Splitting Algorithms in Channels with Markovian Capture

Matan Seri, Moshe Sidi
Department of Electrical Engineering
Technion, Haifa 32000, Israel

Abstract. In this paper we study the performance of tree-like splitting collision resolution algorithms in channels with markovian capture. In particular, we assume that in each slot the channel can be in one of two states -b (for "bad") and g (for "good"). When the channel is in state b, a capture can never occur. When the channel is in state g and n nodes ($n \geq 2$) are transmitting, a capture occurs with probability π_n. The sequence of channel states is assumed to be a homogeneous Markov chain. We derive the throughput of a splitting tree-like multiple access algorithm for this channel. We also provide simulation results for the average delay.

1. INTRODUCTION

In the multiple access systems considered here a time-slotted shared channel is used by many nodes to send packets to a single central receiver. In many studies of these systems it has been assumed that each slot can be either an *idle* slot (no packet is transmitted), or a *success* slot (exactly one packet is transmitted), or a *collision* slot (two or more packets are transmitted and none is correctly received). It has been further assumed that the receiver is able to discriminate between idle, success, and collision slots, and transmit appropriate feedback signals, LACK, ACK, and NACK, respectively. This is known as the ternary feedback model.

Ideally, when each transmission of a node is heard by the receiver and when the forward channel is noiseless, the above feedback signals are always faithful. In practice, however, due to topological and environmental conditions, the receiver is prone to fail to hear some of the packets transmitted in a slot, or it may hear correctly a transmission of a node in presence of other transmissions. The phenomena of detecting a *single* transmission out of many is called *capture*. The reasons for captures in practical systems are that mobile users (nodes) may occasionally be hidden (for example, because of physical obstacles), or have different distances from the receiver, or transmit in different power levels, or because of fading problems. Note that whenever a capture occurs, an ACK is sent with the identity of the node that captured the receiver.

Multiple-access algorithms which handle deterministic captures, in which the nodes of the network are divided into priority groups, have been studied in [2, 9]. Models that considered probabilistic captures were studied in [3, 8]. There it has been assumed that the probability of a capture in a slot may depend on the number of users that transmit during that slot, but is independent from slot to slot, i.e., the capture phenomenon has been assumed to be memoryless. This is of course a simplifying assumption, as the causes for captures can be *with memory*. It is the purpose of this paper to study the performance of tree-like algorithms that operate in channels in which the capture phenomenon is with memory. In particular, we are interested in the effect of different types of memorie, persistent and oscillatory [5], on the performance of a collision resolution algorithm. Persistent memory corresponds to a channel that remains in a state that allows captures to occur for long periods, while oscillatory memory corresponds to a channel that remains in a state that allows captures to occur for short periods.

The paper is structured as follows: in section 2, we describe the channel model and the basic assumptions that we use. In section 3, we describe the tree-based multiple access algorithm. In section 3.A, the algorithm is analyzed and its performance is evaluated in terms of maximal throughput supported. We describe and explain the results in section 3.B for various capture patterns. In section 3.C we provide simulation results for the delay of packets in the system. Finally, in section 4, we summarize the paper.

2. THE MODEL

We consider a communication system that consists of many nodes accessing a common receiver. The forward channel is assumed to be a time-slotted radio channel. In a given slot, each node can transmit at most one packet whose duration is one time slot. The beginning of a transmission is synchronized with the beginning of a time-slot.

During any time-slot one of the following events may occur: a) Idle slot - none of the nodes of the network is transmitting. For an idle slot, the receiver sends a LACK feedback signal that is received by all nodes of the network. b) Success slot - either a single node is transmitting and being received properly or one node out of several transmitting nodes is being properly received by the receiver (*capture*). For a success slot, the receiver sends an ACK (i) feedback signal (i is the identity of the node whose packet is received properly) to all nodes. c) Collision slot - at least two nodes are transmitting and none of them is correctly received by the receiver. For a collision slot the receiver sends a NACK feedback signal to all nodes.

The channel model that we use is similar to [5]. In particular, we assume that in each slot the channel can be in one of two states -b (for "bad") and g (for "good"). When the channel is in state b, a capture can never occur. When the channel is in state g and n nodes ($n \geq 2$) are transmitting, a capture occurs with probability π_n. The state of the channel does not affect idle slots or slots during which a single node is transmitting, i.e., when a single packet is transmitted, it is always received successfully. Let Z_t be the state of the channel at time (slot) t ($t = 0, 1, 2, \ldots$). The sequence of channel states $\{Z_t, t \geq 0\}$ is assumed to be a homogeneous Markov chain with state transition probabilities:

$$P(Z_t = g/Z_{t-1} = g) = r_g \;;\; P(Z_t = b/Z_{t-1} = g) = 1 - r_g \qquad (2.1)$$
$$P(Z_t = g/Z_{t-1} = b) = r_b \;;\; P(Z_t = b/Z_{t-1} = b) = 1 - r_b$$

The probabilities r_g and r_b are the probabilities to occupy the good channel state given that the previous channel state was good or bad, respectively.

