RANDOM ACCESS ALGORITHMS WITH MULTIPLE RECEPTION CAPABILITY AND n-ARY FEEDBACK CHANNEL

N. Likhanov, E. Plotnik, Y. Shavitt, M. Sidi, and B. Tsybakov UDC 621.354.74:519.2

Random access algorithms to a common channel with multiple reception capability by receivers and an n-ary feedback channel are presented. The algorithms belong to the class of splitting algorithms. It is shown that the throughput of these algorithms is 1.5-3% higher than the throughput of the best known algorithm with ternary feedback channel.

1. Introduction and Channel Model

The problem of sharing a common channel by a number of users was extensively addressed in the literature (see [1,2] and references therein). Various access algorithms to the channel were presented and their performance was analyzed. Most of them assumed that when two or more users transmit simultaneously, a collision occurs, and retransmission of colliding packets is needed.

In [3] algorithms with multiple reception capability were presented. They utilize the fact that there are situations where simultaneous transmissions of at most $n-1$ packets is possible by the use of coding to resolve collisions at the receiver. The algorithms in [3] assumed a ternary feedback channel where at the end of each transmission slot in the direct channel all the users are notified whether it was a successful transmission, a collision or an idle slot.

In this paper, we assume an n-ary feedback channel which, in the case of a successful slot, indicate the exact number of succeeding packets. It is shown that algorithms that utilize this fact can gain 1.5-3% in throughput when compared to [3].

Consider the following model for multiaccess communication: An infinite number of independent users are transmitting packets of equal length T over a slotted-time broadcast channel with length of slots equals to T. The number of new messages generated collectively by all users in each slot is a Poisson flow with intensity λ packets per slot. The numbers of new packets generated in different slots are independent.

In the event of less than n packets being transmitted simultaneously in the same slot, the receiver decodes all of them successfully and all the users are informed via a feedback channel on the number of packets that were transmitted. If n or more packets are simultaneously transmitted in the slot, a collision occurs and all colliding packets must be retransmitted. In the case of a collision, the assumption is that the receiver cannot decode even the number of colliding packets, and therefore the feedback just indicates that a collision occurred.

2. Description of the Algorithm

1. Consider two time axes. The first, called the arrival axis, shows the Poisson arrival instants of the packets. The second, called the transmission axis, is segmented into consecutive intervals called *collision resolution intervals* (CRI). In the first slot of the ith CRI, the algorithm enables transmission of all packets that arrive in a fix-length interval in the arrival axis. If no collision occurs in this slot, the ith CRI terminates and a new CRI starts immediately. If a collision occurs, the algorithm initiates a collision resolution process (described below) whose termination defines the termination of the ith CRI.

2. The Collision Resolution Process (CRP). Let a be the collision interval in the arrival axis to be resolved (see Fig. 1). Since a is a collision interval, it is known to contain at least n packets. First, the algorithm enables the packets that arrived at the left part of a of length αa. If a collision occurs, the

Translsted from *Problemy Peredachi Informatsii*, Vol. 29, No. 1, pp. 82–91, January-March 1993. Original article submitted April 10, 1992.

