A Class of Limited Sensing Random Access Algorithms with Resistance to Feedback Errors and Effective Delay Control

Anthony T. Burrell and Titsa P. Papantoni

Abstract: **We present and analyze a class of limited sensing random access algorithms with powerful properties. The algorithms are implementable in wireless mobile environments and their operational properties are simple. Their throughput in the worst case of the limit Poisson user model is 0.4297, while this throughput degrades gracefully in the presence of channel feedback errors.**

Index Terms: **Limited sensing, random access algorithms.**

I. INTRODUCTION

We consider packet networks. When independent users whose identities are initially unknown to the system transmit through a single common channel, the deployed transmission protocol must necessarily belong to the class of random access algorithms (RAAs) [1]. Such scenarios arise, for example, in the Ethernet environment, as well as at the signaling stage of packet radio and cellular telephony. The class of RAAs includes three subclasses, depending on the level of channel sensing required by the users in the protocol operations. The minimal sensing (MS) subclass requires that each user sense the transmission channel only at times when he transmits; the ALOHA protocol and its variations belong in this subclass. The limited sensing (LS) subclass requires that each user sense the transmission channel continuously from the time he generates a packet to the time that this packet is successfully transmitted; this subclass contains a number of algorithms [2]–[4] with various characteristics discussed in the paragraph below. The full sensing (FS) subclass requires that each user know the overall history of the transmission channel from the time when the system starts operating [5]; this subclass contains algorithms [1], [5], whose operations are of just academic interest, since the level of channel sensing they require is clearly non-implementable.

Although no formal proof has been devised yet to this effect, it seems that minimal sensing is inadequate for the development of stable RAAs; the instability of the ALOHA protocol and its variations is well known [6], where as the number of users increases, the throughput of the protocol approaches rapidly zero. In contrast, the MS subclass includes a number of stable protocols, while the channel sensing required by the subclass is clearly reasonable and implementable. The first algorithm devised in the MS subclass is described in [6]; its operations are simple, but its throughput in the presence of the limit Poisson user model[1] is low. The limited sensing modification [3] of Gallager's algorithm [7] has throughput 0.487 in the presence of the limit Poisson user model, but its operations are quite complex and it reaches deadlocks in the presence of channel errors. The algorithm in [4] is simple to implement and attains throughput 0.4297 in the presence of the limit Poisson user model; it also possesses the highest resistance to channel errors among all the existing to this point algorithms.

In this paper, we present a class of LS algorithms that attain the same throughput and also possess similar resistance to channel errors as the algorithm in [4]. The algorithms exhibit varying delay behavior for low rates and possess advantageous operational characteristics. We will focus on a single member in the class. The collision resolution operations of the latter algorithm can be depicted by a 3-cell stack; we thus name the algorithm, *the 3-cell algorithm.*

The organization of the paper is as follows. In Section II, the system model is presented and the algorithms in the class are described, with special emphasis on the 3-cell algorithm. In Section III, the throughput and delay analyses of the 3-cell algorithm in the presence of the limit Poisson user model are included. In Section IV, the performance of the 3-cell algorithm in the presence of feedback channel errors is discussed. In Section V, additional comparisons of the 3-cell algorithm with the algorithm in [4] and the Ethernet protocol are presented. In Section VI, some conclusions are drawn.

II. THE SYSTEM MODEL AND THE ALGORITHMS

We assume packet-transmitting users, a slotted channel, binary collision-versus-non-collision (C-NC) feedback after each slot, zero propagation delays, and initially absence of feedback errors. We also assume that collided packets are fully destroyed and retransmission is then necessary. Time is measured in slot units; slot t occupies the time interval $[t, t+1)$ and x_t denotes the feedback that corresponds to slot t; x_t = C and x_t = NC then represent collision and non-collision in slot, respectively.

Each algorithm in the class is independently and asynchronously implemented by the users. Indeed, in the limited sensing environment, it is required that each user monitor the channel feedback only from the time he generates a packet to the time that this packet is successfully transmitted. Therefore, the users' knowledge of the channel feedback history is asyn-

Manuscript received April 28, 2004; approved for publication by Boris Tsybakov, Division II Editor, May 26, 2005.

A. T. Burrell is with the Oklahoma State University, USA, email: tburrell@cs.okstate.edu.

T. P. Papantoni is with the University of Colorado, USA, email: tosavk7@hotmail.com.

[1]The limit Poisson user model consists of infinitely many independent Bernoulli users and it represents a worst case for the study of RAAs, within a large class of user models, as proven in [8].

chronous. The objective in this case is to prevent new arrivals from interfering with some collision resolution in progress. This is possible if each user can decide whether or not a collision resolution is in progress within a finite number of slots from the time he generates a new packet. The possibility of such decision can only be induced by the operational characteristics of the algorithm. As we will explain below, each algorithm in the class possesses the appropriate operational characteristics for such decisions.