One should observe that with this model when node i transmits a packet in a certain slot and is acknowledged by an ACK (j) ($i \neq j$), it is aware of the capture that occurred. Subsequently, such nodes whose packets were not captured by the receiver will be considered to belong to a *lapsed set* until they retransmit their packets again.

3. THE TREE COLLISION RESOLUTION ALGORITHM

If the channel were free of captures, then the collision resolution algorithm (CRA) is as follows [10, 11]. After a collision, all nodes involved flip a binary coin; those flipping 0 retransmit in the very next slot; those flipping 1 retransmit immediately after the collision (if any) among those flipping 0 has been resolved; no new packets may be transmitted until the initial collision is resolved. It is said that a conflict is resolved precisely when all nodes of the system become aware that all initially colliding packets have been successfully retransmitted. In [10] Massey describes a very simple algorithm that can be distributively implemented by the nodes of the system so that each node will know when to transmit and when a CRI ends. The time elapsed from an initial conflict until it is resolved is called a conflict-resolution-interval (CRI). If captures occur no changes in this basic CRA are needed except that all nodes that transmit in a given slot and are not heard by the receiver due to a capture, retransmit at the beginning of the next CRI. This is called the *wait scheme* [3]. Recall that node i learns about its failure to be heard by the receiver due to capture, by examining the feedback indication; if it transmits and the feedback indication is an ACK (j) with $j \neq i$, then node i knows that it has not been heard by the receiver and therefore it joins the lapsed set.

Regarding the first-time transmission rule, namely, which packets are transmitted for the first time at the beginning of a CRI, we adopt the standard idea to "decouple" the transmission times from arrival times [4, 10, 11]. We define an *arrival epoch* of length Δ (measured in slot units) where the i-th arrival epoch is the semi-opened interval $[i\Delta, (i + 1)\Delta]$. The rule that is used is to transmit a new packet that arrived during the i-th arrival epoch in the first utilizable slot following the CRI for new packets that arrived during the $i - 1$ arrival epoch [10]. Here Δ is a fixed length epoch adjusted to maximize the achievable throughput.

Note that in addition to packets that are transmitted for the first time at the beginning of a CRI (according to the above rule) some *residual* packets are also transmitted. The residual packets are those packets, which joined the lapsed set during the previous CRI.

3.A. *Analysis of the algorithm*

Our goal in this section is to determine the maximal output rate (throughput) attainable with the algorithm described above. Similar to [3] and [5] we define X_i to be the total number of packets transmitted at the beginning of the i-th CRI. In addition, let T_i denote the first slot of the i-th CRI, τ_i denote the last slot of the i-th CRI and Z_t denote the state of the channel (b or g) at slot t. When colliding, nodes split into two subsets; p is the probability of being in the first group, and $(1 - p)$ is the probability of being in the second group. Therefore, $Q_n(j) = \binom{n}{j} p^j (1 - p)^{n-j}$ is the probability of j out of n nodes to be in the first subset.

With the above definitions we have $X_i = X_{T_i}$, Z_{T_i} is the state of the channel at the beginning of the i-th CRI and Z_{τ_i} is the state of the channel at the end of the i-th CRI. Let $\omega_n^s(f) \triangleq P(Z_{\tau_i} = f / Z_{T_i} = s, X_i = n)$ denote the probability that the channel state at the end of the i-th CRI is f, given that the CRI started with the channel being at state s and n transmitting nodes.

The probabilities $\omega_n^s(f)$ can be computed recursively for $n \geq 0$ and $s, f \in (b, g)$ as follows. Obviously, for any

$n \geq 0$ and $s \in (b, g)$ we have $\omega_n^s(b) = 1 - \omega_n^s(g)$. For $n = 0, 1$ we have that

$$\omega_0^g(g) = \omega_1^g(g) = 1 \quad ; \quad \omega_0^b(g) = \omega_1^b(g) = 0 \tag{3.1}$$

The reason for (3.1) is that for $n = 0$ and $n = 1$, the first and the last slot of a CRI are the same slot, so they have the same channel state.