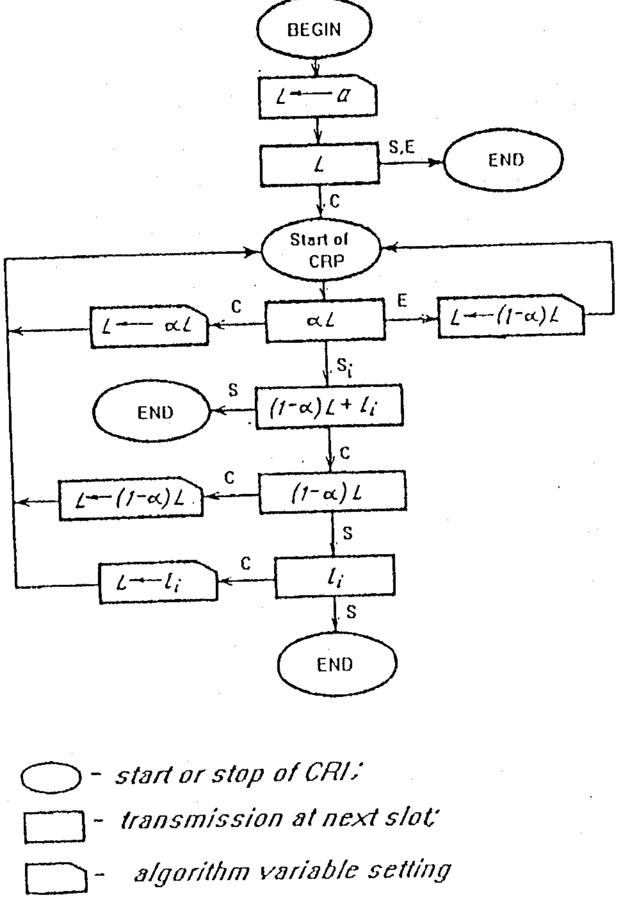

Fig. 1. Diagram of the algorithm. Here, S_i is a simultaneous transmission of i packets, and L is an internal variable of the algorithm.

CRP starts again for the interval αa (i.e., enables all packets in an interval of length $\alpha^2 a$), and the right part of a is disabled and is returned to the arrival axis. If no packet is transmitted, the CRP starts for the interval $(1-\alpha)a$ (i.e., enables all packets in an interval of length $\alpha(1-\alpha)a$), and the left part is considered as resolved. If $i < n$ packets succeed, then the algorithm enables all packets in the second part of the interval. But since in this step the algorithm is re-initiated for an interval that (statistically) contains less packets than the optimal value, an interval l_i from the arrival axis, whose length depends on the value of i, is added to the enabled interval. When all packets succeed, the CRP terminates and a new CRI starts. If a collision occurs in the enabled interval of length $(1-\alpha)a + l_i$, then only the packets in the $(1-\alpha)a$ interval are enabled. A collision of this interval starts a CRP of basic length $(1-\alpha)a$ and a success leads to the transmission of all the packets that were previously in the interval of length l_i. A success here terminates the CRI, while a collision starts a new CRP for the interval of length l_i. The algorithm is graphically summarized in Fig. 1.

3. Analysis of the Algorithm. Let $a \triangleq \lambda T$ be the interval length. Denote by $T^*(a)$ the average number of slots needed to complete a CRI of length a, and let $N^*(a)$ be the average number of packets successfully transmitted during this CRI. Let $T(a)$ be the average number of slots in a CRP for an interval of length a, and $N(a)$ the average number of packets successfully transmitted in that CRP. Also, assume that W_a is the random number of arrivals in an interval of length a.

Clearly,

$$T^*(a) = 1 + T(a) - \sum_{i=0}^{n-1} \Pr\{W_a = i\} T(a), \tag{2.1}$$

$$N^*(a) = N(a) - \sum_{i=0}^{n-1} \Pr\{W_a = i\} (N(a) - i). \tag{2.2}$$

We derive functional equations for $T(a)$ and $N(a)$ based on the CRI depicted in Fig. 1.

$$\begin{aligned} T(a) = 1 &+ \Pr\{W_{\alpha a} \geq n | W_a \geq n\} \cdot T(\alpha n) \\ &+ \Pr\{W_{\alpha a} = 0 | W_a \geq n\} \cdot T((1-\alpha)a) \\ &+ \sum_{i=1}^{n-1} \Pr\{W_{\alpha a} = i | W_a \geq n\} \cdot \\ &\quad [\ \Pr\{(W_{(1-\alpha)a} + W_{l_i}) \leq n-1 | W_a \geq n, \quad W_{\alpha a} = i\} \\ &\quad + \Pr\{W_{(1-\alpha)a} \geq n | W_a \geq n,\ W_{\alpha a} = i\} \cdot (2 + T((1-\alpha)a)) \\ &\quad + \Pr\{W_{(1-\alpha)a} < n, W_{l_i} < n, \left(W_{(1-\alpha)a} + W_{l_i}\right) \geq n | W_a \geq n, W_{\alpha a} = i\} \cdot 3 \\ &\quad + \Pr\{W_{(1-\alpha)a} < n, W_{l_i} \geq n, | W_a \geq n, W_{\alpha a} = i\} \cdot (3 + T(l_i))], \end{aligned} \tag{2.3}$$

