

December 3, 1984
revised August 28, 1985

LIDS-P-1426

ON THE THROUGHPUT OF CHANNEL ACCESS ALGORITHMS WITH LIMITED
SENSING

Pierre A. Humblet

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology

ABSTRACT

We consider some access protocols for Aloha type multiaccess channels. We argue, and show in an important case, that they can be modified to allow new transmitters to join the system at arbitrary times. This feature, known as "limited sensing" or "continuous entry" need NOT reduce throughput performances. In the case presented, the modified algorithm is also robust with respect to feedback errors.

This research was conducted at the M.I.T. Laboratory for Information and Decision Systems with partial support provided by the National Science Foundation under Grant NSF-ECS-8310698 and by the Defense Advanced Research Projects Agency under Contract ONR/N00014-84-K-0357.

ON THE THROUGHPUT OF CHANNEL ACCESS ALGORITHMS WITH LIMITED SENSING

Pierre A. Humblet

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology

INTRODUCTION

We consider the classical Aloha type multiaccess channel where packets are generated at a large number of sites and are eventually transmitted on a common channel. Overlapping transmissions result in a collision and all the packets involved must be retransmitted. Transmitters monitor the activity on the channel and obtain some type of feedback information, depending on the precise model at hand. The problem is to design protocols that exploit feedback information to schedule transmissions so as to maximize the achievable throughput and/or cause little average delay for a given throughput.

The algorithms with the best performances require all transmitters to monitor the channel at all times. Some attention has also been devoted to channels with "limited sensing" or "continuous entry" where a transmitter only monitors the channel while it has a packet ready for transmission. The words "free access" are used to denote "limited sensing" algorithms where a packet MUST be transmitted immediately following its generation. "Limited sensing" algorithms have practical advantages over algorithms that require continuous observations. The "Free access" characteristics on the other hand does not seem to be as important, except that it guarantees minimum delay in very light traffic (however it reduces throughput in a sense explained briefly below).

All the "limited sensing" algorithms described previously exhibit achievable

throughputs lower than those achievable by algorithms monitoring the channel continuously [2],[4],[9]. This situation is rather unexpected! As there is no requirement that delay be kept small, a transmitter can listen to the channel for a long time before transmitting a generated packet. Doing so should put it in a "state of synchronization" close to what it would have had by listening to the channel since the beginning of operations.

We cannot show at this time that in general limited sensing does not reduce achievable throughput, but we will illustrate our contention for the slotted channel with ternary feedback ("idle", "success", "collision"). There packets are only transmitted in predefined slots and transmitters can learn immediately whether zero, one, or more than one transmissions took place in a slot. The degenerate case of binary ("collision", "no-collision") feedback will also be treated.

OUTLINE OF THE METHOD

A class of well known algorithms for that channel are variations of the Gallager [1], Tsybakov and Mikhailov [10] algorithms. For our purpose we view them as being in one of three phases of operation, depending on the previous transmission outcomes. This is illustrated in figure 1 where letters label the channel outcomes associated with phase transitions. In all the phases a subset of the time axis is selected and the packets that were generated during that subset are transmitted. The details of the operations in each phase follow.

Phase I

The algorithm allows transmissions from a set T for which only a priori statistical information is available. (We assume that packets are generated according to a Poisson process of finite rate.) It immediately returns to phase I (choosing another set) if no collision occurs, else it moves to phase C.

Phase C

Set T is partitioned into subsets L and R , and only transmissions from L are allowed.

- (1) An outcome of idle means that R must contain at least two active

transmitters. The algorithm abandons L, partitions R and continues in phase C.

(2) A collision outcome for L implies (under Poisson statistics assumption) that only a priori information is known about transmitters in R. The algorithm partitions L and remains in phase C.

(3) After a success in phase C the algorithm moves to phase S.

Phase S

Transmissions from R are allowed. The next phase is C or I, depending on the outcome, which cannot be "idle" (except for the collisions, we assume error-free transmissions).

As a variation, outcomes of "idle" in phase C could cause transitions to phase S. This modified algorithm does not differentiate between the outcomes of "idle" and "success", but it leads to sure collisions following "idles" in phase C. An analysis of this modification appears in [8].

In Gallager's algorithm packets are transmitted in the order in which they are generated as the sets T are chosen to include outstanding packets that were generated least recently. If their expected number does not exceed 1.266 (we call this case a "hit") they are all included in T, else T consists of a time interval containing an expected number of 1.266 packets.

The subintervals are obtained by divisions in equal parts ([10] partially optimizes on subinterval sizes) . These choices are almost optimal with respect to throughput (.4871) when the outcomes of "idle", "success" and "collision" have the same transmission times. [6] contains a slightly improved version of the algorithm and its extension to channels where different outcomes have unequal duration, e.g. when quick carrier or collision detections are possible. The observations below apply to all these variations.

It has long been recognized that the phase of the algorithm can sometimes

be determined by observing the transmission outcomes. After hearing a collision on the channel one can immediately conclude that the algorithm is in phase C. Similarly a success followed by another success or by an idle unambiguously signals a return to I. Only long strings of idles cause ambiguity as they can occur both in the I and C phases.

We suggest modifying the algorithm to force a collision after $N-1$ idles in phase C (for some $N > 1$) by allowing transmissions from the entirety of set R (a similar method has been proposed [7] to recover from some feedback errors). This modification guarantees that new listeners will be "in phase" within at most N slots, while reducing the achievable throughput by occasionally wasting a slot. The probability of N successive "idle"s in phase C decreases geometrically fast with increasing N , and so does the throughput degradation. As a side effect of the modification, the algorithm that we propose below is also robust with respect to feedback errors that can cause Gallager's algorithm to deadlock. In the case $N = 2$, no effective distinction is made between the outcomes of "idle" and "success", so that the algorithm only requires binary feedback [8]; in that case limited sensing does not cause additional slots to be wasted.

