2^{nd}-ORDER MARKOVIAN APPROXIMATION ON THE OUTPUT PROCESS OF MULTI-USER RANDOM ACCESS COMMUNICATION NETWORKS

I. Stavrakakis, D. Kazakos
Department of Electrical Engineering, University of Virginia
Charlottesville, VA 22901

Abstract

In this paper the output process of a class of slotted multi-user random-access communication networks is approximated by a 2^{nd}-order Markovian process. The output process is defined as the process of the successfully transmitted packets within the network. The parameters of the approximating process are analytically calculated for a network operating under a specific random access algorithm. The involved methods can also be applied in the calculation of these parameters in the case of any random access algorithm within a class.

The performance of the approximation is measured by considering a star topology of interconnected multi-user random-access communication networks. The mean time that a packet spends in the central node of the above topology is calculated under the proposed approximation on the output processes of the interconnected networks. The results are compared with simulation results from the actual system.

I. Introduction

A lot of work has been done towards the direction of developing communication protocols which determine how a single common resource can be efficiently shared by a large population of users. By now, it is well known that fixed assignment techniques are not appropriate for a system with large population of bursty users. In the latter case, random access protocols are more efficient and many of them have been suggested [1], [2]. In most of the systems, time is divided into slots of length equal to the time needed for a packet transmission (slotted systems).

The deployment of an ever increasing number of multi-user random access communication networks brought up the question of how packets, whose destination is another network, should be handled. Thus, the issue of network interconnection or multi-hop packet transmission, arises, [3], [6], [7]. The basic problem in analyzing interconnected systems is that of characterizing the output process of a multi-user random access communication system; i.e., the departure process of the successfully transmitted packets. The output process of a multi-user random-access communication system depends on the deployed protocol. Description of that process is a difficult task and only approximations based on special assumptions have been attempted, [4]-[7].

Research supported by the Air Force Office of Scientific Research through Grant AFOSR 87-0095.

In a previous work, [8], the problem of the characterization of the output process is addressed. There, a Bernoulli and a first-order Markov approximations are proposed. It turns out that the first-order approximation performs better for moderate and heavy traffic load, with respect to a certain measure of performance which is of practical interest. In this paper, a 2^{nd}-order Markov approximation on the output process of a multi-user random-access communication network is proposed. The 2^{nd}-order Markov approximation is intuitively more pleasing than those suggested in [8], since it captures better the dependencies in the output process. These dependencies are introduced by the collision resolution algorithm which the random-access algorithm deploys.

II. The approximation of the output process

A slotted multi-user random access communication network is considered. It is assumed that the packet input rate to the system is λ packets per slot and that λ is in the stability region of the system. For such values of λ, all the processes associated with the description of the system are stationary. The communication channel can be in one of the following states: I (idle), if no user is using the channel at that time; S (success), if only one user is transmitting; C (collision), if more than one users are transmitting at that time. In the above channel description we have made the assumption that if only one user transmits, then his packet is successfully transmitted. Successfully transmitted packets appear in the output process of the network, while the I and C states of the channel do not result in a packet output from the network. It is assumed that capture events are not present, [18], and that channel errors cannot occur.

We define the output process of a slotted multi-user random-access communication system, $\{a^j\}_{j\geq 0}$, to be a discrete-time binary process associated with the end of the slots. The random variable a^j takes the value 1 if a successful packet transmission occurs in the j^{th} slot, and 0 otherwise. It is clear that the output process can be interpreted as a two-state channel-status process $\{x^j\}_{j\geq 0}$, where $x^j \in \{S,NS\}$; by NS we denote the union of the states I and C. The purpose of this interpretation of the output process is to relate it to the channel-status, which is in interaction with the random-access protocol. The latter is true since the evolution of the channel-status process (which determines completely the output proces) depends on the current (and possibly the past) channel state and the deployed protocol. This happens because the state of the channel is fed back to the users, who determine their action based on this feedback and the protocol. A type of feedback information is required by any stable random-access algorithm.

