ISSN 0032-9460, Problems of Information Transmission, 2007, Vol. 43, No. 4, pp. 344–352. © Pleiades Publishing, Inc., 2007.
Original Russian Text © G.S. Evseev, A.M. Turlikov, 2007, published in Problemy Peredachi Informatsii, 2007, Vol. 43, No. 4, pp. 83–92.

COMMUNICATION NETWORK THEORY

Interrelation of Characteristics of Blocked RMA Stack Algorithms

G. S. Evseev and A. M. Turlikov

St. Petersburg State University of Aerospace Instrumentation
egs@vu.spb.ru turlikov@vu.spb.ru

Received April 25, 2007

Abstract—A method to analyze the duration of collision resolution for blocked RMA stack algorithms is proposed. Simple formulas are obtained that express the average length of a collision resolution interval for the modified (frugal) algorithm in a noisy and in a noiseless channel, as well as for the basic algorithm in a noisy channel, through the corresponding parameters for the basic algorithm in a noiseless channel. From estimates of the throughput of the basic algorithm in a noiseless channel, estimates for the throughput in the other three cases are directly constructed.

DOI: 10.1134/S0032946007040072

1. INTRODUCTION

In [1, 2] there was first considered an algorithm using which infinitely many users sharing a communication channel can transmit data with a finite average delay provided that the arrival rate is bounded. Using the terminology of [3], the supremum of arrival rates for which the average delay is finite will be referred to as the throughput of an algorithm.

In what follows, we call the algorithm considered in [1, 2] the basic algorithm (in [4], the term *basic stack algorithm with blocked access* was used). In [1, 2] it was also noted that from the basic algorithm one can also obtain an algorithm with a higher throughput, which will be referred to as the modified (or frugal) algorithm (in [4] it was called the *frugal stack algorithm with blocked access*). The analysis made in [1, 2] has shown that for the throughputs R_b and R_f of, respectively, the basic and modified (frugal) algorithms, we have

$$\begin{aligned} 0.3464 < R_b < 0.3471, \\ 0.3752 < R_f < 0.3759. \end{aligned} \tag{1}$$

After the publication of [1, 2], investigation of algorithms was developed in two directions: refinement of estimates for throughputs and extending results to the case of noisy channels where false collisions are possible.

In [4], results on finding estimates for the throughput in a noiseless channel were summed up and the following refined estimates were given:

$$0.34657320 < R_b < 0.34657397, \tag{2}$$

$$0.3753690 < R_f < 0.3753698. \tag{3}$$

In [5] there was introduced a model of a noisy channel where false collisions may occur, and methods for computing estimates for the throughput in a noisy channel were described. In [6] it was shown that the throughput of the basic algorithm in a noisy channel is approximately

$$\frac{1-2q}{1-q} \cdot 0.346,$$

where $q < 0.5$ is the probability of a false collision.

The aim of the present paper is to show how one can estimate the throughput of the frugal algorithm based on the results of analysis of the basic algorithm in a noiseless channel and to extend the results for the case of a noisy channel. It should be noted that the possibility of this approach was first demonstrated by the authors as early as in 1991 [7].

The paper is organized as follows. In Section 2 we briefly describe the system model and the basic algorithm according to [1]. The algorithm is represented as a tree describing the collision resolution process (we use the abbreviation CRT for the tree). We describe constructions of estimates for the throughput of the basic algorithm from [1, 2, 4, 8, 9]. Then we formulate a statement for the CRT, which is used in what follows. In Section 3 we describe the frugal algorithm from [1] and show how one can find estimates for the throughput of this algorithm using the CRT for the basic algorithm.

In Section 4 we briefly describe the model of a noisy multiple-access channel from [5]. In the framework of this model, we compute estimates for the throughput of the basic algorithm in a noisy channel. Finally, we give estimates for the throughput of the frugal algorithm in a noisy channel.

