A NON-HOMOGENEOUS USER POPULATION MULTI-USER RANDOM ACCESS COMMUNICATION SYSTEM

I. Stavrakakis, D. Kazakos

Department of Electrical Engineering
Thornton Hall, University of Virginia
Charlottesville, VA 22901

Abstract

A multi user random access communication system with a population of two classes of users is considered. It is assumed that packets generated by users from different classes have different priorities.

A binary feedback collision resolution algorithm is developed for such a communication system and both throughput and delay analysis are performed.

I. Introduction

So far the existing literature on the multi user random access communication systems has been dealing with a homogeneous population of users [1]-[5]. There are many practical applications, however, where it is desired that some packets experience shorter delays than the average regular packet of the system. If all users are to use the same communication system, then the need for dividing the population of users into two classes arises.

There are cases of communication systems with homogeneous population of users where, at specific known time periods, the input traffic to the channel decreases significantly with respect to the nominal point of operation of the system. As a result, the average packet delay decreases but the utilization of the system decreases as well. Under those conditions, we can improve the utilization of the system by letting a second class of same priority users have access to the system. By controlling the rate of the input traffic coming from the second class, we can achieve induced average packet delays for both classes around the nominal point of the original class. In that case, the same algorithm applies to both classes and in fact we have a homogeneous user population. A second option is to adopt an algorithm that gives priority to the packets of the original class. In that case, it is expected that if the induced average packet delay of the original (high priority) class is around its nominal value, then the induced delays of the second class will be significantly larger. On the other hand, the low priority packet traffic, that induces the nominal average packet delay for the high priority class, is expected to be much larger than in the previous case of the equivalent classes. If the users of the second class can wait for the occurence of the low traffic time periods of the original system, then it is reasonable to assume that those users can tolerate an additional delay of a small number of packet lengths. Thus, by using a system with users with different priorities, we can greatly increase the utilization of a system at essentially no cost.

In a mobile user environment where users move in and out of the range of the system, or move from region to region, fast moving users may need to experience shorter delays than the regular ones; this may be necessary to make packet transmission possible while the user is still inside the region. Also, users that are close to the boundaries of a region and are going to move outside it, should experience shorter delays.

In a static user environment there are also cases in which some packets have high priority and should reach their destination faster than the regular ones. High priority packets can be those which are generated by high priority users (e.g. important users, or users that can pay more for better service), or can be packets that are generated by any user of the system but the information that is carried is characterized as important and deserves high priority in its transmission.

An important measure of performance of a communication system is the induced average packet delay. In some environments, there may exist strict constraints on the delay that some packets can tolerate. If a threshold is exceeded, the packet is considered to be lost and the average number of those packets can be a measure of performance, [6]. By considering that those special packets form a separate class which is given priority by the system, we might be able to reduce the induced delays of those packets below the rejection threshold and thus greatly improve the performance of the system.

In the next two sections the communication system and the suggested algorithm are described. In section IV throughput and delay analysis are discussed, while in the last section the results are shown and conclusions are drawn.

II. The Communication System

We consider a large population of users that use a single communication channel. We assume that users which for some reason need to have some priority over the rest of the population, form the high priority class. It is assumed that the packet traffic generated by that class represents only a small percentage of the total traffic that is served by the system. In other words, we assume that the packets that need special service are rare and this is a realistic assumption at least for the environments that were described above. The input traffic to the channel that is generated by each class of users is assumed to be Poisson distributed with intensities λ_f and λ_s respectively.

Research jointly supported by the Air Force Office of Scientific Research through Grant AFOSR 82-0030 and the U.S. Office of Naval Research through contract ONR-N14-86-K-0742.

Messages are assumed to be packetized and of fixed length; it is assumed that time axis is slotted and that the beginning of a packet transmission coincides with the beginning of a slot. Because of the freedom that the users enjoy in accessing the channel, a transmission attempt results in either a successful packet transmission, or in a packet collision if more than one packet transmissions were attempted in the same time slot. Thus it becomes obvious that an algorithm is necessary in order for the conflicts to be resolved and the channel to remain usable.