For $n \geq 2$ we have,

$$\begin{aligned}\omega_n^g(g) = \pi_n + (1-\pi_n)\sum_{j=0}^{n} Q_n(j)\cdot \\ \Big\{\big[r_g\,\omega_j^g(g) + (1-r_g)\,\omega_j^b(g)\big]\cdot \\ \big[r_g\,\omega_{n-j}^g(g) + (1-r_g)\,\omega_{n-j}^b(g)\big] + \\ \big[r_g\,\omega_j^g(b) + (1-r_g)\,\omega_j^b(b)\big]\cdot \\ \big[r_b\,\omega_{n-j}^g(g) + (1-r_b)\,\omega_{n-j}^b(g)\big]\Big\}\end{aligned} \tag{3.2}$$

$$\begin{aligned}\omega_n^b(g) = \sum_{j=0}^{n} Q_n(j)\Big\{\big[r_b\,\omega_j^g(g) + (1-r_b)\,\omega_j^b(g)\big]\cdot \\ \big[r_g\,\omega_{n-j}^g(g) + (1-r_g)\,\omega_{n-j}^b(g)\big] + \\ \big[r_b\,\omega_j^g(b) + (1-r_b)\,\omega_j^b(b)\big]\cdot \\ \big[r_b\,\omega_{n-j}^g(g) + (1-r_b)\,\omega_{n-j}^b(g)\big]\Big\}\end{aligned} \tag{3.3}$$

The explanation of (3.2) is as follows. If a capture occurs (with probability π_n) the CRI ends immediately. Otherwise, (with probability $1 - \pi_n$) there is a collision and the CRI is splitted into two sub-CRIs (that on their turn split into two sub-CRIs etc.). Then we condition on j out of n packets transmitted at the beginning of the first sub-CRI. The channel state at the beginning of the first sub-CRI is g with probability r_g and then the state of its last slot is g with probability $\omega_j^g(g)$. The channel state at the beginning of the first sub-CRI is b with probability $(1 - r_g)$ and then the state of its last slot is g with probability $\omega_j^b(g)$. Thus, the first sub-CRI ends in channel state g with probability $r_g\,\omega_j^g(g) + (1 - r_g)\,\omega_j^b(g)$. Then, the second sub-CRI begins with channel state g with probability r_g and ends with channel state g with probability $\omega_{n-j}^g(g)$, or begins with channel state b and ends with channel state g with probabilities $(1 - r_g)$ and $\omega_{n-j}^b(g)$, respectively. Thus, the second sub-CRI ends in this case in state g with probability $r_g\omega_{n-j}^b(g) + (1 - r_g)\,\omega_{n-j}^b(g)$. The other term in (3.2) is related to the case that the first sub-CRI ends in channel state b. This occurs with probability $r_g\,\omega_j^g(b) + (1 - r_g)\,\omega_j^b(b)$, and then the second sub-CRI ends in channel state g with probability $r_b\,\omega_{n-j}^g(g) + (1 - r_b)\,\omega_{n-j}^b(g)$. Altogether, the sum in the curly brackets in (3.2) represents the probability of the whole CRI to end with channel state g. The explanation of (3.3) is similar except that the CRI starts in channel state b and a capture is not possible.

Equs. (3.2) and (3.3) are essentially two recursive equations with the variables $\omega_n^g(g)$ and $\omega_n^b(g)$ that can be computed with the initial conditions (3.1).

Continuing with the analysis, we define A_i to be the number of new packets transmitted at the beginning of the i-th CRI, and Y_i is the number of residual packets at the end of the i-th CRI. We have,

$$X_i = A_i + Y_{i-1} \tag{3.4}$$

since at the beginning of a CRI A_i new arrivals join Y_{i-1} residual packets from the previous CRI for transmission. We assume that $\{A_i, i \geq 1\}$ is a sequence of independent and identically distributed (i.i.d.) random variables with a Poisson distribution, so $P(A_i = m) = (\lambda\Delta)^m e^{-\lambda\Delta}/m!$.

Given Y_{i-1} and Z_{T_i}, the random variables Y_i and $Z_{T_{i+1}}$ are independent of Y_j and $Z_{T_{j+1}}$ for $j < i - 1$. In other words, the number of residual packets and the channel state at the end of a CRI depend upon the number of residual packets and the channel state at the beginning of that CRI - so these two random variables contain all the relevant information regarding the probable evolution of the CRI. Consequently, $\{Z_{T_i}, Y_{i-1}, i \geq 1\}$ form a Markov chain. To proceed, we first need to determine the transition probabilities of this chain:

$$P(z, y/z', k) = \\ P\left(Z_{T_{i+1}} = z, Y_i = y / Z_{T_i} = z', Y_{i-1} = k\right) \quad z, z' \in \{b, g\}$$