2. SYSTEM MODEL AND THE BASIC ALGORITHM

We consider the model of a random multiple access system from [1]. In the system there are infinitely many users and a communication channel; the channel input and output are available to all users. Users exchange packets through the communication channel. All packets are assumed to have the same length. The packet transmission time is taken to be the time unit. Intervals between arrivals of new packets in the system are statistically independent random variables exponentially distributed with mean $\frac{1}{\lambda}$. The parameter λ is called the packet arrival rate in the system. It equals the average number of packets that arrive at the system per time unit.

We formulate a number of assumptions [1] concerning the communication channel and the way of users' access to it.

Assumption 1. Time is slotted. All slots have the same length equal to the transmission time of one packet. Slots are enumerated by nonnegative integers; the slot with number t corresponds to the time interval $[t, t+1)$. Below we for brevity refer to the slot with number t as slot t. Slot separation time instants are known to all users. A user may start transmission at the beginning of a slot only.

Assumption 2. In each slot there can occur one of the following three events:

- one user transmits a packet in the slot (event S: success);
- no user transmits in the slot (event E: empty);
- two or more users transmit in the slot (event C: collision).

Assumption 3. Users observing the channel output know at the end of a slot about the event in the slot.

Assumption 4. A user observing the channel output determines for sure which of the three possible events has occurred in the channel.

Assumption 5. Each user has a buffer to store one packet. A user stores the packet from its arrival until the successful transmission. A packet arrived within time interval $[t-1, t)$ can be transmitted not earlier than in a slot $t' \geq t$.

Algorithms considered here belong to the class of algorithms for which the system operation is described by a sequence of sessions [1]. To each session there corresponds a subset of users that transmitted their packets in the first slot of the session (slot t_0). The number k of packets transmitted simultaneously is called the collision multiplicity in a slot. If in slot t_0 the events E (i.e., $k = 0$) or S (i.e., $k = 1$) are observed, the first slot of a session is also its last slot. Otherwise, the session ends not earlier than all packets that collided at slot t_0 are transmitted. The rule of determining the last slot of a session is based on the analysis of the observed sequence of events in slots; therefore, conclusions of users about the last slot of a session coincide and are made simultaneously. Sessions follow each other without gaps; i.e., the first slot of a next session adjoins the last slot of the previous one. We consider only so-called blocked algorithms [1], for which all packets that arrive during a current session can be transmitted in the next session only; i.e., in slots of each session there are transmitted only packets that collided in the first slot of the session.

We say that a conflict of multiplicity k is resolved during a session if k packets were transmitted in the first slot of this session. A conflict resolution algorithm (CRA) is specified, first, by a rule according to which all users determine the last slot of a session and, second, by a rule according to which each participant of a session finds slots to retransmit his packet in the session.

To describe a CRA, it is convenient to introduce an undirected graph G, which is an infinite binary tree whose nodes correspond to slots in the communication channel and the root corresponds to the first slot of a session t_0, where a collision of multiplicity k occurred. Two nodes of G are said to be adjacent if they have a common parent. For definiteness, we will call one of these nodes the higher child and the other, the lower child (this makes sense if we draw the tree from left to right, from parents to children). Nodes of G that are higher or lower children, will be called for short the higher and lower nodes, respectively. In the process of conflict resolution, at the channel output the users observe a sequence of events from the set $\{E, S, C\}$ and mark the corresponding nodes of the tree G with symbols E, S, and C. As a result of conflict resolution, all nodes in the tree G that correspond to slots of the session are marked, and a binary subtree of G that corresponds to the session with the root node P_{root} is selected. We call this subtree the conflict resolution tree (CRT). Thus, to describe an algorithm, we have, first, to define the correspondence between slots of a session and nodes of the CRT and, second, to specify in which slots each participant of the session must transmit his packets. For the basic algorithm, the corresponding rules are given below.

1. The correspondence between slots of a session and nodes of a CRT is defined by induction.

To the first slot of the session, t_0, there corresponds the root of the tree G.