It is assumed that all users that have a packet to transmit (and only these users need to do that) keep sensing the channel and are capable of detecting a packet collision; that is, we assume that a binary feedback information is available to all active users before the end of the current slot, revealing whether the slot was involved in a packet collision (C) or not (NC). Channel errors are not taken into consideration and packet collision is the only event that results in unsuccessful transmission.

III. Description of the Algorithm

The first time transmission policy is kept the same for both classes of users. It is simple and implies that a packet is transmitted at the beginning of the first slot following the packet generation instant. It is apparent that if the two classes are to experience different delays, they should follow different steps in the collision resolution procedure.

A simple limited sensing collision resolution algorithm is developed. The limited sensing characteristic is apparently important for a mobile user environment since the users may not be able to know the history of the channel before their packet generation instant. We assume that the state of a user is determined by the content of a counter that is assigned to each one of them; this counter is updated according to the steps of the algorithm and the feedback from the channel. Users whose counter content at the beginning of a time slot is equal to one, transmit in that slot.

Let $c_i^f(c_i^s)$ denote the counter content of a high priority (regular) user, at the beginning of the i^th^ time slot. Let also F_i, $F_i \varepsilon (C, NC)$, denote the channel feedback information just before the end of the i^th^ time slot. The steps of the collision resolution algorithm consist of the following counter updating procedures that take place at the end of each time slot.

(A) If $F_i = C$ then

$$c_i^f = 1 \begin{cases} c_{i+1}^f = 1 & \text{with probability } \phi \\ c_{i+1}^f = 2 & \text{with probability } 1-\phi \end{cases}$$

$$c_i^s = 1 \begin{cases} c_{i+1}^s = 2 & \text{with probability } \sigma \\ c_{i+1}^s = 3 & \text{with probability } 1-\sigma \end{cases}$$

$$c_i^j = r \rightarrow c_{i+1}^j = r+2 \,, \ r \geq 2 \,, \ j \varepsilon (s, f)$$

(B) If $F_i = NC$ then

$$c_i^j = r \rightarrow c_{i+1}^j = r-1 \,, \ r \geq 1 \,, \ j \varepsilon (s, f)$$

It did not seem to us reasonable to develop different first time transmission policies for the two classes of users. It would probably be a waste of the channel capacity to give priority to rarely appearing high priority packets, before it becomes known that a collision took place. If a conflict occurs, then the collision resolution algorithm offers some priority to the high priority packets that were involved in the conflict.

IV. Throughput/Delay Analysis

In this section we describe the derivation of the stability region of the algorithm and the calculation of the mean packet delay. For this purpose, we use the concept of the session and develop recursive equations to describe the operation of the system. A session is defined as a number of consecutive slots between properly selected renewal points of the system. If μ high priority users and ν regular ones attempted a packet transmission in the first slot of a session, then the pair (μ, ν) determines the multiplicity of that session.

At this point we give the following definition for the stability region of the system.

Definition:

If for an input traffic pair (λ_f, λ_s), the expected value of the session length of multiplicity (μ, ν) is finite, for μ and ν finite, then we say that the operation of the system is stable and the pair (λ_f, λ_s) belongs to the stability region of the system. The maximum overall sets of stable points (λ_f, λ_s) determines the maximum stable throughput region and is denoted by $S_{\max}$.

By following procedures similar to those which can be found in [4], [7], [8], we calculate a linear upper bound on the mean session length of multiplicity (μ, ν), $L_{\mu,\nu}$. The set of pairs (λ_f, λ_s) for which such a bound was possible to obtain, is a lower bound on the stability region of the algorithm. An upper bound can be obtained by solving a truncated version of an infinite dimensionality linear system of equations with respect to $L_{\mu,\nu}$, [14]. The latter system is obtained by considering the expectations of the recursive equations which describe the operation of the system. The stability region of the algorithm is plotted and it is shown in Fig. 1.