Each algorithm in the class utilizes a window of size Δ as a operational parameter and induces a sequence of consecutive collision resolution intervals (CRIs). The window length Δ is subject to optimal selection for throughput maximization. Each CRI corresponds to the successful transmission of all packet arrivals within an arrival interval of length Δ. The length of the CRI is determined by the number of users in the window Δ and the algorithmic steps of the collision resolution process. The placement of the Δ-size window on the arrival access is determined asynchronously by the users. We will first describe the collision resolution process induced by the algorithm. Then, we will explain the process which determines the placement of the Δ-size window per CRI.

The algorithmic class contains algorithms whose collision resolution process can be depicted by a stack with finite number of cells. Let us consider this algorithm in the class which can be described by a K-cell stack. Then, in the implementation of the collision resolution process, each user utilizes a counter whose values lie in the set of integers, $[1, 2, \cdots, K]$. We denote by r_t the counter value of some user at time t. The K different possible values of the counter place the user in one of the K cells of a K-cell stack. When his counter value is 1, the user transmits; he withholds at $K-1$ different stages otherwise. When a CRI begins, all users in the Δ-size window set their counters at 1; thus, they all transmit within the first slot of the CRI. If the window contains at most one packet, the first slot of the CRI is a non-collision slot and the CRI lasts one slot. If the window contains at least two packets, instead, the CRI starts with a collision which is resolved within the duration of the CRI via the following rules:

The user transmits in slot t if and only if $r_t = 1$. A packet is successfully transmitted in t if and only if $r_t = 1$ and $x_t =$ NC.

The counter values transition in time as follows.

If $x_{t-1} =$ NC and $r_{t-1} = j$, $j = 2, 3, \cdots, K$, then $r_t = j - 1$.
If $x_{t-1} =$ C and $r_{t-1} = j, j = 2, 3, \cdots, K$, then $r_t = j$.
If $x_{t-1} =$ C and $r_{t-1} = 1$, then $r_t =$

$$\begin{array}{l} 1, \text{ w.p. } 1/K \\ 2, \text{ w.p. } 1/K \\ 3, \text{ w.p. } 1/K \\ \quad\vdots \\ K, \text{ w.p. } 1/K. \end{array}$$

From the above rules, it can be seen that a CRI that starts with a collision slot ends with K consecutive non-collision slots, an event which can not occur at any other instant during the CRI. Thus, the observation of K consecutive non-collision slots signals the certain end of a CRI to all users in the system; it either signifies the end of a CRI that started with a collision or the end of a sequence of K consecutive length-one CRIs. Therefore, a user who arrives in the system without any knowledge of the channel feedback history can synchronize with the system upon the observation of the first K-tuple of consecutive non-collision slots. This observation leads to the asynchronous by the users generating of the size-Δ window placement on the arrival axis. Specifically, if a CRI ends with slot t, the window of the next CRI is selected with its right most edge $K-1$ slots to the left of slot t and it contains those packets whose *updates* fall in the interval $(t-K+1-\Delta, t-K+1)$. The *updates* $\{t^k\}$ of a packet are generated as follows: Let t_0 be the slot within which a packet is generated. Then define t^0 to be equal to t_0. Starting with slot t^0, the corresponding user senses continuously the channel feedbacks. He does so passively, until he observes the first K-tuple of consecutive NC slots, ending with slot t_1. If $t^0 \in (t_1 - K + 1 - \Delta, t_1 - K + 1)$, the user participates in the CRI that starts with slot $t_1 + 1$. Otherwise, he updates his arrival instant to $t^1 = t^0 + \Delta$ and waits passively until the end of the latter CRI, ending with slot t_2. If $t^1 \in (t_2 - K + 1 - \Delta, t_2 - K + 1)$, the user participates in the CRI which starts with slot t_2; otherwise, he updates his arrival instant by Δ again and repeats the above process. In general, if $\{t_n\}_{n\geq 1}$ denotes the sequence of consecutive CRI endings since the first K-tuple of consecutive NC slots, the packet participates in the k-th CRI if $t^{k-1} \in (t_k - K + 1 - \Delta, t_k - K + 1)$ and $t^n \notin (t_{n+1} - K + 1 - \Delta, t_n - K + 1)$, for all $n \leq k - 2$.

We note that multiple-cell algorithms were considered in [9] and [10]. We also note that the algorithm in [4] belongs in the class examined in this paper. We expect that the algorithms in the class will have the same throughput with that of the latter algorithm, but different delay and resistance to feedback errors behaviors. As K increases, we expect that the delays for low rates will increase. In the following sections, we focus on the detailed analysis of the 3-cell algorithm.

III. THROUGHPUT AND DELAY ANALYSIS OF THE 3-CELL ALGORITHM

In this section, we present the throughput and delay analyses of any one algorithm in Section II, with specific results included for the 3-cell algorithm only. We adopt the limit Poisson user model. Indeed, as proven in [8], for a large class of random access algorithms, as the user population increases the stability of an algorithm in the class is determined by its throughput under the Poisson user model. Let CRPs denote the beginnings of CRIs.