From the above discussion, we conclude that the output process, $\{a^j\}_{j\geq 0}$, and the two-state channel-status process, $\{x^j\}_{j\geq 0}$, are identical. The problem of characterizing the output process of a multi-user random-access communication network is identical to that of characterizing the channel-status process, $\{x^j\}_{j\geq 0}$. The channel-status process $\{x^j\}_{j\geq 0}$ is controlled by the deployed random-access algorithm. In this paper, we approximate this process by a 2^{nd}-order Markov process, $\{\tilde{x}_j\}_{j\geq 0}$ which has the same state space as $\{x^j\}_{j\geq 0}$ and is ergodic within the stability region of the random access algorithm. To study the process $\{\tilde{x}^j\}_{j\geq 0}$, we can equivalently study the underlying first-order Markov process $\{y^j\}_{j\geq 0}$ defined as follows: $\{y^j\}_{j\geq 0}$ is a discrete time first-order Markov process associated with the end of the slots. The state space at this process is $Z = \{(S,S) = a, (S,NS) = b, (NS,S) = c, (NS,NS) = d\}$; the first part of each pair corresponds to the state of the approximating process $\{\tilde{x}^j\}_{j\geq 0}$ at the end of the $(j-1)^{st}$ slot; the second part corresponds to the state of the process $\{\tilde{x}^j\}_{j\geq 0}$ at the end of the j^{th} slot. Having defined the underlying first-order Markov chain $\{y^j\}_{j\geq 0}$, we can obtain the binary process $\{\tilde{a}^j\}_{j\geq 0}$ with state space $\{0,1\}$ from the stationary function $\tilde{a}: S \rightarrow \{0,1\}$; where

$$\tilde{a}(y^j) = \begin{cases} 1 \text{ if } y=a \text{ or } c \\ 0 \text{ if } y=b \text{ or } d \end{cases} \tag{1}$$

The process $\{\tilde{a}^j\}_{j\geq 0}$ approximates the output process of the random-access communication network. It depends on the first-order Markov process $\{y^j\}_{j\geq 0}$, which also describes the 2^{nd}-order process $\{\tilde{x}^j\}_{j\geq 0}$; the latter approximates the channel-status process. To completely determine the underlying first-order Markov process $\{y^j\}_{j\geq 0}$, we need to estimate its steady-state and transition probabilities. Then the approximating process $\{\tilde{a}^j\}_{j\geq 0}$ is completely determined by (1).

III. Parameters of the Markov process

Since the first-order Markov process, $\{y^j\}_{j\geq 0}$ is only an approximation, it seems natural to estimate its steady state and transition probabilities, by calculating the steady state probabilities that a particular state or state transition occurs in the true process, under stable network operation. This calculation is not always straightforward. The procedure to be followed depends on the class of the deployed random access algorithms.

The possible state transitions of the first-order Markov chain $\{y^j\}_{j\geq 0}$ are shown in Fig. 1. Notice that not all state transitions are possible. Since $\{y^j\}_{j\geq 0}$ is a Markov chain, the following equation must hold:

$$\pi P = \pi \tag{2}$$

where $\pi = (\pi(a), \pi(b), \pi(c), \pi(d))$ is the vector of the steady state probabilities and

$$P = \begin{bmatrix} p(a,a) & p(a,b) & p(a,c) & p(a,d) \\ p(b,a) & p(b,b) & p(b,c) & p(b,d) \\ p(c,a) & p(c,b) & p(c,c) & p(c,d) \\ p(d,a) & p(d,b) & p(d,c) & p(d,d) \end{bmatrix}$$

is the matrix with the transition probabilities of the Markov chain $\{y^j\}_{j\geq 0}$. From Fig. 1, it is easily concluded that the following equations hold.

$$p(a,a) + p(a,b) = 1 \qquad p(b,c) + p(b,d) = 1 \tag{3a}$$

$$p(c,a) + p(c,b) = 1 \qquad p(d,d) + p(d,c) = 1 \tag{3b}$$

$$p(a,c) = p(a,d) = p(b,a) = p(b,b) = 0 \tag{4a}$$

$$p(c,c) = p(c,d) = p(d,a) = p(d,b) = 0 \tag{4b}$$

Assuming that the steady state probabilities and the transition probabilities p(b,c) and p(a,a) are all known, the remaining transition probabilities can be calculated from equations (2), (3) and (4). If one of the steady state probabilities, e.g. $\pi(c)$, is known, then the rest of them can be calculated from the following equations which relate marginal with joint probabilities.

$$\begin{aligned} P(NS,NS) &= P(NS) - P(NS,S) \rightarrow \pi(d) = 1-\lambda - \pi(c) \\ P(S,S) &= P(S) - P(NS,S) \rightarrow \pi(a) = \lambda - \pi(c) \\ P(S,NS) &= P(S) - P(S,S) \rightarrow \pi(b) = \pi(c) \end{aligned} \tag{5}$$

P(S) denotes the probability that a slot is successful and P(NS) is its complement. Under stability, $P(S) = \lambda$.