Note that $Z_{T_{i+1}}$ and Y_i are not statistically independent. However, by conditioning on Z_{τ_i} (the channel state at the last slot of a CRI) we have,

$$\begin{aligned}&P\left(Z_{T_{i+1}} = z, Y_i = y / Z_{T_i} = z', Y_{i-1} = k\right) = \\ &\sum_{\hat{z}} P\left(Z_{T_{i+1}} = z, Y_i = y / Z_{T_i} = z', Y_{i-1} = k, Z_{\tau_i} = \hat{z}\right)\cdot P\left(Z_{\tau_i} = \hat{z} / Z_{T_i} = z', Y_{i-1} = k\right) = \\ &\sum_{\hat{z}} P\left(Z_{T_{i+1}} = z / Z_{\tau_i} = \hat{z}\right)\cdot P\left(Y_i = y / Z_{T_i} = z', Y_{i-1} = k, Z_{\tau_i} = \hat{z}\right)\cdot P\left(Z_{\tau_i} = \hat{z} / Z_{T_i} = z', Y_{i-1} = k\right) = \\ &\sum_{\hat{z}} P\left(Z_{T_{i+1}} = z / Z_{\tau_i} = \hat{z}\right)\cdot P\left(Y_i = y, Z_{\tau_i} = \hat{z} / Z_{T_i} = z', Y_{i-1} = k\right) \qquad z, z', \hat{z} \in \{b, g\}\end{aligned} \tag{3.5}$$

where in the first equality we used Bayes law and in the second equality we used the independence that follows from the fact that the evolution of channel states does not depend on the algorithm.

For a more compact notation let $P_k^{\hat{z}'}(y, \hat{z}) = P\,(Y_i = y, Z_{\tau_i} = \hat{z}/Z_{T_i} = z', Y_{i-1} = k)$. Since τ_i and T_{i+1} are consecutive slots, the four probabilities $P\,(Z_{T_{i+1}} = z/Z_{\tau_i} = \hat{z})$ are defined in (2.1). Substituting in (3.5) the probabilities from (2.1) we obtain for $z = g$:

$$P\,(Z_{T_{i+1}} = g, Y_i = y/Z_{T_i} = z', Y_{i-1} = k) = r_g P_k^{z'}(y, g) + r_b P_k^{z'}(y, b) \tag{3.6}$$

and for $z = b$:

$$P\,(Z_{T_{i+1}} = b, Y_i = y/Z_{T_i} = z', Y_{i-1} = k) = (1 - r_g)\, P_k^{z'}(y, g) + (1 - r_b)\, P_k^{z'}(y, b) \tag{3.7}$$

Using (3.4) we have,

$(Z_{T_i} = z', Y_{i-1} = k)$ and the last slot of the CRI $(Z_{\tau_i} = \hat{z}, Y_i = y)$. With the aid of (3.6) and (3.7) $P_k^{z'}(y, \hat{z})$ is used to compute the probabilities $P(Z_{T_{i+1}} = z, Y_i = y/Z_{T_i} = z', Y_{i-1} = k)$ which are the transition probabilities of the $\{Z_{T_i}, Y_{i-1}, i \geq 1\}$ Markov chain. Following the analysis in [3] and using these transition probabilities from (3.6) and (3.7), the steady-state probabilities $P_Y^z(m) \triangleq P(Z_T = z, Y = m)$, $m \geq 0$ of the chain $\{Z_{T_i}, Y_{i-1}\ i \geq 1\}$ can be computed (assuming that it is ergodic) via

$$\begin{aligned} P_Y^z(m) &= \sum_k \sum_{z'} P\left(Z_{T_{i+1}} = z, Y_i = m, Z_{T_i} = z', Y_{i-1} = k\right) = \\ &\sum_k \sum_{z'} P(m, z/k, z') \cdot P_Y^{z'}(k) = \\ &\sum_{k=0} \left[P(m, z/k, g)\, P_Y^g(k) + P(m, z/k, b)\, P_Y^b(k)\right] \\ &\qquad z \in \{b, g\}, m \geq 0 \end{aligned} \tag{3.10}$$

$$\begin{aligned} P_k^{z'}(y, \hat{z}) &= P\left(Y_i = y, Z_{\tau_i} = \hat{z}/Z_{T_i} = z', Y_{i-1} = k\right) = P\left(Y_i = y, Z_{\tau_i} = \hat{z}/Z_{T_i} = z', X_i - A_k = k\right) = \\ &\sum_{m=0}^{\infty} P\left(Y_i = y, Z_{\tau_i} = \hat{z}/Z_{T_i} = z', X_i = m + k, A_i = m\right) \cdot P\left(A_i = m\right) = \\ &\sum_{m=0}^{\infty} p_{m+k}^{z'}(y, \hat{z})\, P\left(A_i = m\right) \end{aligned} \tag{3.8}$$