The mean delay of the high and low priority packets is also calculated but only for input traffic pairs (λ_f, λ_s) such that $\lambda_f \leq .065$ packets per packet length. For that region, bounds on the involved quantities was possible to obtain. This range of pairs determines the operation region of the algorithm; i.e.

$$S_{op} = \left\{ (\lambda_f, \lambda_s) : 0 \leq \lambda_f \leq 0.065, \ 0 \leq \lambda_s \leq \lambda_{s,\max}(\lambda_f) \right\}$$

where $\lambda_{s,\max}(\lambda_f)$ can be obtained from Fig. 1. The delay analysis is performed by applying the regeneration theory procedures that appear in [12], [4], [9], [10], or by using directly the strong law of large numbers, [11], [7]. Very tight upper and lower bounds on the mean delay of the high and low priority packets, D_f and D_s, respectively, were calculated for some values of the input traffic; the results appear in Table 1.

V. Results and Conclusions

The algorithm that we developed and analyzed is supposed to operate in an environment where two classes of users with different priorities are accommodated. An algorithm for a homogeneous user population that would work in a similar way and use binary feedback information and simple splitting after a collision, has been found to achieve a maximum stable throughput of ~ .36 [13]. The algorithm that we suggest for a non-homogeneous population achieves total throughput, at least, between .320 - .357 depending on the contribution of the two classes to the total input traffic.

In Fig. 2, Fig. 3 and Fig. 4, plots of the bounds on D_f and D_s versus λ_s, for λ_f=0.01, λ_f=0.03 and λ_f=0.065 respectively, are shown. These values of λ_f correspond to an input traffic coming from the high priority class equal to ~ 3%, ~ 10% and ~ 20% of the total traffic that can be served by the system. From the plots it can be observed that the high priority packets experience shorter delays than the packets of the other class; the difference is essential for λ_s>.5$\lambda_{s,max}$. If the nominal point of operation of the system is set around λ_s=.9$\lambda_{s,max}$, then the average high priority packet delay is less than half the one of the other class.

In table 1, the delay results of the suggested algorithm are compared with the delay, D*, that the homogeneous class equivalent algorithm (as described above), induces [13]. Again we can observe that always D_f<D* and particularly D_f<.5D* around the nominal point, the latter being defined as before.

Since privileged service is offered to some users, there has to be a price that the rest of the population must pay. The first consequence is the small reduction in the total throughput, as mentioned before. The other penalty is the increased average low priority packet delay compared with the one that the homogeneous population equivalent algorithm induces. From table 1 we can see that, indeed, D_s>D*, as it was expected. The increase in D_s is far from catastrophic and it is realistic to consider that it is possible for a system to tolerate these delay increases for the low priority class, especially if strict limitations exist for the high priority users.

As an example, consider the communication system described in the second paragraph of the Introduction. Assume that the input traffic of the original class at the nominal operating point is .25 packets/packet length and thus the (desired) induced average packet delay is 5.5 - 6.0 packet lengths (last column of table 1). Assume that at night, the input rate falls to 0.065 packets/packet length. At that time, a second class of users is given permission to use the channel. If the induced average packet delay of the original class has to be at most ≈ 6.00 packet lengths, then depending on the case we observe the following: (a) If the second class has the same priority as the original, then the additional input traffic rate that can be accommodated by the system is 0.185 packets/packet length. (b) If the second class has low priority, then the additional input traffic rate becomes .25 packets/packet lengths (table 1). Thus, there is an increase by ≈ 35% of the additional traffic that can be accommodated, if the population of users is divided into two classees with different priorities. The increase in the average packet delay of the low priority users is rather negligible compared to a realistic waiting time until these users are given permission to access the channel.

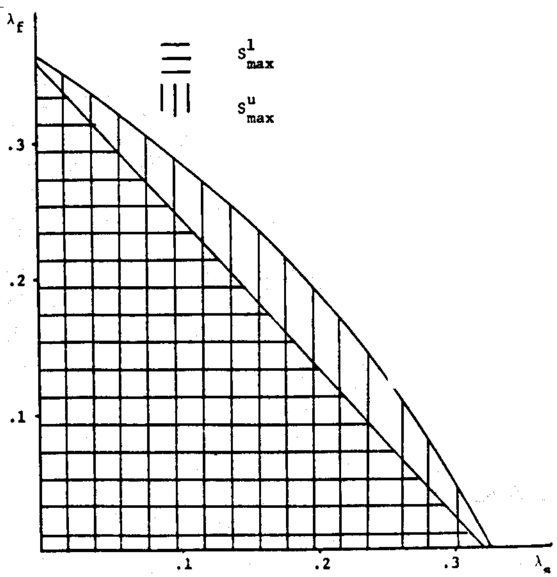

Figure 1. Upper, S^u_{max}, and lower, S^1_{max}, bounds on the maximum stable throughput; λ_f and λ_s are in packets/packet length.