To summarize, all parameters of the Markov process $\{y^j\}_{j\geq 0}$ can be calculated from the probabilities $\pi(c)$, p(b,c) and p(a,a). The rest of this section is devoted to the calculation of those probabilities. The procedure to be followed depends on the deployed random-access algorithm within the network.

We consider multi-user random-access slotted communication networks in which a binary-feedback, (collision/non-collision, C/NC), limited-sensing collision-resolution algorithm is deployed. The network traffic is assumed to be Poisson with intensity λ packets per slot. The considered algorithm has been developed and analyzed in [11], [10] and [12]. The characterization of the process of the successfully transmitted packets, i.e, the output process of the network, is an open problem.

A brief description of the collision-resolution algorithm is provided at this point. Each user is assigned a counter whose initial value is zero (no packet to be transmitted). This counter is updated according to the steps of the algorithm and the feedback from the channel. Upon packet arrival, the counter value increases to one. Users whose values are equal to one at the beginning of a slot, transmit in that slot. If the channel feedback is collision (C), each counters whose value is greater than one increases it by one; each counter whose value equals one, maintain this value with probability p (splitting probability) or increases it to two with probability 1-p. If the channel feedback is non-collision (NC), all non-zero counters decrease their values by one. A detailed description of the algorithm can be found in [10], [11].

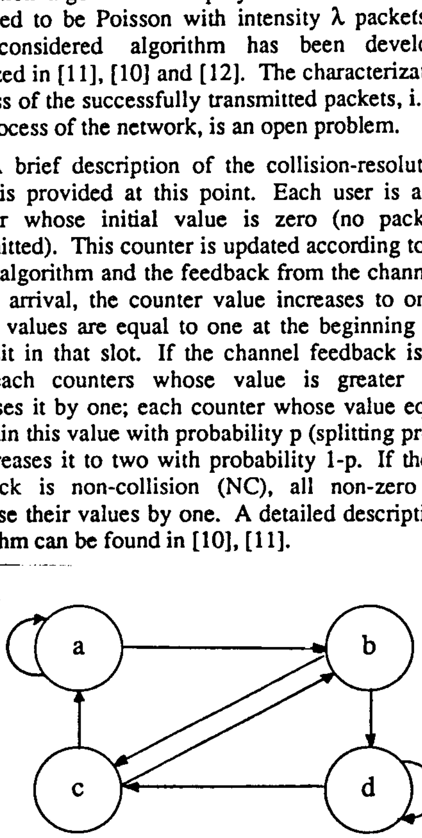

Figure 1.

State transition diagram of the Markov chain $\{y^j\}_{j\geq 0}$.

An important quantity for the analysis of random-access algorithms which induce regenerative points, is the session. A session is defined as the time interval between two renewal points in the operation of the system, [11, [14]. The length of such sessions is easy to describe via recursive equations. The multiplicity of a session is defined as the number of packet transmission attempts in the first slot of the session.

At this point we calculate the probabilities $\pi(c)$, $p(b,c)$ and $p(a,a)$ of the Markov process $\{y^j\}_{j\geq 0}$ for the previously described random-access algorithm. The procedure to be followed can be applied to any limited-sensing stack-type random-access algorithm, [13], [14]. The authors believe that the method can also be applied in the case of other limited-sensing or continuous-sensing random-access algorithms, [15], [16], [17], which operate in statistically identical cycles of finite length (under stability). The following quantities are useful in the analysis that is presented in this section.

(NS,S) pair: A pair of consecutive slots with the first slot being in state NS and the second in state S.

internal (NS,S) pair: An (NS,S) pair is internal if both slots belong to the same session.