If to a current slot of a session there corresponds a vertex P_{cur} in G marked with symbol C, then to the next slot of the session there corresponds the higher child of P_{cur} in G. If to a current slot of a session there corresponds a node P_{cur} in G marked with either E or S, then to the next slot of the session there corresponds a node P_{next} in G with the following properties:

- P_{next} is not yet marked;
- A node adjacent to P_{next} is already marked;
- Among all nodes with these two properties, we choose the one with the smallest number of edges in the path from it to P_{cur}.

If there is no node P_{next} satisfying the required properties, the construction of the CRT terminates.

2. The rule to choose slots in which participants of the session should transmit packets is also defined by induction.

In the first slot of a session, all participants of the session transmit packets.

If a user (participant of the session) transmitted a packet in a slot corresponding to a node P_{cur} which was marked with S at the end of the slot, this packet is no more transmitted.

If a user (participant of the session) transmitted a packet in a slot corresponding to a node $P_{\rm cur}$ which was marked with G at the end of the slot, then he equiprobably marks one of the two children of $P_{\rm cur}$ in G with a temporary label *SfT* (selected for transmission). After that, the user will transmit the packet only in the slot t corresponding to the node marked with *SfT*; at the end of the slot t, the temporary label *SfT* is deleted.

In the CRT constructed according to the above-described rules, all terminal nodes of the tree have labels E or S; the other nodes are marked with C.

The number of time units required to construct the tree is called the collision resolution time. For the basic algorithm, the number of nodes in the CRT is equal to the collision resolution time. Denote by τ the random variable equal to the collision resolution time. The conditional expectation

$$T_k = \mathbf{E}[\tau \,|\, \text{a collision of multiplicity } k \text{ is resolved}]$$

will be called the average resolution time for a collision of multiplicity k. Below, when considering expectation, for simplicity of notation we omit the condition indicating the collision multiplicity k.

It was shown in [1] that with the use of the ratio $\frac{k}{T_k}$ one can establish the following bounds on the algorithm throughput:

$$\liminf_{k\to\infty} \frac{k}{T_k} < R_b < \limsup_{k\to\infty} \frac{k}{T_k}. \tag{4}$$

In [1, 2], methods to estimate the upper and lower bounds in (4) are proposed and numerical values of the upper and lower bounds for the throughput of the basic algorithm (1) are given.

In [8] there was considered a generalization of the basic algorithm to the case where “branching” in the tree is into not two but m branches; in particular, the behavior of T_k for large values of k was analyzed. Formula (28) from [8] describes the asymptotic form of the ratio $\frac{T_k}{k}$ for any $m \geq 2$. A particular case of $m = 2$ and $k \gg 1$ is presented in [9] in the form

$$\frac{T_k}{k} = \frac{2}{\ln 2} + A\sin(2\pi \log_2 k + \varphi) + O\Big(\frac{1}{k}\Big), \tag{5}$$

where $A = 3.127 \cdot 10^{-6}$ and $\varphi = 0.9826$.

From (5) and (4) it immediately follows that

$$\left(\frac{2}{\ln 2} + A\right)^{-1} < R_b < \left(\frac{2}{\ln 2} - A\right)^{-1}. \tag{6}$$

Substitution of the values of A into (6) yields numerical values for the refined estimates (1) of the throughput of the basic algorithm, which coincide with the values obtained in [4] using another method.

To apply the results of the described analysis of the basic algorithm directly to the analysis of the frugal algorithm, we use the following statement, whose validity immediately follows from the construction of the CRT.

Proposition 1. *Introduce the following notation*: n *is the total number of nodes in the CRT*; n_e, n_s, *and* n_c *are the numbers of nodes labeled with* E, S, *and* C, *respectively. For any* $k > 1$, *for the number of nodes in the tree for resolution of a collision of multiplicity* k *we have*

$$n_s = k, \tag{7}$$

$$n_c = \frac{n-1}{2}, \tag{8}$$

$$n_e = \frac{n+1}{2} - k. \tag{9}$$

3. FINDING THE THROUGHPUT OF THE FRUGAL ALGORITHM

Consider the operation of the basic algorithm in the following situation. Assume that slot t corresponds to a higher node and that users observing the channel output find by the end of this slot that there was no transmission in the slot. Then at the end of slot t this higher node is labeled by E. In this case, according to the basic algorithm, slot $t+1$ corresponds to a lower node, and in this slot all packets that were previously transmitted in slot $t-1$ are transmitted again. Thus, in slot $t+1$ there again occurs a collision. In [1, 2], a modified (frugal) algorithm was proposed, which prohibits the occurrence of such repeated collisions.