λ_f	λ_T	λ_s	$D_f^l - D_f^u$	$D_s^l - D_s^u$	D^*
.01	.02	.01	1.555	1.590	~1.57
	.11	.10	1.829	2.369	~2.10
	.18	.17	2.186	3.815	~2.90
	.26	.25	3.095	9.922	~6.20
	.31	.30	5.793	39.793	~16.00
	.32	.31	8.718	78.748	~23.00
.03	.04	.01	1.632	1.678	~1.66
	.13	.10	1.951	2.571	~2.21
	.20	.17	2.389	4.312	~3.33
	.28	.25	3.672	12.681	~8.33
	.31	.28	5.453	28.961	~16.00
	.32	.29	7.113	45.905	~23.00
.065	.075	.01	1.800	1.878	~1.82
	.165	.10	2.234	3.054	~2.70
	.235	.17	1.159	2.900	5.595
	.315	.25	5.801	23.101	~18.00
	.325	.26	7.200	33.080	~26.00

Table 1

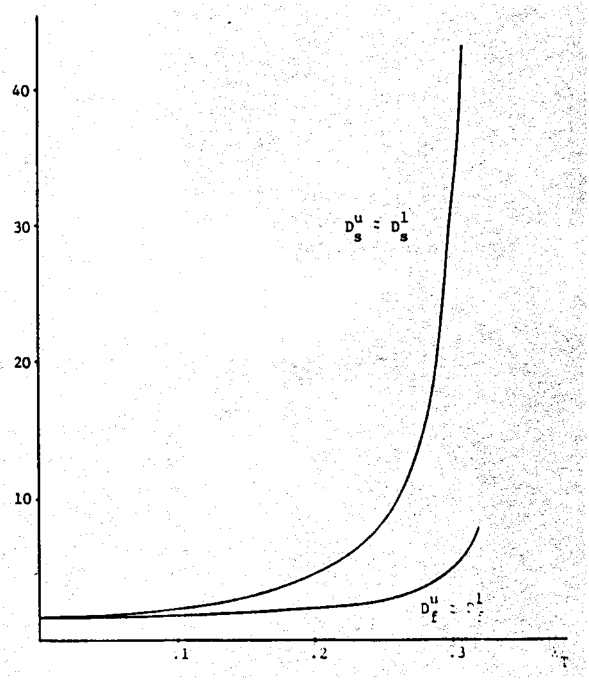

Figure 2. Average packet delay of the high, D_f, and the low, D_s, priority classes (in packet lengths) versus the total input traffic rate, λ_T, (in packet length), for λ_f = .01.

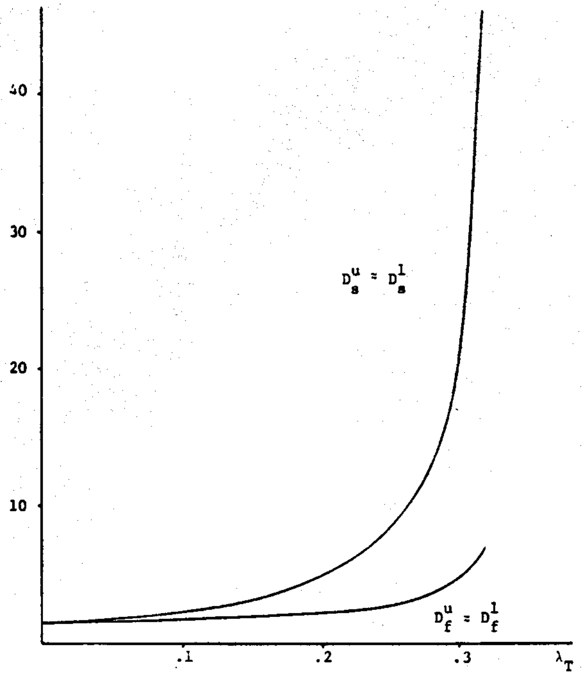

Figure 3. Average packet delay of the high, D_f, and the low, D_s, priority classes (in packet lengths), versus the total input traffic rate, λ_T, (in packets/packet length), for λ_f = .03.