(S,NS,S) triplet: A triplet of consecutive slots that are in states S, NS, S.

(S, S, S) triplet: A triplet of consecutive slots that are in states S, S, S.

internal (x,y,z) triplet: A (x,y,z) triplet whose all three slots belong to the same session

l_k: Length of a session of multiplicity k (in slots).

L_k: Expected value of l_k.

L: Expected value of L_k with respect to k.

$\tau_k^{NS,S}$: Number of internal (NS,S) pairs in a session of multiplicity k.

$T_k^{NS,S}$: Expected value of $\tau_k^{NS,S}$.

$T^{NS,S}$: Expected value of $T_k^{NS,S}$ with respect to k.

l_k^{NS}: A random variable associated with the last slot of a session of multiplicity k; $l_k^{NS} = 1$ if that slot is idle; $l_k^{NS} = 0$ if that slot is involved in a successful transmission.

L_k^{NS}: Expected value of l_k^{NS}.

L^{NS}: Expected value of L_k^{NS} with respect to k.

$\tau_k^{S,NS,S}$ ($\tau_k^{S,S,S}$): Number of internal (S,NS,S) triplets (internal (S,S,S) triplets) in a session of multiplicity k.

$T_k^{S,NS,S}$ ($T_k^{S,S,S}$): Expected value of $\tau_k^{S,NS,S}$ ($\tau_k^{S,S,S}$)

$T^{S,NS,S}$ ($T^{S,S,S}$): Expected value of $T_k^{S,NS,S}$ ($T_k^{S,S,S}$)

$l_k^{S,NS}$ ($l_k^{S,S}$): A random variable associated with the last two slots of a session of multiplicity k; $l_k^{S,NS} = 1$ if the last pair of slots of that session is (S,NS); $l_k^{S,NS} = 0$, otherwise ($l_k^{S,S} = 1$ if the last pair of slots of that session is (S,S); $l_k^{S,S} = 0$, otherwise).

l_k^S: A random variable associated with the last slot of a session; l_k^S=1 if the last slot of that session is S; $l_k^s = 0$ otherwise.

$i^{NS,S}$: A random variable associated with the first two slots of a session of multiplicity k; $i^{NS,S} = 1$ if the first pair of slots of that session is (NS,S); $i^{NS,S} = 0$ otherwise.

$L_k^{S,NS}$ (L_k^S, $I_k^{NS,S}$, $L_k^{S,S}$): Expected value of $l_k^{S,NS}$ (l_k^S, $i_k^{NS,S}$, $l_k^{S,S}$).

$L^{S,NS}$ (L^S, $I^{NS,S}$, $L^{S,S}$): Expected value of $L_k^{NS,S}$ (L_k^S, $I_k^{NS,S}$, $L_k^{S,S}$)

An important quantity for the calculation of the desired probabilities is the mean session length, L. The latter can be calculated by following procedures similar to those that appear in [11], [13], [14], [15]. In fact, for the specific algorithm under consideration, L has been calculated in [10] and [11]. We believe that the recursive equations with respect to l_k which describe the operation of the system will be very helpful for the better understanding of the procedure for the calculation of P(NS,S). For this reason we start by calculating L.

From the description of the algorithm the following equations can be written, with respect to l_k, k=1,2,... .

$$l_0 = 1 \ , \ l_1 = 1 \tag{6a}$$

$$l_k = 1 + l_{\phi_1 + f_1} + l_{k-\phi_1 + f_2} \ , \ k\geq 2. \tag{6b}$$

f_1 and f_2 come from two independent Poisson random variables over T=1 (length of a slot) with probability function $P_f(\,)$ and intensity λ; ϕ_1 comes from a Binomial with parameters k and p (p = .5) and probability function $b_k(\,)$. Equation (6b) can be explained as follows: The length of a session of multiplicity k≥2 consists of the slot wasted in the collision, plus the length of the sub-session of multiplicity $\phi_1 + f_1$ (which will be initiated in the next slot), plus the length of the sub-session of multiplicity $k-\phi_1+f_2$, (which will be initiated after the end of the sub-session of multiplicity $\phi_1 + f_1$). Sub-sessions are statistically identical to the sessions of the same multiplicity. ϕ_1 is the number of users whose counter content remained one after the splitting; f_1, f_2 is the number of new users which will be activated (have a packet for transmission) and enter the system in the first slot of the corresponding sub-session.