Consider the system model and the algorithms in Section II. Consider any one of the algorithms in the latter class be active. Then, let the system start operating at time zero. Let t_i, $i \geq 1$ be the sequence of successive CRP's and let X_i be the lag at t_i. The sequence X_i, $i \geq 1$ is a Markov chain with state space $\boldsymbol{F}$. If Δ is rational, then $\boldsymbol{F}$ is an at most denumerable subset of $[1, \propto)$. The ergodicity condition in [11] gives that the Markov chain is ergodic and the system is stable if and only if,

$$E(l/\Delta) < \Delta(1)$$

where $E(l/\Delta)$ denotes the expected length of a CRI given that it starts with an examined interval of length Δ.

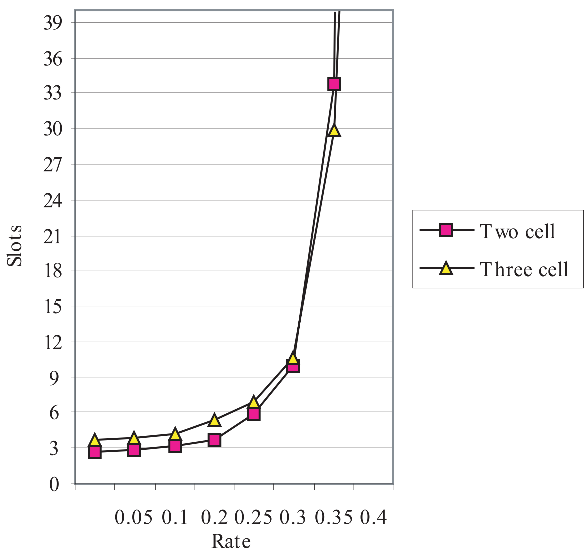


Fig. 1. Expected delays.

Table 1. Throughput and optimal window sizes.

3-cell algorithm	$\lambda^* = 0.4297$	$\Delta^* = 2.5599$
2-cell algorithm	$\lambda^* = 0.4297$	$\Delta^* = 2.330$

Let L_k denote the expected length of a CRI given that it starts with a collision of multiplicity k. We can then write

$$E(l/\Delta) = \sum_{k=0}^{\infty} E(l/\Delta, k) e^{-\lambda\Delta} \frac{(\lambda\Delta)^k}{k!} = \sum_{k=0}^{\infty} L_k e^{-\lambda\Delta} \frac{(\lambda\Delta)^k}{k!} \tag{1}$$

since

$$E(l/\Delta, k) = L_k$$

depends only on k. In the Appendix, and focusing on the 3-cell algorithm, we show that
1) L_k, $k \geq 0$ can be computed recursively, and
2) L_k are quadratically upper-bounded.

Equation (1) together with 1) and 2) are used in the computation of the algorithmic throughput and the optimal window size Δ^* that attains it. Throughput is defined as the maximum Poisson rate λ^* that the algorithm maintains with finite delays. The throughput and the optimal window results for the 3-cell algorithm are included in Table 1, together with those of the 2-cell algorithm in [4]. We note that the methodology exhibited in the Appendix can be used for the throughput evaluation of any algorithm in the class; the complexity of the induced recursive equations increases, however, as the number of cells in the stack which depicts the collision resolution process of the corresponding algorithm increases.

We define the delay D_n experienced by the n-th packet as the time difference between its arrival and the end of its successful transmission. We are interested in evaluating the first moment of the steady state delay process, when it exists. It can be seen that the delay process D_n, $n \geq 1$ is regenerative. The regenerative points are the sequence of consecutive CRPs at which the lag equals one. The method for the delay analysis is given in [12].

Table 2. Throughput as a function of ε and δ.

ε	δ	λ^* 3-cell algorithm		λ^* 2-cell algorithm	
		Case (A)	Case (B)	Case (A)	Case (B)
0	0.2	0.33641	0.3125	0.3463	——
0	0.5	0.190916	0.15625	0.2251	——
0.1	0.1	0.34978	0.33204	0.3706	0.3630
0.1	0.4	0.213361	0.15625	——	——
0.2	0.1	0.318349	0.253916	——	0.328
0.2	0.2	0.273956	0.17578	0.3139	0.279
0.5	0	0.245783	0.019538	0.3250	——
0.5	0.5	0.096538	0.019538	——	——
0.8	0	0.104019	0.019538	0.2280	——

In Fig. 1, we exhibit the expected delays induced by the 3-cell algorithm, together with those induced by the 2-cell algorithm in [4].

Remarks: We note that, as compared to the 2-cell algorithm, the 3-cell algorithm induces somewhat increased delays at low rates at the gain of lower delays at high rates. We also note that, as in [4], we can use the regenerative theory to compute packet inter-departure distributions. Their nature is exponential for low Poisson rates, and approaching mass concentrations at low inter-departure times as the Poisson rates increase.