where $p_n^{z'}(y, \hat{z}) \triangleq P\,(Y_i = y,\ Z_{\tau_i} = \hat{z}/Z_{T_i} = z', X_i = n)$ is a probability similar to $P_k^{z'}(y, \hat{z})$ defined above, except for the conditioning on X_i, the total number of packets transmitted at the beginning of CRI, instead of conditioning on Y_{i-1}. Using this notation and changing indices we have:

$$\begin{aligned} P_k^{z'}(y, \hat{z}) &= \sum_{m=0}^{\infty} p_{m+k}^{z'}(y, \hat{z}) \cdot P\left(A_i = m\right) = \\ &\sum_{n=k}^{\infty} p_n^{z'}(y, \hat{z}) \cdot P\left(A_i = n - k\right) \end{aligned} \tag{3.9}$$

The equations for computing $p_n^{z'}(y, \hat{z})$ recursively for $n \geq 0$, $0 \leq y < n$ and $z', \hat{z} \in \{b, g\}$ are similar to those of $\omega_n^s(f)$ in (3.1) - (3.3) and they are given in Appendix A.

Having computed $p_n^{z'}(y, \hat{z})$, we use the Poisson probability for A_i and Eq. (3.9) to compute $P_k^{z'}(y, \hat{z})$. Note that in (3.9) as n increases beyond a certain point, $P(A_i = n - k)$ becomes very small and the contribution of additional terms of $p_n^{z'}(y, \hat{z})$ to $P_k^{z'}(y, \hat{z})$ computation is negligible. This consideration also determines how many terms of $p_n^{z'}(y, z)$ are to be computed to achieve adequate precision in $P_k^{z'}(y, \hat{z})$ computation. The probability $P_k^{z'}(y, \hat{z})$ represents the transition in channel state and number of residual packets between the first slot of the i-th CRI

To continue with the analysis, let the average length (in slots) of a CRI that started with m residual packets and with channel state z be denoted by $L_m^z = E\,[l_i/Z_{T_i} = z, Y_{i-1} = m]$. As with the computation of $P_k^{z'}(y, \hat{z})$ and $p_n^{z'}(y, \hat{z})$, we first compute the average length of CRI, which started with $X_i = n$ total number of packets and with channel state z, which is denoted by $l_n^z = E\,[l_i/Z_{T_i} = z, X_i = n]$.

For $n = 0, 1$ we have:

$$l_0^z = l_1^z = 1 \tag{3.11}$$

For $n \geq 2$ and $z = g$:

$$\begin{aligned} l_n^g = 1 + \left(1 - \pi_n\right) \sum_{j=0}^{n} Q_n(j) &\Big\{r_g \Big[l_j^g + \left(\omega_j^g(g)\, r_g + \omega_j^g(b)\, r_b\right) l_{n-j}^g + \\ &\left(\omega_j^g(g)\left(1 - r_g\right) + \omega_j^g(b)\left(1 - r_b\right)\right) l_{n-j}^b\Big] + \\ &\left(1 - r_g\right)\Big[l_j^b + \left(\omega_j^b(g)\, r_g + \omega_j^b(b)\, r_b\right) l_{n-j}^g + \\ &\left(\omega_j^b(g)\left(1 - r_g\right) + \omega_j^b(b)\left(1 - r_b\right)\right) l_{n-j}^b\Big]\Big\} \end{aligned} \tag{3.12}$$

For $n \geq 2$ and $z = b$:

$$l_n^b = 1 + \sum_{j=0}^{n} Q_n(j)$$

$$\Big\{r_b\Big[l_j^g + \left(\omega_j^g(g)r_g + \omega_j^g(b)r_b\right)l_{n-j}^g +$$

$$\left(\omega_j^g(g)\left(1-r_g\right) + \omega_j^g(b)\left(1-r_b\right)\right)l_{n-j}^b\Big] + \qquad (3.13)$$

$$\left(1-r_b\right)\Big[l_j^b + \left(\omega_j^b(g)r_g + \omega_j^b(b)r_b\right)l_{n-j}^g +$$

$$\left(\omega_j^b(g)\left(1-r_g\right) + \omega_j^b(b)\left(1-r_b\right)\right)l_{n-j}^b\Big]\Big\}$$

The explanation of (3.12) - (3.13) is similar to that of (3.2) - (3.3).