By considering the expected values in the previous equations with respect to all random variables involved, we obtain an infinite dimensional linear system of equations of the form

$$L_k = h_k + \sum_{j=0}^{\infty} a_{kj} L_j \ , \ k\geq 0 \tag{7}$$

The most widely used definition of stability is the one which relates it with the finiteness of L_k, for $k<\infty$. In [10], [11] it has been found that the system is stable for Poisson input rates $\lambda < S_{max} = .36$ (packets/packet length). The authors in [10], [11] were actually able to find a (linear) upper bound on L_k which is finite for $k<\infty$. S_{max} is then defined as the supremum over all rates λ for which such a bound, L_k^u, was possible to obtain.

The existence of $L_k^u<\infty$, for $k<\infty$, implies that (7) has a non-negative solution, L_k; the solution $\bar{L}_k$ of the finite dimensional system of equations

$$\bar{L}_k = h_k + \sum_{j=0}^{J} a_{kj} \bar{L}_j \ , \ 0\leq k\leq J, \tag{8}$$

is a lower bound on L_k and $\bar{L}_k \rightarrow L_k$ as $J\rightarrow\infty$, [11], [14], [15].

It turns out that for sufficiently large J (e.g. 15), $\tilde{L}_k$ is extremely close to L_k; thus, for all practical purposes, L_k is considered to be equal to L_k, especially for λ outside the neighborhood of S_{max}. The latter can be shown by calculating a tight upper bound on L_k and observing that it almost coincides with $\tilde{L}_k$ (see [11], [14], [15] for the procedure). By solving (8), we calculate the mean session length of multiplicity k. Since the multiplicities of successive sessions are independent and identically distributed random variables, the mean session length, L, is calculated by averaging L_k over all k; k is the number of arrivals in a slot from a Poisson process with intensity λ. In fact, the average for $k \le J$ is sufficient.

To calculate $T^{NS,S}$, $T^{S,NS,S}$, $T^{S,S,S}$, L^{NS}, L^S, $L^{S,NS}$ and $I^{NS,S}$, we follow a procedure similar to that in the calculation of the mean session length L; i.e, by writing recursive equations with respect to the corresponding random variables and by considering the resulting systems of linear equations. By solving truncated versions of those infinite dimensional systems of linear equations, we compute the above quantities. Actually, as it is the case with the calculation of L, what is calculated is a lower bound on the corresponding quantities. As the number of equations considered increases, the bound converges to the true value.

It can be shown, [22], that the steady-state probability $\pi(c) = P(NS,S)$ and the joint probability of having a triplet (S, NS, S) and a triplet (S, S, S) in the output process are given by the expressions

$$L \qquad + \lambda e^{-\lambda} \frac{L^{NS}}{L} \qquad (9a)$$

$$P(S,NS,S) = \frac{T^{S,NS,S}}{L} + \frac{L^{S,NS}}{L} \lambda e^{-\lambda} + \frac{L^S}{L} I^{NS,S} + \frac{L^S}{L} e^{-\lambda} \lambda e^{-\lambda} \qquad (9b)$$

$$P(S,S,S) = \frac{T^{S,S,S}}{L} + \frac{L^{S,S}}{L} \lambda e^{-\lambda} + \frac{L^S}{L} \lambda e^{-\lambda} \lambda e^{-\lambda} \qquad (9c)$$

The steady-state probabilities are calculated from (9a) and (5). The transition probabilities p(b,c) and p(a,a) can be computed from the equations

$$p(b,c) = \frac{P(S,NS,S)}{P(S,NS)} = \frac{P(S,NS,S)}{\pi(b)} \qquad (10a)$$

$$p(a,a) = \frac{P(S,S,S)}{P(S,S)} = \frac{P(S,S,S)}{\pi(a)} \qquad (10b)$$

The rest of the transition probabilities are calculated from equations (2), (3) and (4).