$$\begin{aligned} N(a) &= \Pr\{W_{\alpha a} \geq n | W_a \geq n\} \cdot N(\alpha a) \\ &+ \Pr\{W_{\alpha a} = 0 | W_a \geq n\} \cdot N((1-\alpha)a) \\ &+ \sum_{i=1}^{n-1} \Pr\{W_{\alpha a} = i | W_a \geq n\} \cdot \\ &\quad [i + \sum_{j=n-i}^{n-1} \Pr\left\{(W_{(1-\alpha)a} + W_{l_i}) = j | W_a \geq n, \quad W_{\alpha a} = i\right\} \cdot j \\ &\quad + \Pr\left\{W_{(1-\alpha)a} \geq n | W_a \geq n,\ W_{\alpha a} = i\right\} \cdot N((1-\alpha)a) \\ &\quad + \sum_{j=n}^{2(n-1)} \Pr\left\{(W_{(1-\alpha)a} + W_{l_i}) = j,\ W_{(1-\alpha)a} < n,\ W_{l_i} < n | W_a \geq n,\ W_{\alpha a} = i\right\} \cdot j \\ &\quad + \sum_{j=n-i}^{n-1} \Pr\left\{W_{(1-\alpha)a} = j,\ W_{l_i} \geq n | W_a \geq n,\ W_{\alpha a} = i\right\} (j + N(l_i))]. \end{aligned} \tag{2.4}$$

Conditional probabilities Pr{.} from (2.3) and (2.4) are easy derived from the Poisson distribution. For example,

$$\Pr\{W_{\alpha a} \geq n | W_a \geq n\} = F(n, \alpha a)/F(n, a), \tag{2.5}$$

where

$$F(i, a) \triangleq 1 - \sum_{j=0}^{i-1} p(j, a), \tag{2.6}$$

$$p(j,a) = e^{-a}a^j/j. \tag{2.7}$$

The throughput R of the algorithm is defined as

$$R = N^*(a)/T^*(a). \tag{2.8}$$

The functional equations (2.3)-(2.4) are solved by discretization of the interval a into equal length segments τ. This yields a set of linear equations with a finite number of variables $T(i\tau)$ and $N(i\tau)$ ($0 \leq i \leq I$, I taken big enough), where $T(a)$ ($N(a)$) is replaced with the closest $T(i\tau)$ ($N(i\tau)$). By making the discretization finer (i.e., taking smaller τ), the discretization error can be reduced below any desired value. The optimization procedure for a, α, and l_i yields the optimal values shown in Table 1.

Table 1.

n	Throughput	a	α	l_1	l_2	l_3	l_4	l_5	l_6	l_7	l_8	l_9	l_{10}
2	0.4743	1.19	0.500	0.12									
3	0.9758	1.81	0.544	0.0	.625								
4	1.5237	2.49	0.572	0.0	.259	1.04							
5	2.1103	3.19	0.587	0.0	.032	0.85	1.47						
6	2.7224	3.90	0.603	0.0	0.0	0.59	1.38	1.925					
7	3.3542	4.62	0.613	0.0	0.0	0.43	1.18	1.90	2.39				
8	4.0027	5.35	0.627	0.0	0.0	0.33	1.03	1.78	2.44	2.88			
9	4.6656	6.08	0.640	0.0	0.0	0.27	0.92	1.65	2.38	2.99	3.39		
10	5.3414	6.82	0.653	0.0	0.0	0.22	0.84	1.54	2.28	2.98	3.55	3.92	
11	6.0290	7.57	0.666	0.0	0.0	0.19	0.78	1.46	2.19	2.92	3.60	4.145	4.47

3. Improved Algorithms

1. From Table 1 it is clear that there are situations where the algorithm presented in Section 2 is not efficient and can be easily improved.

For example, consider the case where $n = 4$ and in the first slot of the CRI a collision occurs. According to the algorithm the packets arriving in interval αa are enabled for transmission in the second slot, and assume that one ($i = 1$) packet is transmitted. Next (in the third slot), the packets in $(1-\alpha)a + l_1$ are enabled, and a collision is assumed to occur again. Following the algorithm, the packets in the interval $(1-\alpha)a$ should be transmitted in the next (fourth) slot. Since $l_1 = 0$, a collision will definitely occur and therefore the next slot is a wasted one.