Similarly to the basic algorithm, operation of the frugal algorithm can be described by a tree. As compared with the basic algorithm, only the rules of establishing correspondence between slots of a session and nodes of CRT are changed. If to a current slot t there corresponds a node P_{cur} marked with E and this is a higher node, then a modified rule is used to establish a correspondence between slots of the session and nodes of the CRT; in all other cases, rules of the basic algorithm described in Section 2 are used. To describe the modified rule, denote by P_{tmp} the node adjacent to P_{cur}. This will be a lower node. According to the modified rule, in slot t the node P_{tmp} is labeled together with P_{cur}, and the node P_{tmp} gets the label C. Thus, two nodes in the CRT are put into correspondence with the slot t.

It follows from this description that the collision resolution time for the frugal algorithm can be less than the number of nodes in the CRT.

Proposition 2. *For any $k > 1$, the average collision resolution time T_k^f for the frugal algorithm and the average collision resolution time T_k for the basic algorithm are related by*

$$T_k^f = \frac{3}{4}T_k + \frac{k}{2} - \frac{1}{4}. \tag{10}$$

Proof. Consider a CRT for a collision of multiplicity $k > 1$. For all higher and lower nodes, let us introduce the additional labels H and L, respectively. Denote by n_{eh} the number of nodes labeled with E and H, i.e., the number of higher nodes that were chosen for retransmission by no user. Similarly, denote by $n_{e\ell}$ the number of nodes labeled with E and L. Using this notation together with the notation introduced in Proposition 1, we obtain an expression relating the collision resolution time for the frugal algorithm and the number of nodes in the CRT:

$$\tau^f = n_e + n_s + (n_c - n_{eh}) = n_{e\ell} + k + n_c. \tag{11}$$

For the expectations of the random variables n_{eh}, $n_{e\ell}$, and n_e, we have

$$\mathbf{E}[n_{eh}] = \mathbf{E}[n_{e\ell}] = \frac{\mathbf{E}[n_e]}{2}.$$

Using this equality together with (8) and (9), pass to expectations in (11):

$$T_k^f = \mathbf{E}[\tau^f] = \frac{\mathbf{E}\Big[\dfrac{n+1}{2} - k\Big]}{2} + k + \mathbf{E}\left[\frac{n-1}{2}\right]. \tag{12}$$

Now (10) is obtained from (12) by simple transformations. △

It follows from (5) and (10) that the asymptotics in k for the frugal algorithm are

$$\frac{T_k^f}{k} = \frac{3}{2\ln 2} + \frac{1}{2} + \frac{3}{4}A\sin(2\pi\log_2 k + \varphi) + O\Big(\frac{1}{k}\Big), \tag{13}$$

where A and φ are the same as in (5).

Using (4) and (10), we obtain the following bounds for the throughput of the frugal algorithm:

$$\left(\frac{3}{2\ln 2}+\frac{1}{2}+\frac{3}{4}A\right)^{-1} < R_f < \left(\frac{3}{2\ln 2}+\frac{1}{2}-\frac{3}{4}A\right)^{-1}. \tag{14}$$

Substituting the numerical value of A into (14), we obtain

$$0.375369048 < R_f < 0.375369709.$$

Note that these bounds improve the estimates of [4] in the seventh digit after the decimal point.

4. FINDING THE THROUGHPUT OF THE BASIC ALGORITHM IN A CHANNEL WITH FALSE COLLISIONS

In [5] there was proposed a model of a noisy channel based on the same assumptions as the model described in Section 2; only Assumption 3 is changed.