IV. Performance of the approximation

Perhaps, the most interesting application, for which the characterization of the output process of a multi-user random-access communication network is of great importance, is that of analyzing the performance of systems of interconnected multi-user random access communication networks. In such systems, one can find star topologies of interconnected networks. There, the mean time that a packet spends in the central node is an important performance measure of the interconnection; it is thus desired that this quantity be calculated. This is the reason for the selection of the previous mean time as a performance measure for the proposed approximation. The value of the mean time is not by itself a performance measure for the approximation. It is the comparison of this quantity, calculated under the approximation, with the one from the simulation of the actual system that indicates how good the approximation is.

A star topology of N interconnected networks is shown in Fig. 2. Each input stream represents the output process from a multi-user random-access slotted communication system. Let λ_i be the output rate (in packets per slot) of the i^{th} network. A packet arrival in the central node is declared at the end of the slot in which the packet was successfully transmitted. Thus, the arrival process associated with each input line is a discrete process. The arrival points in all streams coincide; that is, the networks are assumed to be synchronized and all slots are of the same length.

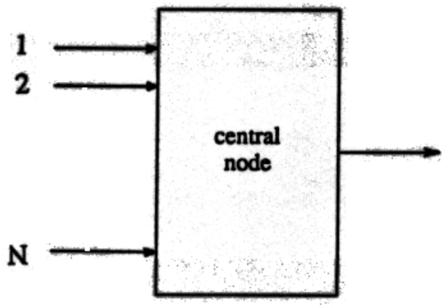

Figure 2

A star topology of interconnected networks.

The service time in the central node is constant and equal to one, which is assumed to be the length of a slot. This implies that arriving and departing packets have the same length. The first in-first out (FIFO) service policy is adopted. More than one arrivals (from different input streams) that occur at the same arrival point are served in a randomly chosen order. The buffer capacity of the central node is assumed to be infinite.

A discrete time single server queueing system with finite number of independent input streams and per stream arrivals governed by an underlying finite-state Markov chain, has been analyzed in [9]. The system that is considered in this section is a special case of the general system in [9]. If λ_i is within the stability region of the corresponding network and if, [18],

$$\sum_{i=1}^{N} \lambda_i < 1,$$

then the queueing system is stable. The average number of packets, Q, in the central node can be calculated as the sum of the solutions of 4^N linear equations, [9]. Then, the mean time that a packet spends in the central node, D, is given in conjunction with Little's formula by the following expression.

$$D = \frac{Q}{\sum_{i=1}^{N} \lambda_i} \qquad (11)$$

Note that under stable operation of the networks the adopted mapping rule in (1) implies that the input rate of the i^{th} stream to the central node is equal to the input rate to the corresponding network.

Let us denote by $\bar{x}$ and $\bar{y}$ the N-dimensional vectors that describe the states of the N Markov chains in two consecutive time slots, $\bar{x},\bar{y}\varepsilon\bar{Z}=Z^1 \times Z^2 \times \cdots \times Z^N$; $p(\bar{x},\bar{y})$ denotes the transition probability from state $\bar{x}$ to state $\bar{y}$. Let $p(j;\bar{y})$ denote the probability that there are j packets in the central node and that the N-dimensional Markov chain is in state $\bar{y}$, and let $P(z;\bar{y})$ be the corresponding generating function. Then, the average number of packets in the system, Q, is given by the sum of the solutions of 4^N linear equations, [9]. These equations are given by

$$\sum_{v=0}^{N} \sum_{\bar{x}\varepsilon F_v} [2(v-1)P'(1;\bar{x}) + (v-1)(v-2)P(1;\bar{x})$$

$$+ 2(v-1)p(0;\bar{x})]=0 \qquad (12a)$$

and any 4^N-1 from the following:

$$P'(1;\bar{y}) = \sum_{v=0}^{N} \sum_{\bar{x}\varepsilon F_v} [(v-1)P(1;\bar{x})+P'(1;\bar{x})+p(0;\bar{x})]p(\bar{x},\bar{y})$$

$$+p(0;\bar{x})]p(\bar{x},\bar{y}) \ , \ \bar{y}\varepsilon\bar{S} \qquad (12b)$$