As a second example of the inefficiency of the algorithm, suppose that in the second slot of the above-mentioned case of CRI, two packets (instead of one considered earlier) succeed, and consider the following situation where a collision occurs in the next (third) slot, when the packets in the interval of length $(1-\alpha)a + l_2$ were enabled for transmission. Since l_2 is almost zero, there is a high probability that by omitting it, a collision still occurs in the next slot when the packets in the interval of length $(1-\alpha)a$ are enabled for transmission.

These examples show that it is possible to propose the following improvement to the algorithm.

Let $b(n)$ denote the maximal number of succeeding packets in interval αa, for which it is not worthwhile to add a "fresh water" interval of length $l_{b(n)}$ to the one of length $(1-\alpha)a$. In the event that $b(n)$ or less packets succeed, no "fresh water" will be added and only packets from the interval of length $(1-\alpha)a$ will be enabled. If a success occurs the CRI terminates, and if a collision occurs a new CRP starts with basic length $(1-\alpha)a$.

When the number of succeeding packets in the first interval of the CRP is more then $b(n)$, the algorithm proceeds as that from Section 2. The algorithm with this improvement is called simplified improved algorithm.

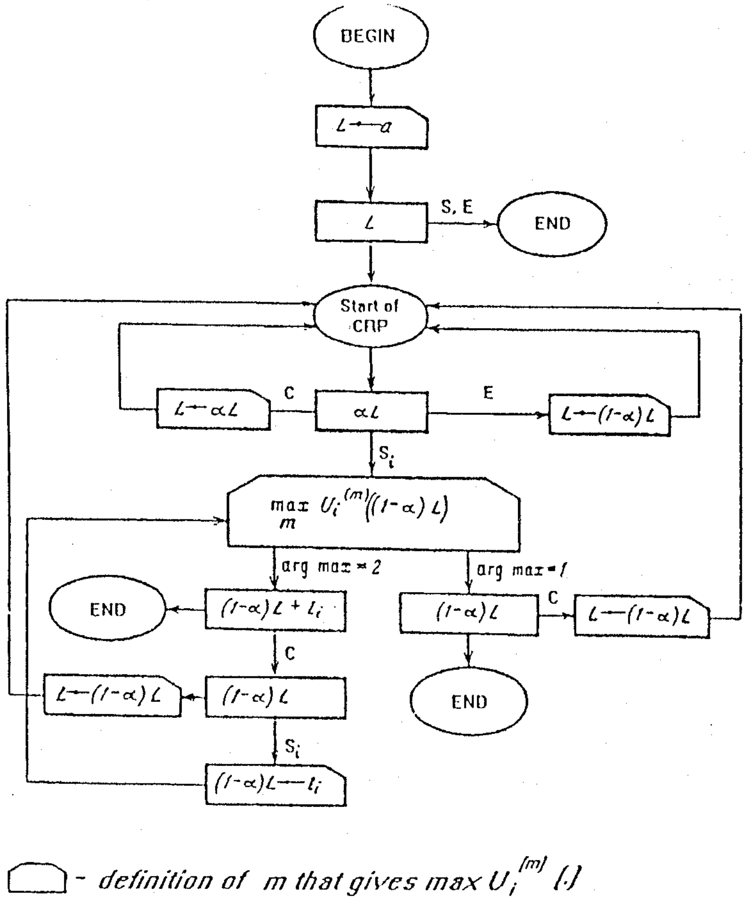

Fig. 2. Diagram of the improved algorithm. Here, argmax=1 means that the inequality $U_i^{(1)}((1-\alpha)L) \geq U_i^{(2)}((1-\alpha)L)$ holds and argmax=2 means that the reverse inequality is true. Other blocks are the same as in Fiq. 1.