IV. RESISTANCE TO CHANNEL ERRORS

We wish to study the performance of the 3-cell algorithm in the presence of channel errors, when feedback outputs may be read wrongly with certain probabilities. Specifically, let us assume that due to noisy conditions, the following type of feedback errors may occur. With probability ε, an empty slot may be seen by the users as a collision slot. Also, with probability δ, a slot occupied by a single transmission may be seen by the users as a collision slot. Let us also assume that a collision slot is always recognized correctly by the users. We will consider the cases where, (A) the values ε and δ are known a priori, as system characteristics, and (B) the values ε and δ are unknown to the designer. Then, in Case (A), the window size Δ is optimized for throughput maximization, while in Case (B), the throughput is found for Δ window size as in Table 1. We performed throughput analysis for both Cases (A) and (B), whose details are included in the Appendix. The results are exhibited in Table 2, together with those for the 2-cell algorithm in [4].

From the results in Table 2, we notice that the throughput of the 3-cell algorithm degrades gracefully in the presence of feedback errors, in contrast to the algorithms in [7] and [5] that are then led to deadlocks. As compared to the 2-cell algorithm in [4], the throughput degradation of the 3-cell algorithm in the presence of feedback errors is somewhat higher, due to the finer tuning required by its splitting operation. We expect that such throughput degradation will then increase for algorithms in the class depicted by more than three stack cells.

V. NUMERICAL EVALUATIONS AND COMPARISONS

In this section, we compare the 3-cell algorithm with the 2-cell algorithm in [4] and the Ethernet protocol, when all algo-

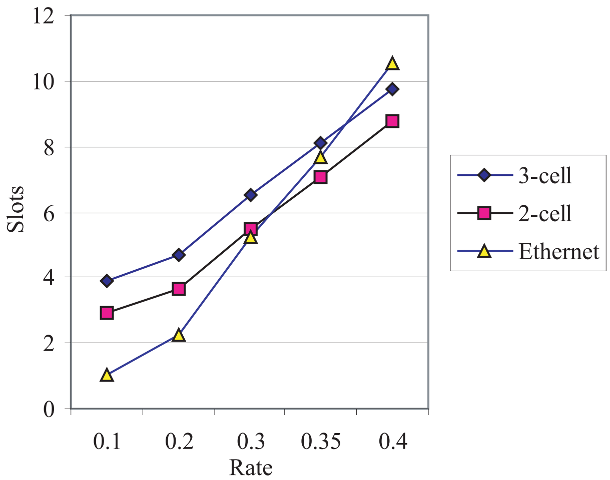


Fig. 2. Expected delays. Admission delay 80 slots.

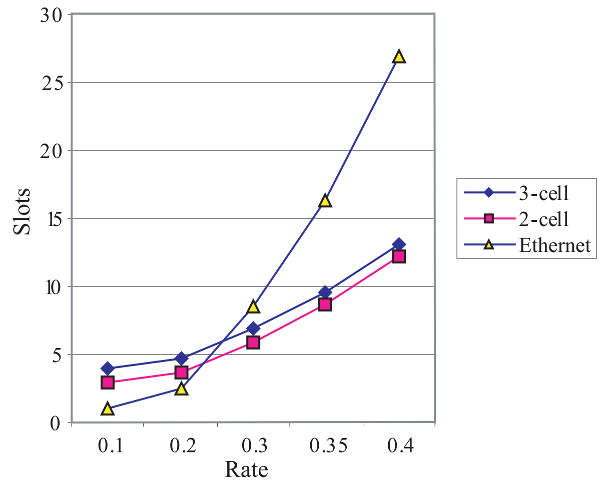


Fig. 4. Expected delays. Admission delay 200 slots.

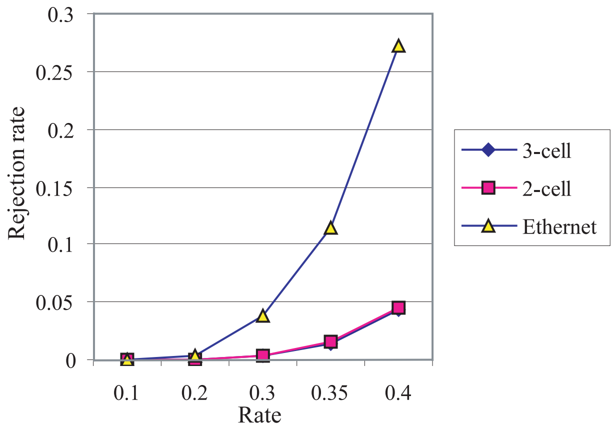


Fig. 3. Rejection rates. Admission delay 80 slots.