Averaging on the number of new arrivals we compute L_m^z,

$$L_m^z = E\left[l_i / Z_{T_i} = z, Y_{i-1} = m\right] =$$

$$E\left[l_i / Z_{T_i} = z, X_i - A_i = m\right] =$$

$$\sum_{k=0}^{\infty} E\left[l_i / Z_{T_i} = z, X_i = m+k, A_i = k\right] \qquad (3.14)$$

$$P\left(A_i = k\right) = \sum_{k=0}^{\infty} l_{m+k}^z\, P\left(A = k\right) =$$

$$\sum_{n=m}^{\infty} l_n^z\, P\left(A = n-m\right) \qquad z \in \{b, g\}$$

where $P\ (A = k)$ is the Poisson probability to have k new arrivals in a time interval of length Δ. With the the stationary probabilities obtained in (3.10) the average length of a CRI is

$$L = E\left[L_m^z\right] = \sum_m \sum_z L_m^z \cdot P\left(Z_T = z, Y = m\right) = \qquad (3.15)$$

$$\sum_m \sum_z L_m^z\, P_Y^z(m) = \sum_{m=0}^{\infty}\left[L_m^g\, P_Y^g(m) + L_m^b\, P_Y^b(m\right.$$

The average number of new arrivals in a Δ-interval is $E\ (A) = \lambda\Delta$. Therefore,

$$T \triangleq \text{throughput of the system} = \frac{\lambda\Delta}{L} \qquad (3.16)$$

Note that a necessary condition for the stability of the system is that the average length of a CRI would be smaller than the arrival epoch Δ (see [11], page 127).

3.B. *Numerical results*

Let $V\ (g)$ and $V\ (b)$ be the invariant probabilities of $[Z_t, t \geq 0]$. We have $V\ (g) = r_b/(1 - r_g + r_b)$ and $V\ (b) = 1 - V\ (g)$. In Fig. 1 we depict the throughput T as a function of r_b that is changed from 0 to 1 while keeping $V\ (g)$ constant. Note that for constant $V\ (g)$, small r_b represents very slow oscillations between the two channel states - which we call a *persistent* type of memory. The other case when r_b is close to 1 represents relatively quick transitions between the two states - which we call an *oscillatory* type of memory. In Fig. 1 we use $\pi_n = 1$, $n = 2, 3$; $\pi_n = 0$, $n > 3$. We note that the throughput is higher when the channel is persistent. Similar behavior has been observed for all other patterns of capture probabilities that were considered. The reason for this is not readily seen and in the following we provide an explanation.

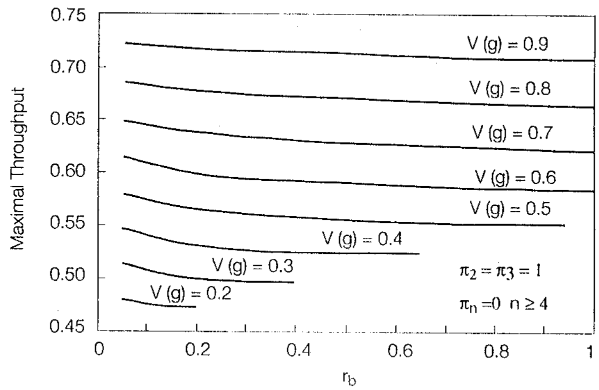

Fig. 1 - Maximal throughput versus r_b for various values of $V\ (g)$.

In the memoryless case $r_b = r_g = V\ (g)$. Holding $V\ (g)$ constant, let us denote this r_b as $r_b\ (ML)$ and the corresponding throughput as T^{ML}. From the numerical computations of the stationary probabilities $P_Y^g\ (m)$ we observe that for the memoryless case $\sum_m P_Y^g\ (m) = V\ (g)$. In other words, the channel states at CRI beginnings are as likely to be *good* as those of any arbitrary slot along the time axis. For the persistent type channel $r_b < r_b\ (ML)$ and $T > T^{ML}$, we observed that $\sum_m P_Y^g\ (m) > V\ (g)$. There are more *good* channel states at CRI beginnings than their occurrence in an arbitrary slot. The opposite is true for the oscillatory type channel in which $r_b > r_b\ (ML)$, $T < T^{ML}$ and $\sum_m P_Y^g\ (m) > V\ (g)$. The reason for this is that the last slot in each CRI is always either empty or a success. This gives the channel state in the last slot of a CRI a preference to be more likely in the *good* state. Then, the channel state in the next slot, which starts a new CRI, is more likely to remain in the same state when the channel is persistent, and is more likely to change its state when the channel is oscillatory. When a CRI begins with a good state, it exploits the capture more efficiently because this slot is the more likely to contain two or more transmitters. This explains the improvement in throughput when the channel is persistent.