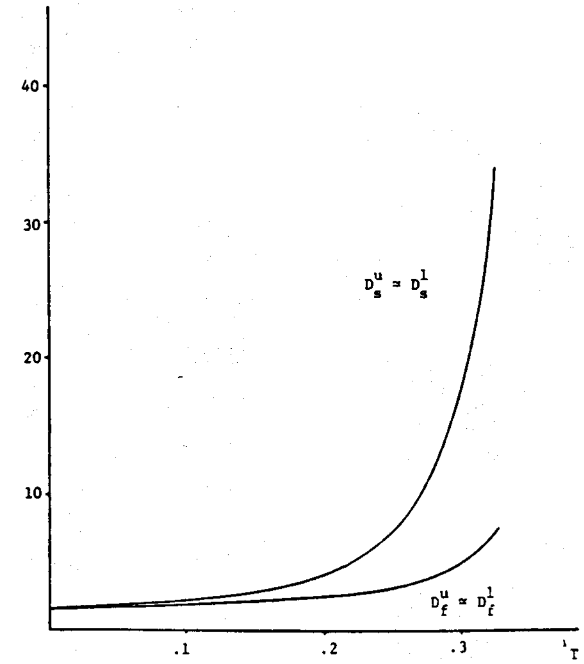

Figure 4. Average packet delay of the high, D_f, and the low, D_s, priority classes (in packet lengths) versus the total input traffic rate, λ_T, (in packets/packet length), for λ_f = .065.

References

[1] B.S. Tsybakov, N.D. Vvedenskaya, "Random Multiple Access Stack Algorithm", translated from Problemy Peredachi Informatsii, Vol. 16, No. 3, pp. 80-94, July-September 1980.

[2] J.I. Capetanakis, "Tree Algorithm for Packet Broadcast Channel", IEEE Transactions on Information Theory, Vol. IT-25, No. 5, pp. 505-515, Sept. 1979.

[3] 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.

[4] 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.

[5] N. Abramson, "The ALOHA System - Another Alternative for Computer Communications", Proc. AFIPS Fall Joint Computer Conference, Houston, Texas, Nov. 17-19, 1970, AFIPS Press, Montvale, N.J., pp. 281-285.

[6] J. Kurose, M. Schwartz, Y. Yemini, "Multiple Access Protocols and Time Constrained Communication", ACM Computing Surveys, Vol. 16, No. 1, March 1984.

[7] I. Stavrakakis, D. Kazakos, "A Simple Stack Algorithm for a Code Division Multiple Access Communication System", Technical Report, No. UVA/525656/EE87/101, University of Virginia, October 1986.

[8] N.D. Vvedeuskaya, B.S. Tsybakov, "Random Multiple Access of Packets to a Channel with Errors", Problemi Peredachi Informatsii, Vol. 19, No. 2, pp. 52-68, April-June 1983.

[9] J.W. Cohen, "On Regenerative Processes in Queueing Theory", New York: Springer-Verlag, 1976.

[10] S. Stidham, Jr., "Regenerative Processes in the Theory of Queues, with Applications to the Alternating-Priority Queue", Adv. Appl. Prob., Vol. 4, pp. 542-577, 1972.

[11] G.L. Chung, "A course in probability theory", Academic Press, Inc. 1974.

[12] L. Georgiadis, L. Merakos, P. Papantoni-Kazakos, "Unified Method for Delay Analysis of Random Multiple Access Algorithms", Technical Report, UCT/DEECS/TR-85-8, University of Connecticut, Aug. 1985. Also submitted for publication.

[13] P. Mathys, "Analysis of Random Access Algorithms", Ph.D. dissertation, Diss. ETH No. 7713, Swiss Federal Institute of Technology (ETH), Zurich, 1984.

[14] L.V. Kantorovich, V.I. Krylov, "Approximate methods of higher analysis", pp. 21, Interscience Publishers, 1958.