored conflict signals M_t in the slot t.

In the case of successful decoding of the (successfully) received signal, the decoder also informs the logic module and tries to decode in turn the difference between the received signal and each of the stored signals. The results are communicated to the logic module which as before controls memory access, and the "new" conflicting signals obtained by subtraction replace the "older" ones.

At operation of the described interference cancelation procedure, each successfully decoded signal gives rise to a new iteration of successive decoding. At that, the differences of the given signal and the rest of the stored signals are fed into the decoder input. In the course of such iterative decoding, all successfully restored data packets are transmitted upward along the ascending datapath, and the corresponding stored signals are removed from the memory. The logical module of the physical level processes the results of each attempt of decoding and upon completion transmits the feedback of a predefined type to the logical of the CAE sublevel.

REFERENCES

1. Bertsekas, D. and Gallager, R., *Data Networks*, Englewood Cliffs: Prentice Hall, 1987. Translated under the title *Seti peredachi dannykh*, Moscow: Mir, 1989.

2. Rom, R. and Sidi, M., *Multiple Access Protocols: Performance and Analysis*, New York: Springer, 1990.

3. Abramson, N., The Aloha System—Another Alternative for Computer Communications, *Proc. AFIPS Conf.*, 1970, vol. 36, pp. 295–298.

4. Tsybakov, B.S. and Mikhailov, V.A., Free Synchronous Access of Packets to a Broadcast Feedabck Channel, *Probl. Peredachi Inf.*, 1978, vol. 14, no. 4, pp. 32–59.

5. Capetanakis, J.I., Tree Algorithms for Packet Broadcast Channels, *IEEE Trans. Inform. Theory*, 1979, vol. 25, no. 4, pp. 505–515.

6. Pedersen, K.I., Kolding, T.E., Seskar, I., and Holtzman, J.M., Practical Implementation of Successive Interference Cancellation in DS/CDMA Systems, *Proc. Int. Conf. Universal Personal Commun.*, 1996, pp. 321–325.

7. Andrews, J., Analysis of Cancellation Error for Successive Interference Cancellation with Imperfect Channel Estimation, in *Multiuser Wireless Commun.*, 2002, pp. 1–17.

8. Agrawal, A., Andrews, J.G., Cioffi, J.M., and Meng, T., Iterative Power Control for Imperfect Successive Interference Cancellation, *IEEE Trans. Wireless Commun.*, 2005, vol. 4, pp. 878–884.

9. Weber, S., Andrews, J.G., Yang, X., and Veciana, G.D., Transmission Capacity of Wireless ad hoc Networks with Successive Interference Cancellation, *IEEE Trans. Inform. Theory*, 2007, vol. 53, pp. 2799–2814.

10. *IEEE Std 802.16e-2005*, Piscataway, New Jersey, USA, December 2005.

11. Yu, Y., SICTA: A 0.693 Contention Tree Algorithm Using Successive Interference Cancellation, *Proc. IEEE Conf. Comput. Commun.*, 2005, vol. 3, pp. 1908–1916.

12. Yu, Y. and Giannakis, G.B., High-throughput Random Access Using Successive Interference Cancellation in a Tree Algorithm, *IEEE Trans. Inform. Theory*, 2007, vol. 53, no. 12, pp. 4628–4639.

13. Wang, X., Yu, Y., and Giannakis, G.B., Combining Random Backoff with a Cross-layer Tree Algorithm for Random Access in IEEE 802.16, *Proc. IEEE Wireless Commun. Networking Conf.*, 2006, vol. 2, pp. 972–977.

14. Wang, X., Yu, Y., and Giannakis, G.B., A Deadlock-free High-throughput Tree Algorithm for Random Access over Fading Channels, *Proc. Conf. Inform. Sci. Syst.*, 2006, vol. 22, pp. 420–425.

15. Peeters, G.T., Houdt, B.V., and Blondia, C., A Multiaccess Tree Algorithm with Free Access, Interference Cancellation and Single Signal Memory Requirements, *Performance Evaluat.*, 2007, vol. 64, no. 9–12, pp. 1041–1052.

16. Houdt, B.V. and Peeters, G.T., FCFS Tree Algorithms with Interference Cancellation and Single Signal Memory Requirements, *Proc. Int. Workshop on Multiple Access Commun.*, 2008, vol. 1, pp. 1–6.

17. Evseev, G.S. and Tyurlikov, A.M., Interrelation of the Characteristics of the Blocked Stack Algorithms of Random Multiple Access, *Probl. Peredachi Inf.*, 2007, vol. 43, no. 4, pp. 83–92.

18. Massey, J.L., Collision Resolution Algorithm and Random Access Communications, in *Multiuser Commun. Syst. G., Longo: CISM Course and Lecture Notes*, 1981, pp. 73–131.

19. Tsybakov, B.S. and Mikhailov, V.A., Random Multiple Packet Access. Fracturing Algorithm, *Probl. Peredachi Inf.*, 1980, vol. 16, no. 4, pp. 65–79.

20. Gyorfi, L., Gyori, S., and Massey, J.L., Principles of Stability Analysis for Random Accessing with Feedback, *Proc. NATO Security Through Sci., Ser. Inform. Commun. Security*, 2007, vol. 10, pp. 214–250.

21. Szpankowski, W., *Average Case Analysis of Algorithms on Sequences*, New York: Wiley, 2001.

22. Mikhailov, V.A., On a Recurrent Equation in the Theory of Random Multiple Access, *IX Symp. Inf. Redundancy in Inform. Systems*, 1986, vol. 2, pp. 148–150.

23. Gyorfi, L. and Gyori, S., Analysis of Tree Algorithm for Collision Resolution, *Proc. Int. Conf. Anal. Algorithms*, 2005, pp. 357–364.

24. Andreev, S., Pustovalov, E., and Turlikov, A., SICTA Modifications with Single Memory Location and Resistant to Cancellation Errors, *Proc. Int. Conf. Next Generation Teletraffic and Wired/Wireless Advanced Networking*, 2008, pp. 13–24.

25. Andreev, S., Pustovalov, E., and Turlikov, A., Tree Algorithms with Free Access and Interference Cancellation in Presence of Cancellation Errors, *Proc. Int. Symp. Wireless Personal Multimedia Commun.*, 2008, vol. 1, pp. 1–5.

This paper was recommended for publication by V.M. Vishnevskii, a member of the Editorial Board