A further improvement to the algorithm can be obtained if the functions $U_i^{(1)}(a)$ and $U_i^{(2)}(a)$ of i and a are introduced instead of the function $b(n)$. The algorithm enables the interval $(1-\alpha)L$ if $U_i^{(1)}((1-\alpha)L) \geq U_i^{(2)}((1-\alpha)L)$, and the interval $(1-\alpha)L$ with additional "fresh water" interval otherwise. The functions should be chosen so that the throughput of the algorithm is maximized.

Instead of using the two functions $U_i^{(m)}(a)$ $(m = 1, 2)$, we can equivalently work with one indicator function $g(i, a)$ that equals to 1 when $U^{(1)} \geq U^{(2)}$ and 0 otherwise. However, we decide to introduce the two functions $U_i^{(m)}(a)$ because they are used in the equations below (see (3.10) - (3.13)).

To see intuitively that using the two functions $U_i^{(1)}(a)$ and $U_i^{(2)}(a)$ can indeed increase the algorithm throughput, recall (see [4]) that in the part-and-try algorithm, addition of "fresh water" yields a throughput increase only if the length of the interval (to which "fresh water" is added) is small enough. But in the part-and-try algorithm, a fixed "fresh water" interval was added, regardless of the length of the splitt interval, thus causing a decrease in the throughput.

Finally, another improvement that we make is to use the function $\alpha(a)$ instead of the constant α and to use the functions $l_i(a)$ instead of l_i.

The algorithm with these improvements is called improved algorithm.

The improved algorithm is described in Fig. 2.

2. Now we give the equations for $T(a)$ and $N(a)$ for the improved algorithm, replacing the conditional probabilities for W_i by their expressions in terms of the Poisson functions (2.6) and (2.7). We start with

the equation for $T(a)$,

$$T(a) = 1 + T(\alpha a)F(n,\alpha a)/F(n,a) + T((1-\alpha)a)\,p(0,\alpha a)F(n,(1-\alpha)a)/F(n,a) + \sum_{i=1}^{n-1} T_i((1-\alpha)a)\,p(i,\alpha a)F(n-i,(1-\alpha)a)/F(n,a), \quad (3.1)$$

where $T_i(a)$ as a function of a and i is given by

$$T_i(a) = 1 + T(a)F(n,a)/F(n-i,a) \quad (3.2)$$

when "fresh water" is added, and

$$T_i(a) = 1 + (1+T(a))\,F(n,a)/F(n-i,a) + \sum_{j=n-i}^{n-1} (1+T_j(l_j))\,p(j,a)F(n-j,l_i)/F(n-i,a) \quad (3.3)$$

otherwise.

The equation for $N(a)$ takes the following form:

$$N(a) = N(\alpha a)F(n,\alpha a)/F(n,a) + N((1-\alpha)a)\,p(0,\alpha a)F(n,(1-\alpha)a)/F(n,a) + \sum_{i=1}^{n-1} (i + N_i((1-\alpha)a))\,p(i,\alpha a)F(n-i,(1-\alpha)a)/F(n,a) \quad , \quad (3.4)$$

where $N_i(a)$ as a function of a and i is given by

$$N_i(a) = N(a)F(n,a)/F(n-i,a) + \sum_{j=n-i}^{n-1} jp(j,a)/F(n-i,a) \quad (3.5)$$

when "fresh water" is added, and

$$N_i(a) = N(a)F(n,a)/F(n-i,a) + \sum_{j=n-i}^{n-1} \sum_{k=n-i+j}^{n-1} (j+k)p(j,a)p(k,l_i)/F(n-j,a) + \sum_{j=n-i}^{n-1} (j + N_i(l_i))\,p(j,a)F(n-j,l_i)/F((n-i),a) \quad (3.6)$$

otherwise.

The throughput of the improved algorithm is given as before by (2.8), where $T^*(a)$ and $N^*(a)$ are given by (2.1) and (2.2).