3.C. *Average delay - simulation results*

The delay in a multiple access system is defined as the time from the arrival of a packet to the node until its successful transmission. A program that simulates the operation of the algorithm was written in order to obtain its performance in terms of the delay induced on the packets. The average delay in slots is depicted as a function of λ (the arrivals rate) in Fig. 2 for the two types of memory. For both types the same Δ is used which is not necessarily the optimal Δ of both (each type has a different optimal Δ). This gives a common base for the comparison between the two types. For small arrival rates both types of memory induce the same average delay, which is reasonable because no collisions, nor captures occur, and the algorithm is not affected by the channel states. For arrival rates close to the throughput, the delay with the persistent type channel becomes smaller than the oscillatory type channel, which is consistent with the results in the previous section. For moderate arrival rates, the oscillatory type channel shows better performance, because its lapsed set (and the occurance of captures) is smaller than that of the persistent type channel. This reduces the average delay because the first time transmission rule we used in the simulation does not let packets transmit when the current time is less than the time that was resolved (on the arrival time axis) plus Δ.

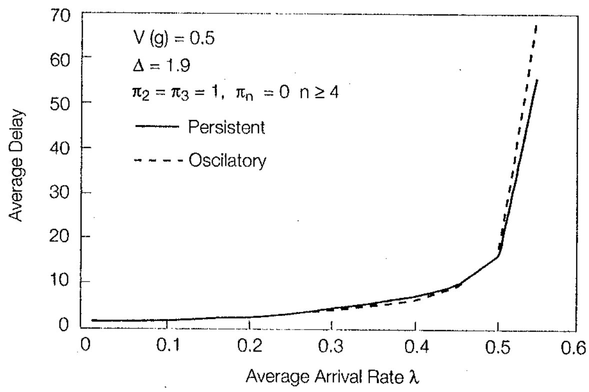

Fig. 2 - Average delay as a function of the arrival rate – λ .

APPENDIX A

Recursive equations for $p_n^{z'}(y,\hat{z})$.

For $n = 0, 1$ we have that

$$p_0^{z'}(y,\hat{z}) = p_1^{z'}(y,\hat{z}) = \begin{cases} 0 & & z' \neq \hat{z} \\ 1 & y = 0 & z' = \hat{z} \\ 0 & y \neq 0 & \end{cases} \tag{A.1}$$

For $n \geq 2$, $0 \leq y \leq n-2$

$$\begin{aligned} p_n^g(y,\hat{z}) = (1-\pi_n)\sum_{j=0}^{n} Q_n(j) \sum_{m=0}^{y} \Big\{ & \Big(r_g\big[p_j^g(m,g)\,r_g + p_j^g(m,b)\,r_b\big] + \\ & (1-r_g)\big[p_j^b(m,g)\,r_g + p_j^b(m,b)\,r_b\big]\Big) p_{n-j}^g(y-m,\hat{z}) + \Big(r_g\big[p_j^g(m,g)(1-r_g) + p_j^g(m,b)(1-r_b)\big] + \\ & (1-r_g)\big[p_j^b(m,g)(1-r_g) + p_j^b(m,b)(1-r_b)\big]\Big) p_{n-j}^b(y-m,\hat{z})\Big\}, \qquad \hat{z} \in \{b,g\} \end{aligned} \tag{A.2}$$

Similarly:

$$\begin{aligned} p_n^b(y,\hat{z}) = \sum_{j=0}^{n} Q_n(j) \sum_{m=0}^{y} \Big\{ & \Big(r_b\big[p_j^g(m,g)\,r_g + p_j^g(m,b)\,r_b\big] + \\ & (1-r_b)\big[p_j^b(m,g)\,r_g + p_j^b(m,b)\,r_b\big]\Big) p_{n-j}^g(y-m,\hat{z}) + \Big(r_b\big[p_j^g(m,g)(1-r_g) + p_j^g(m,b)(1-r_b)\big] + \\ & (1-r_b)\big[p_j^b(m,g)(1-r_g) + p_j^b(m,b)(1-r_b)\big]\Big) p_{n-j}^b(y-m,\hat{z})\Big\} \qquad \hat{z} \in \{b,g\} \end{aligned} \tag{A.3}$$