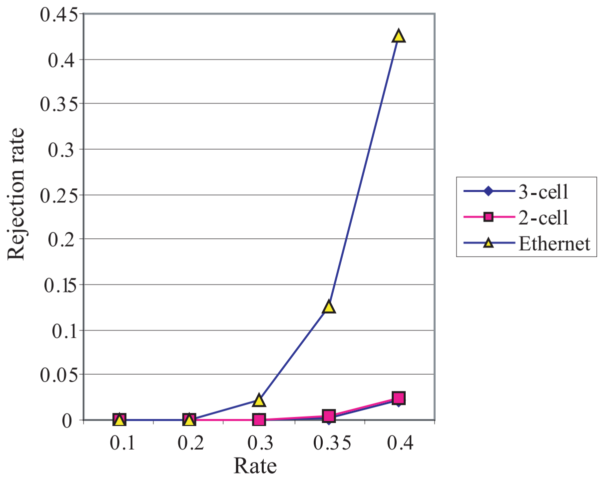


Fig. 5. Rejection rates. Admission delay 200 slots.

rithms operate in environments imposing admission delay constraints. We remind the reader that the Ethernet protocol is a form of the ALOHA protocol with exponential backoff retransmission policy and imposed abortions after sixteen attempts. We simulated all three algorithms in the presence of the limit Poisson user model for admission delay constraints equal to 80, 200, and 400 slots and we computed average delays and packet rejection rates. Our results are exhibited in Figs. 2–7. From the latter figures we observe that, for relatively tight (80 slots) admission delay constraints, the Ethernet protocol does somewhat better than the 2-cell and 3-cell algorithms in terms of expected delays of those packets that are successfully transmitted, while it does significantly worse in terms of rejection rates: For 0.4 input traffic rate, Ethernet rejects then 27% of the traffic versus the less than 5% rejection rate induced by the 2-cell and 3-cell algorithms, while, at the same time, all three algorithms induce similar expected delays. As the admission delay constraint becomes looser, the Ethernet protocol exhibits terrible expected delays and rejection rate behavior, as expected, versus the excellent such behavior of both the 2-cell and the 3-cell algorithms: For 400 slots admission delay and 0.4 input traffic rate, Ethernet rejects 60% of the input traffic and induces expected delay of 68 slots per transmitted packet, while the 2-cell and 3-cell algorithms impose then almost no rejections and induce less than 18 slots expected delays.

VI. CONCLUSIONS

We introduced a class of window limited sensing random access algorithms, whose collision resolution process can be depicted by a stack with finite number of cells. All algorithms in the class are stable and implementable, possess relatively high throughput, are robust in the presence of channel feedback errors and exhibit good delay characteristics. As the latter number increases, the complexity of the equations leading to the throughput evaluation of the corresponding algorithm increases as well. We conjecture that the throughput remains unchanged and equal to 0.4297 for all algorithms in the class. As the number of cells in the stack which depicts the collision resolution process increases, the resistance to some channel feedback errors decreases and the delays to low traffic rates increase, while the delays for high traffic rates decrease. The superiority of the algorithms in the class to the Ethernet protocol has been numer-

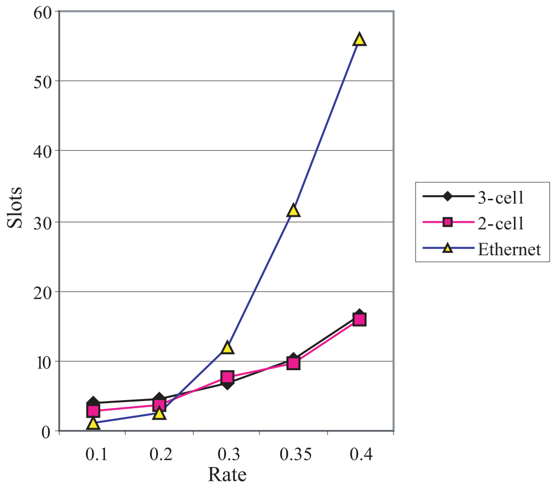


Fig. 6. Expected delays. Admission delay 400 slots.

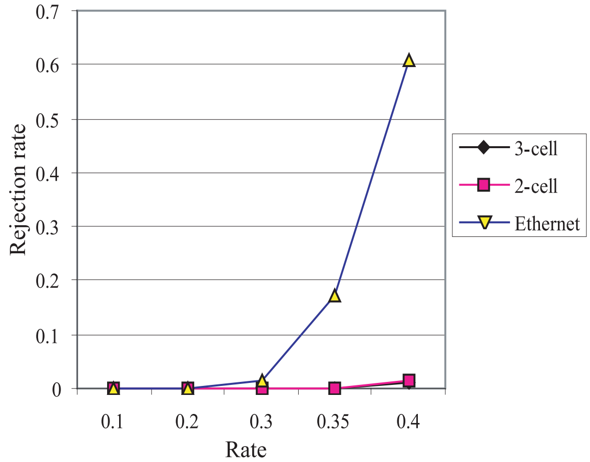


Fig. 7. Rejection rates. Admission delay 400 slots.

ically exhibited.

APPENDIX: THROUGHPUT AND STABILITY ANALYSIS

We will present our analysis in the general case where feedback errors may occur. Setting $\varepsilon = \delta = 0$ in the latter analysis, provides the desirable results in the absence of errors. Let us define:

$L(n,k,m)$: The average length required by the algorithm to transmit $n+k+m$ packets when n of them have counter value 1, k of them have counter value 2 and m of them have counter value 3.

L_k: The average length of a CRI starting with a k multiplicity collision, as induced by the algorithm.

$G(k,m)$: The average length to transmit $k+m$ packets when k of them have counter value 1 and m of them have counter value 2, while it is known that no packets have counter value 3.