Assumption 3 for a noisy channel model. An event S is observed by a user in a current slot as an event C with probability q_1 and as an event S with probability $1-q_1$. Similarly, an event E is observed by a user in a current slot as an event C with probability q_0 and as an event E with probability $1-q_0$. All users make false decisions simultaneously; i.e., all users have the same sequence of observed events.

It was noted in [5] that the basic algorithm properly works in a channel with false collisions for $q_0 < 0.5$ and $q_1 < 1$. The average resolution time for collisions of multiplicity 0 and 1 can be computed by the formulas

$$T_0^n = \frac{1}{1-2q_0},$$
$$T_1^n = \frac{1-2q_0+q_1}{(1-q_1)(1-2q_0)}.$$

Proposition 3. *For the basic algorithm, for any $k>1$, the average collision resolution time T_k^n in a noisy channel and the average collision resolution time T_k in a noiseless channel are related by*

$$T_k^n = T_k\frac{T_0^n+1}{2} + k(T_1^n - T_0^n) + \frac{T_0^n-1}{2}. \tag{15}$$

Proof. Consider the collision resolution tree for a collision of multiplicity k in a noisy channel. Because of false collisions, labels in the tree do not necessarily correspond to events in the channel. Moving along the tree starting from the root, let us additionally label nodes according to the following rule.

If a node P corresponds to a slot t where a false event C was observed, the node P gets an additional label E or S depending on the real event in slot t, and all descendants of P do not get additional labels.

It immediately follows from this rule of assigning additional labels that nodes that get additional labels correspond to terminal nodes in some CRT for a collision of multiplicity k in a noiseless channel, and the average number of nodes in such a tree is T_k. The root of this tree and all its internal nodes must have label C; denote the number of these nodes by m_c. Denote by M_e and M_s the sets of nodes with additional labels E and S, respectively, and let m_e and m_s be the number of nodes with these additional labels; i.e., $|M_e| = m_e$ and $|M_s| = m_s$. For this tree, Proposition 1 is valid.

Let us return to a CRT in a noisy channel. Denote by $L(P)$ the number of nodes in a subtree with root P. Using this and previously introduced notation, the total number of nodes in a CRT for a noisy channel and, accordingly, the collision resolution time τ^n in a noisy channel can be computed as follows:

$$\tau^n = m_c + \sum_{P\in M_e} L(P) + \sum_{P\in M_s} L(P). \tag{16}$$

Passing to expectations and taking into account that $\mathbf{E}[L(P)] = T_1^n$ if P is additionally labeled by S and $\mathbf{E}[L(P)] = T_0^n$ if P is additionally labeled by E, we obtain

$$T_k^n = \mathbf{E}[m_c] + kT_1^n + \mathbf{E}[m_e]T_0^n. \tag{17}$$

Applying Proposition 1 for the random variables m_c and m_e and expressing the expectations of these variables via T_k, we get

$$T_k^n = \frac{T_k - 1}{2} + kT_1^n + \left(\frac{T_k + 1}{2} - k\right) T_0^n. \tag{18}$$

Simplifying (18), we obtain (15). △

Using expression (4), which describes the asymptotic behavior of the ratio $\frac{T_k}{k}$, from (15) we get

$$\frac{T_k^n}{k} = \frac{2}{\ln 2} + A\sin(2\pi\log_2(k) + \varphi)\frac{T_0^n + 1}{2} + (T_1^n - T_0^n) + O\Big(\frac{1}{k}\Big). \tag{19}$$

From this equality and (4), we obtain the following expressions for the lower, R_{bn}^-, and upper, R_{bn}^+, bounds on the throughput of the basic algorithm R_{bn} in a channel with false collisions:

$$R_{bn}^- = \left(\frac{T_0^n + 1}{\ln 2} + (T_1^n - T_0^n) + \frac{A(T_0^n + 1)}{2}\right)^{-1},$$
$$R_{bn}^+ = \left(\frac{T_0^n + 1}{\ln 2} + (T_1^n - T_0^n) - \frac{A(T_0^n + 1)}{2}\right)^{-1}.$$