For $n \geq 2$, $y = n - 1$

$$p_n^g(n-1,g) = \pi_n + (1-\pi_n)\{Q_n(0)\, r_g\left[r_g p_n^g(n-1,g) + (1-r_g)\, p_n^b(n-1,g)\right] + Q_n(0)(1-r_g)\left[r_b\, p_n^g(n-1,g) + (1-r_b)\, p_n^b(n-1,g)\right] + Q_n(n) r_g\left[p_n^g(n-1,g) r_g + p_n^g(n-1,b)\, r_b\right] + Q_n(n)(1-r_g)\left[p_n^b(n-1,g) r_g + p_n^b(n-1,b) r_b\right]\} \tag{A.4}$$

$$p_n^g(n-1,b) = (1-\pi_n)\{Q_n(0)\, r_g\left[r_g p_n^g(n-1,b) + (1-r_g)\, p_n^b(n-1,b)\right] + Q_n(0)(1-r_g)\cdot \left[r_b\, p_n^g(n-1,b) + (1-r_b)\, p_n^b(n-1,b)\right] + Q_n(n) r_g\left[p_n^g(n-1,g)(1-r_g) + p_n^g(n-1,b)(1-r_b)\right] + Q_n(n)(1-r_g)\left[p_n^b(n-1,g)(1-r_g) + p_n^b(n-1,b)(1-r_b)\right]\} \tag{A.5}$$

$$p_n^b(n-1,g) = Q_n(0)\left\{r_b\left[r_g\, p_n^g(n-1,g) + (1-r_g)\, p_n^b(n-1,g)\right] + (1-r_b)\cdot \left[r_b\, p_n^g(n-1,g) + (1-r_b)\, p_n^b(n-1,g)\right]\right\} + Q_n(n)\{r_b\left[p_n^g(n-1,g) r_g + p_n^g(n-1,b) r_b\right] + (1-r_b)\left[p_n^b(n-1,g) r_g + p_n^b(n-1,b) r_b\right]\} \tag{A.6}$$

$$p_n^b(n-1,b) = Q_n(0)\left\{r_b\left[(1-r_g)\, p_n^b(n-1,b) + r_g\, p_n^g(n-1,b)\right] + (1-r_b)\cdot \left[r_b\, p_n^g(n-1,b) + (1-r_b)\, p_n^b(n-1,b)\right]\right\} + Q_n(n)\{r_b\left[p_n^g(n-1,g)(1-r_g) + p_n^g(n-1,b)(1-r_b)\right] + (1-r_b)\left[p_n^b(n-1,g)(1-r_g) + p_n^b(n-1,b)(1-r_b)\right]\} \tag{A.7}$$

Manuscript received on May 15, 1993.

REFERENCES

[1] J. I. Capetanakis: *Tree algorithms for packet broadcast channels.* "IEEE Trans. Inform. Theory", Vol. IT-25, Sept. 1979, p. 505-515.

[2] I. Cidon, M. Sidi: *The effect of capture on collision resolution algorithms.* "IEEE Trans. Commun.", Vol. COM-33, Apr. 1985, p. 317-324.

[3] I. Cidon, H. Kodesh, M. Sidi: *Erasure, capture, and random power level selection in multiple-access systems.* "IEEE Trans. Commun.", Vol. COM-36, No. 3, Mar. 1988, p. 263-271.

[4] R. G. Gallager: *Conflict resolution in random access broadcast networks.* "In Proc. AFOSR Workshop Commun. Theory Appl., Provincetown", Ma.,Sept. 17-20, 1978, p. 74-76.

[5] I. Kessler, M. Sidi: *Splitting algorithms in noisy channels with memory.* "IEEE Trans. Inform. Theory", Vol. IT-35, No. 5, Sept. 1989, p. 1034-1043.

[6] Y. Gong, M. Paterakis: *A robust random multiple-access algorithm for packet transmissions over noisy channels with error memory.* Preprint.

[7] B.S. Tsybakov, V. A. Mikhailov: *Free synchronous packet access in a broadcast channel with feedback.* "Prob. Inform. Trans.", Vol. 14, Oct.-Dec. 1978, p. 259-280.

[8] B.S. Tsybakov, N.D. Vvedenskaya: *Stack algorithms in broadcast channel with capture.* "Prob. Inform. Trans.", Vol. 27, 1991.

[9] M. Sidi, I. Cidon: *Splitting protocols in presence of capture.* "IEEE Trans. Inform. Theory", Vol. IT-31, Mar. 1985, p. 295-301.

[10] J. L. Massey: *Collision resolution algorithms and random access communications.* Univ. California, Los Angeles, Tech. Rep. UCLA-ENG-8016, Apr. 1980; Also in Multi-User Communications Systems (CISM Courses and Lecture Series), G. Longo, Ed. New York: Springer-Verlag, p. 73-137, 1981.

[11] R. Rom, M. Sidi: *Multiple access protocols: performance and analysis.* Springer-Verlag, 1990.

M. Seri, M. Sidi: **Splitting Algorithms in Channels with Markovian Capture.**

ETT, Vol. 5 - No. 1 Jan. - Feb. 1994, p. 19 - 25