Maximization of the throughput over parameters $a, \alpha(a)$ and $l_i(a)$ is carried out in the following way. We define the function $V^*(a)$ by

$$V^*(a) \triangleq (N^*(a) - RT^*(a))\,F(n,a). \quad (3.7)$$

As in [5], it is easy to see that the maximum throughput R is the solution to the equation

$$\max_a V^*(a) = 0, \quad (3.8)$$

and $V^*(a)$ can be expressed by the equation

$$V^*(a) = \sum_{i=0}^{n-1} ip(i,a) - R + V(a), \tag{3.9}$$

where $V(a)$ is a solution to the functional equation

$$\begin{aligned} V(a) &= \max_{\alpha(a)} [V(\alpha a)F(n,\alpha a) + V((1-\alpha)a)\, p(0,\alpha a) F(n,(1-\alpha)a) \\ &+ \sum_{i=1}^{n-1} U_i((1-\alpha)a)\, p(i,\alpha a) \\ &+ \sum_{i=1}^{n-1} ip(i,\alpha a) F(n-i,(1-\alpha)a) - RF(n,a)], \end{aligned} \tag{3.10}$$

where

$$U_i(a) = \max\left(U_i^{(1)}(a),\, U_i^{(2)}(a)\right), \tag{3.11}$$

$$U_i^{(1)}(a) = V(a) + \sum_{j=n-i}^{n-1} jp(j,a) - RF(n-i,a), \tag{3.12}$$

$$\begin{aligned} U_i^{(2)}(a) &= \max_{l_i(a)} \left[V(a) + \sum_{j=n-i}^{n-1} U_j(l_j) p(j,a) \right. \\ &+ \sum_{j=n-i}^{n-1} (j-R) p(j,a) F(n-j,l_i) \\ &\left. + \sum_{j=n-i}^{n-1} \sum_{m=n-i+j}^{n-1} (j+m) p(j,a) p(m,l_i) - RF(n-i,a) \right]. \end{aligned} \tag{3.13}$$

The parameters $a, \alpha(a)$ and $l_i(a)$ that give maximum R are the same as those that yield maximum in (3.8), (3.10), and (3.13) respectively.

Equation (3.10) can be solved by iterations as in [5].

3. The simplified improved algorithm is described by Fig. 3.

The equations for $T(a)$ and $N(a)$ take the following form:

$$\begin{aligned} T(a) = 1 &+ \Pr\{W_{\alpha a} \geq n | W_a \geq n\} \cdot T(\alpha a) \\ &+ \Pr\{W_{\alpha a} = 0 | W_a \geq n\} \cdot T((1-\alpha)a) \\ &+ \sum_{i=1}^{b(n)} \Pr\{W_{\alpha a} = i | W_a \geq n\} \left[1 + \Pr\{W_{(1-\alpha)a} \geq n | W_{\alpha a} = i, W_a \geq n\} T((1-\alpha)a)\right] \end{aligned}$$