The unknown quantities in (12) are $P'(1;\bar{y})$, $\bar{y}\varepsilon\bar{S}$; $P'(1;\bar{y})$ denotes the value of the derivative of $P(z;\bar{y})$ at z=1. The set F_v is given by

$$F_v = \left\{ \bar{x} = (x_1, \cdots x_N) \,\varepsilon\, \bar{Z} \ : \ \sum_{i=1}^{N} a_i(x_i) = v \right\}.$$

where $a_i(\,)$ is the mapping associated with the i^{th} network. Since the input streams to the central node are independent, we have that

$$p(\bar{x},\bar{y}) = \prod_{i=1}^{N} p_i(x_i,y_i) \ , \ p(0;\bar{x}) = p_0 \prod_{i=1}^{N} \pi_i(x_i)$$

$$P(1;x) = \pi(\bar{x}) = \prod_{i=1}^{N} \pi_i(x_i)$$

where $\pi_i(x_i)$ and $p_i(x_i, y_i)$ are the steady state and state transition probabilities of the Markov chain associated with the i^{th} input stream; p_0 is the probability that the central node is empty. The latter is given by, [9],

$$p_0 = 1 - \sum_{i=1}^{N} \lambda_i.$$

By solving the 4^N linear equations that are given by (12) and summing up the solutions, the average number of packets in the central node, Q, is obtained. Then, the mean time that a packet spends in the system is calculated from (11).

V. Results and conclusions

In this section, the performance of the proposed approximation model of the output process is compared with the performance of the actual system. The mean time that a packet spends in a central node which receives and retransmits packets originating from N=3 slotted multi-user random access networks, is used as the performance measure. It is assumed that the random-access algorithm described in section III is deployed within each of the networks.

The mean time that a packet spends in the central node of the star topology was calculated from the expressions given in the previous section. The results (in slots) are shown in Table 1, together with the results obtained from the simulation of the actual system. The maximum per network output rate under stable operation of the particular algorithm is .36 packets per slot. On the other hand, the queueing system of the star topology is stable for total input rates less than 1.00 packets per slot, [18].

By comparing the analytical results, obtained under the approximation of the output process by a 2^{nd}-order Markov chain, with the simulations we conclude that the approximation performs well for the whole range of per network input rates. The case of N=2 is not of practical interest, since the resulting queueing problem is not severe ($\lambda_1+\lambda_2<.72$). The proposed approximation seems to perform better than the Bernoulli or the first-order Markov approximations discussed in [8], under heavy traffic. Under heavy traffic the dependencies introduced by the random-access algorithms are strong and it seems that they are best captured by the proposed approximation. Of course, the performance of the proposed approximation depends on the random-access algorithm deployed within the network. In this paper, we developed the approximating model and the methods to compute its parameters for a class of random-access algorithms. Results are presented only for a special case and the conclusions can be extended to other cases only intuitively.

λ	π(c)	P(a,a)	P(b,c)	Markov	Sim.
.01	.0098	.0136	.0100	1.01	1.00
.10	.0862	.1271	.0999	1.15	1.01
.20	.1488	.2381	.2006	1.58	1.21
.25	.1728	.2896	.2520	2.19	1.70
.30	.1922	.3403	.3050	4.69	4.28
.31	.1955	.3506	.3160	6.49	6.25
.32	.1985	.3609	.3271	10.99	11.37
.33	.2014	.3714	.3385	42.55	48.89

Table I.

Results for the mean packet delay in the central node of a star topology of 3 interconnected networks; λ is the per network input (output) rate. The results are under the Markov approximation and from the simulation of the actual system.

For N≥4, the dimensionality of the system of linear equations which need to be solved, (12), increases rapidly. For such systems, simulation results have shown that the Bernoulli approximation on the output process performs well; its performance improves as N increases. The latter can be explained by the fact that the increased number of independent input streams reduces the dependencies in the total input traffic to the central node. The per network output traffic must also decrease, for the queueing system to be stable. The latter implies that either each network operates away from its stability region and thus the dependencies in its output process are not strong, or that not all successful packets are forwarded to the central node; the packet selection introduced in the latter case results in increased independence in the output process.