Taking into account that

$$\frac{T_0^n + 1}{2} = \frac{1 - q_0}{1 - 2q_0}, \qquad T_1^n - T_0^n = \frac{2(q_1 - q_0)}{(1 - q_1)(1 - 2q_0)},$$

we obtain final expressions for the bounds on the throughput:

$$R_{bn}^- = \frac{1 - 2q_0}{1 - q_0}\left(\frac{2}{\ln 2} + \frac{2(q_1 - q_0)}{(1 - q_1)(1 - q_0)} + A\right)^{-1},$$
$$R_{bn}^+ = \frac{1 - 2q_0}{1 - q_0}\left(\frac{2}{\ln 2} + \frac{2(q_1 - q_0)}{(1 - q_1)(1 - q_0)} - A\right)^{-1}.$$

In a particular case of $q_1 = q_0 = q$, we get

$$\frac{1 - 2q}{1 - q} \cdot 0.34657320 < R_{bn} < \frac{1 - 2q}{1 - q} \cdot 0.34657397. \tag{20}$$

The frugal algorithm in a noiseless channel has a higher throughput than the basic algorithm. However, as was noted in [5], in a noisy channel the frugal algorithm has infinite average delay for any $q_0 > 0$, so that its throughput is zero. We confine ourselves with the case where $q_0 = 0$ and $q_1 > 0$; i.e., false collisions may only occur in slots where there is no transmission. In this case the frugal algorithm has a nonzero throughput. Using the above-described approaches, we obtain bounds on the throughput of the frugal algorithm.

It is easy to show that for the frugal algorithm in the case $q_0 = 0$ and $q_1 > 0$, the average resolution time for a collision of multiplicity 1 is as follows:

$$T_1^{fn} = \frac{1 + 2q_1}{(1 - q_1)}.$$

Consider the resolution of a collision of some multiplicity $k > 1$. Repeat the arguments of Section 2, taking into account the fact that from each terminal node labeled with S in a collision resolution tree for a noiseless channel there "grows a subtree." The average number of nodes in such a subtree is T_1. Using the notation introduced in the proof of Proposition 2, we obtain

$$T_k^{fn} = \mathbf{E}[\tau^{fn}] = \mathbf{E}[n_e] + \mathbf{E}[n_s]T_1^{fn} + [n_c - n_{eh}] = \mathbf{E}[n_{e\ell}] + kT_1^{fn} + \mathbf{E}[n_c]$$
$$= \frac{\dfrac{T_k + 1}{2} - k}{2} + kT_1^{fn} + \frac{T_k - 1}{2} = \frac{3}{4}T_k + \frac{k(1 + 5q_1)}{2(1 - q_1)} - \frac{1}{4}.$$

Hence we get bounds for the frugal algorithm in a noisy channel:

$$R_{fn}^- = \left(\frac{3}{2\ln 2} + \frac{(1 + 5q_1)}{2(1 - q_1)} + \frac{3}{4}A\right)^{-1},$$
$$R_{fn}^+ = \left(\frac{3}{2\ln 2} + \frac{(1 + 5q_1)}{2(1 - q_1)} - \frac{3}{4}A\right)^{-1}.$$

Let us find for which values of the probability q_1 the frugal algorithm has a higher throughput than the basic algorithm, and vice versa.

For $q_0 = 0$ and $q_1 > 0$, the bounds for the basic algorithm take the form

$$R_{bn}^- = \left(\frac{2}{\ln 2} + \frac{2q_1}{1 - q_1} + A\right)^{-1},$$
$$R_{bn}^+ = \left(\frac{2}{\ln 2} + \frac{2q_1}{1 - q_1} - A\right)^{-1}.$$

For the values of q_1 such that $R_{bn}^+ < R_{fn}^-$, the throughput of the frugal algorithm is higher. For the values of q_1 such that $R_{bn}^- > R_{fn}^+$, the throughput of the basic algorithm is higher.

Denote by q^f and q^b the solutions of the equations $R_{bn}^+ - R_{fn}^- = 0$ and $R_{bn}^- - R_{fn}^+ = 0$, respectively.