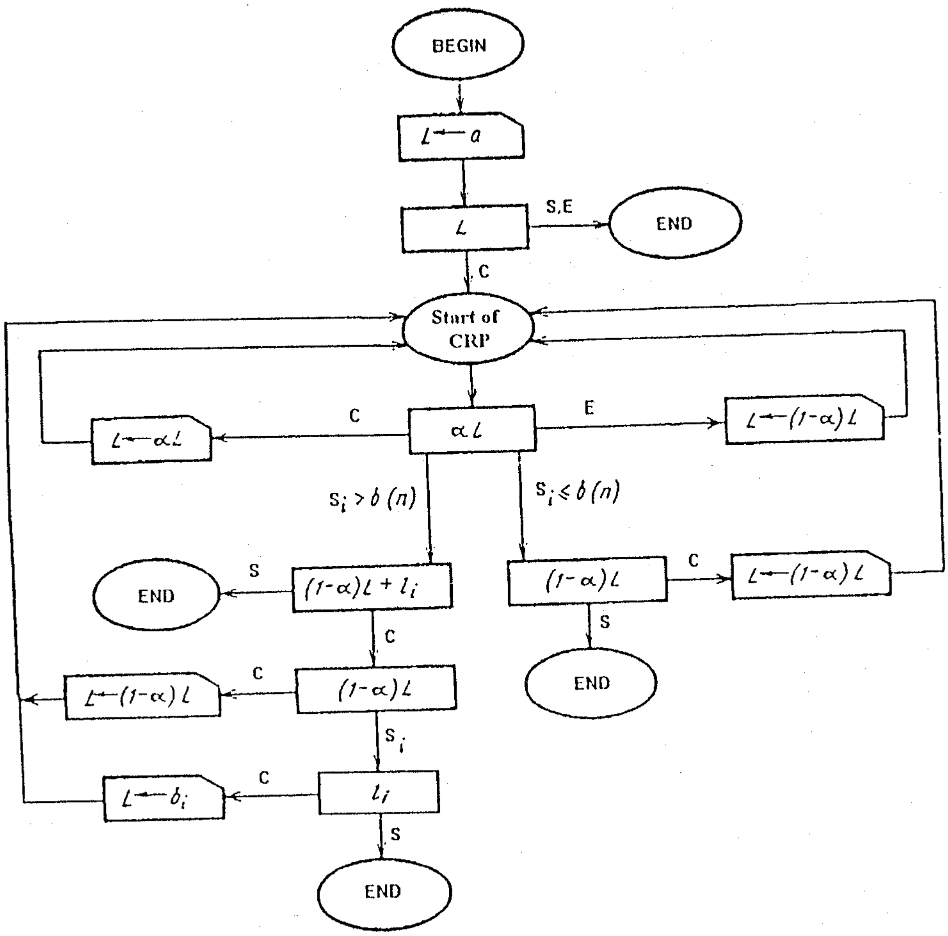

Fig. 3. Diagram of the simplified improved algorithm. Here, $S_i > b(n)$ means that the number of successfully transmitted packets in a slot is greater than $b(n)$. Inequality $S_i \leq b(n)$ has the same meaning. Other blocks are the same as in Fiq. 1.

$$
\begin{aligned}
&+ \sum_{i=b(n)+1}^{n-1} \Pr\{W_{\alpha a} = i | W_a \geq n\} \cdot \\
&\quad [\ \Pr\{(W_{(1-\alpha)a} + W_{l_i}) \leq n-1 | W_a \geq n, \quad W_{\alpha a} = i\} \\
&\quad + \Pr\{W_{(1-\alpha)a} \geq n | W_a \geq n,\ W_{\alpha a} = i\} \cdot (2 + T((1-\alpha)a)) \\
&\quad + \Pr\{W_{(1-\alpha)a} < n, W_{l_i} < n, (W_{(1-\alpha)a} + W_{l_i}) \geq n | W_a \geq n, W_{\alpha a} = i\} \cdot 3 \\
&\quad + \Pr\{W_{(1-\alpha)a} < n, W_{l_i} \geq n, | W_a \geq n, W_{\alpha a} = i\} \cdot (3 + T(l_i))],
\end{aligned} \tag{3.14}
$$

$$
\begin{aligned}
N(a) &= \Pr\{W_{\alpha a} \geq n | W_a \geq n\} \cdot N(\alpha a) \\
&+ \Pr\{W_{\alpha a} = 0 | W_a \geq n\} \cdot N((1-\alpha)a) \\
&+ \sum_{i=1}^{b(n)} \Pr\{W_{\alpha a} = i | W_a \geq n\}[i + \sum_{j=n-i}^{n-1} \Pr\{W_{(1-\alpha)a} = j | W_a \geq n, W_{\alpha n} = i\} \cdot j \\
&\qquad + \Pr\{W_{(1-\alpha)a} \geq n | W_a \geq n, W_{\alpha a} = i\} \cdot N((1-\alpha)a)] \\
&+ \sum_{i=b(n)+1}^{n-1} \Pr\{W_{\alpha a} = i | W_a \geq n\} \cdot \\
&\quad [i + \sum_{j=n-i}^{n-1} \Pr\{(W_{(1-\alpha)a} + W_{l_i}) = j | W_a \geq n, \quad W_{\alpha a} = i\} \cdot j \\
&\quad + \Pr\{W_{(1-\alpha)a} \geq n | W_a \geq n,\ W_{\alpha a} = i\} \cdot N((1-\alpha)a) \\
&\quad + \sum_{j=n}^{2(n-1)} \Pr\{(W_{(1-\alpha)a} + W_{l_i}) = j,\ W_{(1-\alpha)a} < n,\ W_{l_i} < n | W_a \geq n,\ W_{\alpha a} = i\} \cdot j \\
&\quad + \sum_{j=n-i}^{(n-1)} \Pr\{W_{(1-\alpha)a} = j,\ W_{l_i} \geq n | W_a \geq n,\ W_{\alpha a} = i\}(j + N(l_i))].
\end{aligned} \tag{3.15}
$$

The equations for $T^*(a)$ and $N^*(a)$ given by (2.1) and (2.2) and the throughput R given by (2.8) remain as before. Optimization over a, α, l_i, and $b(n)$ yields the results in Table 2, showing improvements of up to 1.5% in the throughput.