Then, we can write the following recursions:

$$L_0 = \begin{cases} 1, & \text{w.p. } (1-\varepsilon) \\ 1 + L(0,0,0), & \text{w.p. } \varepsilon \end{cases}$$

$$L(0,0,0) = \begin{cases} 1 + G(0,0), & \text{w.p. } (1-\varepsilon) \\ 1 + L(0,0,0), & \text{w.p. } \varepsilon \end{cases}$$

$$G(0,0) = \begin{cases} 1 + L_0, & \text{w.p. } (1-\varepsilon) \\ 1 + L(0,0,0), & \text{w.p. } \varepsilon \end{cases}$$

$$L(0,k,m) = \begin{cases} 1 + G(k,m), & \text{w.p. } (1-\varepsilon) \\ 1 + L(0,k,m), & \text{w.p. } \varepsilon \end{cases}$$

$$G(0,m) = \begin{cases} 1 + L_m, & \text{w.p. } (1-\varepsilon) \\ 1 + L(0,m,0), & \text{w.p. } \varepsilon \end{cases}$$

$$G(1,m) = \begin{cases} 1 + L_m, & \text{w.p. } (1-\delta) \\ 1 + L(1,m,0), & \text{w.p. } \frac{\delta}{3} \\ 1 + L(0,m+1,0), & \text{w.p. } \frac{\delta}{3} \\ 1 + L(0,m,1), & \text{w.p. } \frac{\delta}{3} \end{cases}$$

$$k \geq 2, \quad G(k,m) = L(k,m,0)$$
$$n \geq 2, \quad L_n = L(n,0,0) = G(n,0)$$

$$L(1,k,m) = \begin{cases} 1 + G(k,m), & \text{w.p. } (1-\delta) \\ 1 + L(0,k+1,m), & \text{w.p. } \frac{\delta}{3} \\ 1 + L(0,k,m+1), & \text{w.p. } \frac{\delta}{3} \\ 1 + L(1,k,m), & \text{w.p. } \frac{\delta}{3} \end{cases}$$

$$L_1 = \begin{cases} 1, & \text{w.p. } (1-\delta) \\ 1 + L(1,0,0), & \text{w.p. } \frac{\delta}{3} \\ 1 + L(0,1,0), & \text{w.p. } \frac{\delta}{3} \\ 1 + L(0,0,1), & \text{w.p. } \frac{\delta}{3}. \end{cases}$$

Via partial substitutions in the above system, we find

$$\begin{aligned} L_0 &= (1-\varepsilon)^{-3} \\ L(0,0,0) &= (1-\varepsilon)^{-3}[3(1-\varepsilon)+\varepsilon^2] \\ L(0,1,0) &= 3(3-\delta)^{-1}(1-\varepsilon)^{-3}[(2-\varepsilon)(1-\varepsilon)^{-2}+1-\delta] \\ &\quad + \delta(3-\delta)^{-1}L(1,0,0) + \delta(3-\delta)^{-1}L(0,0,1) \end{aligned} \tag{2}$$

$$\begin{aligned} L(0,0,1) &= 3(1-\varepsilon)^{-1}[3-(1-\varepsilon)\delta]^{-1}[2-\varepsilon+(1-\varepsilon)^2] \\ &\quad + \delta(1-\varepsilon)[3-(1-\varepsilon)\delta]^{-1}L(1,0,0) \\ &\quad + 3[3-(1-\varepsilon)\delta]^{-1}[3\varepsilon+(1-\varepsilon)\delta]L(0,1,0) \\ L(1,0,0) &= 3(3-\delta)^{-1}(1-\varepsilon)^{-3}[(1-\varepsilon)^3+(2-\varepsilon)(1-\delta)] \\ &\quad + \delta(3-\delta)^{-1}L(0,1,0) + \delta(3-\delta)^{-1}L(0,0,1) \\ L_1 &= 1 + 3^{-1}\delta L(1,0,0) + 3^{-1}\delta L(0,1,0) \\ &\quad + 3^{-1}\delta L(0,0,1) \end{aligned}$$

$$L_m \triangleq L(m,0,0), \forall m \geq 2 \tag{3}$$

$$L(0,0,m) = (2-\varepsilon)(1-\varepsilon)^{-1} + (1-\varepsilon)L_m + \varepsilon L(0,m,0), \quad m \geq 2 \tag{4}$$