References

1. F. A. Tobagi, "Multiaccess Protocols in Packet Communication systems", IEEE Transactions on Communications, Vol COM-28, No. 4, April 1980.
2. B. S. Tsybakov, "Survey of USSR Contributions to Random Multiple-access Communications", IEEE Transactions on Information Theory, Vol. IT-31, No. 2, March 1985.
3. L. Merakos, G. Exley, C. Bisdikian, "Interconnection of CSMA Local Area Networks: The Frequency Division Approach", IEEE Transactions on Communications, Vol. COM-35, No. 7, July 1987..
4. F. A. Tobagi, "Packet Delay and Interdeparture Time", Journal of the Association for Computing Machinery, Vol. 23, No. 4, Oct. 1982.
5. H. Takagi, L. Kleinrock, "Output Processes in Contention Packet Broadcasting Systems", IEEE Transactions on Communications, Vol. COM-33, No. 11, Nov. 1985.
6. F. A. Tobagi, "Analysis of a Two-Hop Centralized Packet Radio Network-Part I: Slotted ALOHA", IEEE Transactions on Communications, Vol. COM-28, No. 2, Feb. 1980.
7. F. A. Tobagi, "Analysis of a Two-Hop Packet Radio Network-Part II: Carrier Sense Multiple Access", IEEE Transactions on Communications, Vol. COM-28, No. 2, Feb. 1980.
8. I. Stavrakakis, D. Kazakos, "On the Approximation of the Output Process of Multi-User Random-Access Communication Networks", IEEE INFOCOM'88, March 28-31, 1988, N. Orleans (to appear).
9. I. Stavrakakis, D. Kazakos, "A Queueing System with Independent Markov Input Streams", Technical Report, University of Virginia, UVA/525677/EE87/101, April 1987 (also submitted for publication).
10. P. Mathys, "Analysis of Random Access Algorithms", Ph.D. dissertation, Diss. ETH No. 7713, Swiss Federal Institute of Technology (ETH), Zurich, 1984.
11. N. D. Vvedenskaya, B. S. Tsybakov, "Random Multiple Access of Packets to a Channel with Errors", pp. 131-147 in Plenum Press translation of article in Problemy Peredachi Informatsii, Vol. 15, No. 2, pp. 52-68, Apr.-Jun. 1983.
12. G. Fayolle, P. Flajolet, M. Hofri, P. Jacket, "Analysis of a Stack Algorithm for Random Multiple-Access Communication", IEEE Transactions on Information Theory, Vol. IT-31, No. 2, pp. 244-254, 1985.
13. B.S. Tsybakov, N.D. Vvedenskaya, "Random Multiple Access Stack Algorithm", Problems of Information Transmition, Vol. 16, No. 3, pp. 230-243 translated from Problemy Peredachi Informatsii, Vol. 16, No. 3, pp. 80-94, Jul.-Sept., 1980.
14. I. Stavrakakis, D. Kazakos, "A Multi-user Random-Access Communication System for Users with Different Priorities", Proceedings of 26th Conference on Decision and Control (CDC), Dec. 1988, Los Angeles.
15. L. Georgiadis, P. Papantoni-Kazakos, "Limited Feedback Sensing Algorithms for the Packet Broadcast Channel", IEEE Transactions on Information Theory, Special Issue on Random Access Communications, Vol. IT-31, No. 2, pp. 280-294, March 1985.
16. R.G. Gallager, "Conflict Resolution in Random Access Broadcast Networks", Proc. AFOSR Workshop on Commun. Theory and Applications, Provincetown, MA, pp. 74-76, Sept. 1978.
17. J. L. Massey, "Collision-Resolution Algorithms and Random Access Communications", in Multi-User Communication Sustems, G. Longo, ed., CISM Courses and Lectures No. 265, pp. 73-137, Springer Verlag, New York, 1981.
18. L. Kleinrock, "Queueing Systems", Vol. I, pp. 279, John Wiley and Sons, 1976.
19. G. L. Chung, "A course in probability theory", Academic Press, Inc., 1974.
20. D. P. Heyman, M. J. Sobel, "Stochastic Models in Operations Research", Vol. I, McGraw-Hill, 1982.
21. J. L. Doob, "Stochastic Processes", pp. 464, John Wiley and Sons, 1953.
22. I. Stavrakakis, D. Kazakos, "2nd-order Markov Approximation on the Output Process of Multi-User Random-Access Communication Networks", Technical Report, University of Virginia (to be published).