Table 2.

n	Throughput	a	α	$b(n)$	l_2	l_3	l_4	l_5	l_6	l_7	l_8	l_9	l_{10}
2	0.4867	1.27	0.487	1									
3	0.9903	1.87	0.515	1	0.59								
4	1.5391	2.55	0.535	2	—	1.007							
5	2.1212	3.22	0.565	2	—	0.81	1.445						
6	2.7282	3.92	0.588	2	—	0.57	1.36	1.89					
7	3.3569	4.63	0.606	2	—	0.42	1.17	1.89	2.37				
8	4.0038	5.35	0.623	2	—	0.33	1.02	1.77	2.43	2.86			
9	4.6660	6.08	0.638	3	—	—	0.91	1.65	2.37	2.98	3.38		
10	5.3416	6.82	0.653	3	—	—	0.84	1.54	2.28	2.98	3.55	3.92	
11	6.0290	7.57	0.666	3	—	—	0.78	1.46	2.19	2.93	3.60	4.14	4.47

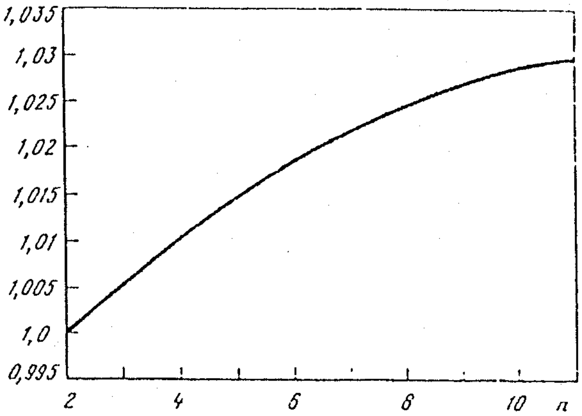

Fig. 4. Ratio of troughputs for the simplified improved algorithm and the best known algorithm for ternary feedback and n-conflict.

4. Discussion and Summary

An algorithm based on an n-ary feedback is proposed. In Fig. 4. the ratio between the simplified improved algorithm and the best results known for a ternary feedback [3], which in the case of a success does not provide information on the number of successful packets, is plotted. The improvement in the throughput grows with n up to 3% for $n = 11$. However, the performance of the algorithm is still below the upper limit for a ternary feedback [3].

It is an open question to find how large, is the throughput of the optimal algorithm for a channel with n-ary feedback in comparison with the throughput of the optimal algorithm for a channel with ternary feedback.

So, using (3.1)–(3.6) and the optimization procedure (3.7)–(3.13), it is interesting to find numerical results for the maximum throughput R for the improved algorithm.

LITERATURE CITED

1. R. G. Gallager, "A Perspective on Multiaccess Channels," *IEEE Trans. Inform. Theory*, 31, 124–142 (1985).
2. B. S. Tsybakov, "Survey of USSR Contributions to Random Multiple-Access Communications," *IEEE Trans. Inform. Theory*, 31, 143–165 (1985).
3. B. S. Tsybakov, V. A. Mikhailov, and N. B. Likhanov, "Bounds for Packet Transmission Rate in a Random-Mutiple-Access System," *Probl. Peredachi Inform.*,19, 61–81 (1983).
4. R. Z. Khasminskii, "Controllable Markov Process and Optimization of Random Multiple Access Algorithms", *Dokl. Akad. Nauk SSSR*, 301, 1057–1060 (1988).
5. B. S. Tsybakov and N. B. Likhanov, "RMA with Known Times of Appearance of Successfully Transmitted Packets," *Probl. Peredachi Inform.*, 26, 62–74 (1990).

Translated from Russian by B.S.Tsybakov and N.B.Likhanov