$$\begin{aligned}
L(0,1,m) &= (2-\varepsilon)(1-\varepsilon)^{-1} + (1-\delta)L_m \\
&\quad + 3^{-1}\delta L(1,m,0) + 3^{-1}\delta L(0,m,1) \\
&\quad + 3^{-1}\delta L(0,m+1,0), \qquad m \geq 1 \\
L(0,k,m) &= (1-\varepsilon)^{-1} + L(k,m,0), \quad k \geq 2, m \geq 0 \\
L(1,0,m) &= (3-\delta)^{-1}\{3(2-\delta) + 3(1-\delta)(1-\varepsilon)L_m \\
&\quad + 3\varepsilon(1-\delta)L(0,m,0) + \delta L(0,1,m) \\
&\quad + \delta L(0,0,m)\}, \qquad m \geq 1 \qquad (5) \\
L(1,1,m) &= (3-\delta)^{-1}\{3(2-\delta) + 3(1-\delta)^2 L_m \\
&\quad + \delta(1-\delta)L(1,m.0) + \delta(1-\delta)L(0,m+1,0) \\
&\quad + \delta(1,\delta)L(0,m,1) + \delta L(0,2,m) \\
&\quad + \delta L(0,1,m+1)\}, \qquad m \geq 0 \\
L(1,k,m) &= (3-\delta)^{-1}\{\, 3 + 3(1\delta)L(k,m,0) \\
&\quad + \delta L(0,k+1,m) + \delta L(0,k,m+1)\,\}, \\
&\qquad k \geq 2, m \geq 0.
\end{aligned}$$

In addition, we obtain

$$\begin{aligned}
&L(n-n_2-n_3,n_2,n_3) \\
&= 1 + \frac{n-n_2-n_3}{3^{n-n_2-n_3}} \sum_{m_2=0}^{n-n_2-n_3-1} \binom{n-n_2-n_3-1}{m_2} \\
&\times L\left(1, n_2+m_2, n-n_2-1-m_2\right) + \frac{1}{3^{n-n_2-n_3}} \\
&\times \sum_{m_2=0}^{n-n_2-n_3} \binom{n-n_2-n_3}{m_2} L\left(0, n_2+m_2, n-n_2-m_2\right) \\
&+ \frac{1}{3^{n-n_2-n_3}} \sum_{k=0}^{n-n_2-n_3} \left\{ \binom{n-n_2-n_3}{k} \right. \\
&\qquad \times \sum_{m_2=0}^{n-n_2-n_3-k} \binom{n-n_2-n_3-k}{m_2} \\
&\qquad \left. \times L\left(k, n_2+m_2, n-n_2-k-m_2\right) \right\}, \\
&\qquad\qquad n-n_2-n_3 \geq 2.
\end{aligned} \tag{6}$$

The system of equations in (2) to (6) can be easily shown to have an inductive form, where the system of all possible placements of n packets in 3 cells and its corresponding L's is solved inductively using answers of the parallel quantities corresponding to $n-1$ packets instead. We found the L_k, $k \leq 50$ values solving the above systems of equations. For $k \geq 51$, we used tight upper bounds for the L_k's. These bounds can be shown to be quadratic. That is,

$$k > 50, L_k < \alpha(\varepsilon,\delta)k^2 + \beta(\varepsilon,\delta)k + \gamma(\varepsilon,\delta) \triangleq \quad L_k^u \tag{7}$$

where the coefficients $\alpha(\varepsilon,\delta)$, $\beta(\varepsilon,\delta)$, and $\gamma(\varepsilon,\delta)$ can be found for the different values of the pair (ε,δ). For $\varepsilon = \delta = 0$, the coefficients are $\alpha = 0.01614$, $\beta = 5.53042$, and $\gamma = -29.456$. Using the bounds in (7), we conclude that a sufficient condition for stability is

$$f(x) \triangleq \sum_{k=0}^{50} L_k e^{-x}\frac{x^k}{k!} + \sum_{k=51}^{\infty} L_k^u e^{-x}\frac{x^k}{k!} < \Delta \tag{8}$$

where $x \triangleq \lambda\Delta$.

The condition in (8) can be rewritten as

$$\begin{aligned}
f(x) &\triangleq \alpha(\varepsilon,\delta)x(x+1) + \beta(\varepsilon,\delta)x + \gamma(\varepsilon,\delta) \\
&\quad + \sum_{k=0}^{50}[L_k - \alpha(\varepsilon,\delta)k^2 - \beta(\varepsilon,\delta)k - \gamma(\varepsilon,\delta)]e^{-x}\frac{x^k}{k!} < \Delta
\end{aligned} \tag{9}$$

or as

$$f(x) < \frac{x}{\lambda} \Rightarrow \lambda < \frac{x}{f(x)}. \tag{10}$$

Equation (10) provides the stability region of the algorithm for arbitrary x. When the supremum of the ratio $x/f(x)$ is found, it provides the throughput λ^*, the algorithm and the optimal window size Δ^* which attains it. That is,

$$\begin{aligned}
\lambda^* &= \sup_x \frac{x}{f(x)} \text{ attained at } x^* \\
\Delta^* &= \frac{x^*}{\lambda^*}.
\end{aligned}$$

The function $x/f(x)$ has a unique supremum which was found numerically. The supremum was found to be equal to 0.4297, attained at $x^* = 1.1$, which gave $\Delta^* = 2.5599$.

REFERENCES

[1] J. L. Massey, "Collision resolution algorithms and random access communications," in *Multiuser Communication,* G. Longo Ed., New York, Springer-Verlag.

[2] B. S. Tsybakov and V. A. Mikhailov, "Free synchronous packet access in a broadcast channel with feedback," *Probl. Pered. Inform.,* vol. 14, no. 4, pp. 259–280, 1978.