Solving the equations, we obtain

$$q^f = \frac{1 - \left(1 + \frac{7}{2}A\right)\ln 2}{1 + \left(1 - \frac{7}{2}A\right)\ln 2},$$
$$q^b = \frac{1 - \left(1 - \frac{7}{2}A\right)\ln 2}{1 + \left(1 + \frac{7}{2}A\right)\ln 2}.$$

If $q_1 < q^f$, the frugal algorithm has a higher throughput; if $q_1 > q^b$, the throughput of the basic algorithm is higher. If $q^f \le q_1 \le q^b$, the relation between the throughputs of the basic and frugal algorithms is indefinite. Taking into account that $A = 3.127 \cdot 10^{-6}$, we may say that $\dfrac{1 - \ln 2}{1 + \ln 2}$ is a threshold for q_1 below which the throughput of the frugal algorithm is higher.

5. CONCLUSION

The described approach to the analysis of the collision resolution time in a multiple-access channel allows us, using known properties of the trees, to obtain formulas for computing the average collision times for both the frugal and basic algorithms in a noisy channel via the corresponding values for the basic algorithm in a noiseless channel. These formulas establish a correspondence between the throughput R_b of the basic algorithm and the throughput R_f of the frugal algorithm in a noiseless channel:

$$R_f \approx \left(\frac{3}{4R_b} + \frac{1}{2}\right)^{-1}.$$

We can also obtain a relation between the throughput R_{bn} of the basic algorithm in a noisy channel and that in a noiseless channel:

$$R_{bn} \approx \frac{1-2q_0}{1-q_0}\left(\frac{1}{R_b} + \frac{2(q_1-q_0)}{(1-q_1)(1-q_0)}\right)^{-1},$$

where q_0 and q_1 are probabilities of false collision for an empty slot and a slot where one user transmits.

Note that the analysis can be generalized to the case of a channel where the duration of a slot is determined by an event observed in the slot.

REFERENCES

1. Tsybakov, B.S. and Mikhailov, V.A., Free Synchronous Packet Access in a Broadcast Channel with Feedback, *Probl. Peredachi Inf.*, 1978, vol. 14, no. 4, pp. 32–59 [*Probl. Inf. Trans.* (Engl. Transl.), 1978, vol. 14, no. 4, pp. 259–280].

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

3. Tsybakov, B.S., Survey of USSR Contributions to Random Multiple-Access Communications, *IEEE Trans. Inform. Theory*, 1985, vol. 31, no. 2, pp. 143–165.

4. Györfi, L., Györi, S., and Massey, J.L., Principles of Stability Analysis for Random Accessing with Feedback, *Multiple Access Channels: Theory and Practice*, Biglieri, E. and Györfi, L., Eds., Amsterdam: IOS Press, 2007, pp. 214–249.

5. Evseev, G.S. and Ermolaev, N.G., Performance Evaluation of the Collision Resolution for a Random-Access Noisy Channel, *Probl. Peredachi Inf.*, 1982, vol. 18, no. 2, pp. 101–105.

6. Evseev, G.S. and Turlikov, A.M., Throughput Analysis for a Noise-Resistant Multiple Access Algorithm, *Probl. Peredachi Inf.*, 1986, vol. 22, no. 2, pp. 104–109.

7. Evseev, G. and Turlikov, A., The Multiple-Random-Access Algorithms Analysis Based on Tree Properties, in *Proc. 5th Joint Soviet–Swedish Int. Workshop on Information Theory, Moscow, 1991.*

8. Janssen, A.J.E.M. and de Jong, M.J.M., Analysis of Contention Tree Algorithms, *IEEE Trans. Inform. Theory*, 2000, vol. 46, no. 6, pp. 2163–2172.

9. Györfi, L. and Györi, S., Analysis of Tree Algorithm for Collision Resolution, in *Proc. 2005 Int. Conf. on Analysis of Algorithms, Barcelona*, Nancy: Assoc. Discrete Math. Theor. Comput. Sci., 2005, pp. 357–364.