[3] L. Georgiadis and P. Papantoni-Kazakos, "A 0.487 throughput limited sensing algorithm," *IEEE Trans. Inform. Theory*, vol. IT-33, no. 2, pp. 233–237, Mar. 1987.

[4] M. Paterakis and P. Papantoni-Kazakos, "A simple window random access algorithm with advantageous properties," *IEEE Trans. Inform. Theory*, vol. 35, pp. 1124–1130, Sept. 1989.

[5] J. I. Capetanakis, "Tree algorithms for packet broadcast channel," *IEEE Trans. Inform. Theory*, vol. IT-25, pp. 505–515, Sept. 1979.

[6] B. S. Tsybakov and V. A. Mikhailov, "Ergodicity of a slotted ALOHA system," *Probl. Pered. Inform.,* vol. 15, no. 4, pp. 73–87, Oct.–Dec. 1979.

[7] R. G. Gallager, "Conflict resolution in random access broadcast networks," in *Proc. AFOSR Workshop Commun. Theory Appl.,* Provincetown, MA, Sept. 1978, pp. 74–76.

[8] M. Paterakis, L. Georgiadis, and P. Papantoni-Kazakos, "On the relation between the finite and the infinite population models for a class of RAAs," *IEEE Trans. Commun.,* vol. COM-35, pp. 1239–1250, Nov. 1987.

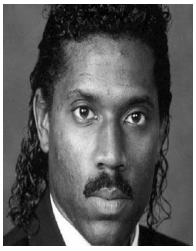
[9] B. S. Tsybakov and N. B. Likanov, "Some new random multiple access algorithms," *Probl. Pered. Inform.*, vol. 21, no. 2, pp. 134–154, 1985.

[10] B. S. Tsybakov and V. B. Fayngold, "A non-homogeneous frame RMA algorithm," *Probl. Pered. Inform.*, vol. 25, no. 2, pp. 126–136, 1989.

[11] S. Stidman, Jr., "Regenerative processes in the theory of queues with applications to the alternating priority queue," *Adv. Appl. Prob.,* vol. 4, pp. 542–577, 1972.

[12] L. Georgiadis, L. Merakos, and P. Papantoni-Kazakos, "A method for the delay analysis of random multiple access algorithms whose delay process

is regenerative," *IEEE J. Select. Areas Commun.,* vol. SAC-5, pp. 1051–1062, July 1987.



Anthony T. Burrell was born in New Orleans, Louisiana. He received the B.A. from the University of California at Santa Barbara, in 1982, the M.S. from West Coast University in 1986 and the Ph.D. in August 1994, from the University of Virginia. In 1982, he joined Unisys Corporation, in 1984, he joined William O'Neil & Company. Concurrently, he joined the faculty at West Coast University. In 1986, he joined the faculty of California State University at Los Angeles as an assistant professor. From 1987–1989, he joined the faculty at Santa Monica College. In 1988, he joined the radar systems group at the Hughes Aircraft Company. In January, 1994, he was a research associate at the University of Ottawa, Canada. From September 1994 to August 1999, he was faculty at the department of electrical & computer engineering at the University of Alabama. In August of 1999, he joined Oklahoma State University where he is now an associate professor. Dr. Burrell's research interests include operating systems, artificial intelligence, and high-speed computer-communication networks, with special emphasis on ATM in the Broadband Integrated Services Digital Network environment. Dr. Burrell is a member of IEEE and ACM.

Titsa P. Papantoni was born in Greece. She received the diploma in Electrical, Mechanical, and Industrial Engineering from the National Technical University of Athens, in 1968; the M.S. degree from Princeton University, Princeton, NJ, in 1970; and the Ph.D. from the University of Southern California, in 1973, both in Electrical Engineering with specialization in Statistical Communications.

From 1973 to 1978, she was a faculty member in Electrical Engineering at Rice University, Houston, TX. From September 1978 to August 1986, she was with the faculty of the Electrical Engineering Department at the University of Connecticut, Storrs, CT. In September 1986, she joined the University of Virginia, where she was professor of Electrical Engineering and Mathematics until December 1994. From September 1993 to August 1994, she was the holder of the NSERC/OCRI Industrial Research Chair in High-Speed Networks at the University of Ottawa, Ottawa, Canada, on leave from the University of Virginia. From January 1995 to August 2005, she held the E. A. "Larry" Drummond Chair in Computer Engineering at the University of Alabama, Tuscaloosa, Alabama. She has been with the University of Colorado since September 2005, first as Chair of the Electrical Engineering department and then as professor. In 1977, she spent one year with Bell Laboratories, in Holmdel, NJ, on leave from Rice University. In 1981, she spent one year on special assignment as a Scientific Officer and Contract Monitor at the U.S. Office of Naval Research, Arlington, VA, on leave from the University of Connecticut.

Dr. Papantoni is Fellow of IEEE, a member of the American Mathematical Society, the Institute of Mathematical Statistics, Eta Kappa Nu, Sigma Xi, the Society of Women in Science, and the Society of Women in Engineering.