Now that we can synchronize new listeners, it is a simple matter to make limited sensing work. We will allow "new" transmitters to transmit only when the algorithm is in the I phase and we will let transmissions be essentially Last In First Out, as in many protocols with limited sensing. Thus "old" transmitters, which have more information, defer to "new" transmitters in such a way that the properties of the original algorithm are preserved.

Imagine that an observer watching the channel since the beginning of operations has iteratively produced the picture of the time axis appearing in figure 1.a). As transmissions out of L take place, current time is advanced. The W set is extended to the right, while its left part is possibly updated into The L set is updated into x (upon idle or success) or split into L and R (upon collision). The R set is returned to (upon collision) or split into L and R (upon idle).

When the algorithm reaches the I phase a new set T is selected from the

set, starting at the left boundary of the updated s set, in a Last In First Out fashion. For example if the L and R sets in the figure 2.a) each contained one packet, the new figure might be as in figure 2.b). The phenomenon corresponding to a "hit" in the original algorithm is that the T set includes times before the system has started.

Observers that have joined the channel at some time can recreate the part of the previous picture to the right of their arrival time by listening to the channel outcomes, so that in particular transmitters can decide to what set they belong and if they must transmit.

It has come to our attention that similar results have also been described by L. Georgiadis and P. Papantoni-Kazakos [3].

B. Tsybakov [11] has also observed that limited sensing does not decrease capacity, but under assumptions different from ours. In his scheme slots are grouped into "frames" of fixed size and all transmitters are somehow A PRIORI aware of the frame boundaries. Packets generated during a frame start accessing the channel (following the usual method) at the beginning of the following frame. Often they can all be successfully transmitted in less than a frame. However if the collisions between them are not resolved in a single frame then the collision resolution process for that frame is suspended and resumes (in last in first out fashion) when there is time available at the end of a subsequent frame. The assumption that frame boundaries are known avoids the need to synchronize and the occasional wasting of a slot.

The practical importance of limited sensing should not be overrated. The situation of greatest practical interest is that where a transmitter joins the channel while the system is in operation, but not just before a packet is generated locally. The transmitter could start measuring time and follow the original algorithm (selecting T sets, splitting etc...) from the time it joins the system. As soon as a true "hit" occurs, and if no idle ensues in the possible transmission of a L set, then the transmitter will be effectively synchronized with the rest of the system.

THROUGHPUT ANALYSIS

Conclusions about achievable throughputs can readily be obtained from existing results on the original algorithms. For example in the canonical case of equal outcome durations, with splitting into equal subintervals, we know that the achievable throughput for $N = 2$ is .4494 [8]. The limit of the throughput as N increases is .4871 [1] and, as suggested above, one expects the throughput loss to be divided by a factor of 4 (the probability of an "idle" in the C phase is about .25) each time N is increased by one.

Using the fact that returns to the I state constitute renewal points in a system where full size intervals T can be selected, the achievable throughputs are readily obtained as the ratio of the expected number of successes to expected number of slots between passages through the I state in the Markov chain in figure 3. That chain is essentially the same as that considered in [1], with the provision that in the C phase the state must also keep track of the number of successive "idle"s.

The quantities of interest are readily computed recursively from right to left in a truncated chain. Note that the expected number of successes in a state of the C phase is independent of the successive number of "idle"s, which simplifies the computation. Results are presented in table I; there we have optimized on the size of the interval T .

It is also interesting to consider delay. In very light traffic the expected delay will be exactly N slots, which shows that there is a tradeoff between delay in light traffic and achievable throughput. At higher loads limited sensing will increase expected delay by at most N but will also considerably increase the variance of the delay due to the Last In First Out nature of the algorithm.

A NOTE ON THE ACHIEVABLE THROUGHPUT OF FREE ACCESS ALGORITHMS

If a genie were to identify the terminals involved in a collision, one could do no better than allow them to retransmit successively. In a free access environment there is always the possibility of collision with freshly generated (at rate t packets/slot, say) packets. When a packet is retransmitted, the retransmission is successful with probability $\exp(-t)$. This leads to the condition $t < \exp(-t)$ to have a stable system, or $t < .5671$.

This number is smaller than the best known upperbound [5] on the achievable throughput with ternary feedback when there is no free access restriction.

CONCLUSION

We have shown in specific instances that existing access algorithms can be modified to use limited sensing only, while degrading achievable throughput by arbitrarily small amounts, at the expense of some extra delay. We believe that this holds true in more general settings.

ACKNOWLEDGEMENTS

Dr. P. Mathys kindly provided us with a copy of [8] and participated in discussions on the throughput of free access algorithms. Helpful comments from the reviewers are gratefully acknowledged.

REFERENCES

- [1] R.G. Gallager, "Conflict Resolution in Random Access Broadcast Networks", Proceedings AFOSR Workshop on Communication Theory Applications, Provincetown, MA, pp.74-76, Sept. 17-20, 1978.
- [2] L. Georgiadis and P. Papantoni-Kazakos, "Limited Feedback Sensing Algorithms for the Packet Broadcast Channel", IEEE Transactions on Information Theory, Vol IT-31, No 2, pp. 280-294, March 1985.
- [3] L. Georgiadis and P. Papantoni-Kazakos, "A 0.487 Throughput Limited Sensing Algorithm", University of Connecticut, Storrs, 1985
- [4] P. Mathys and P. Flajolet, "Q-ary Collision Resolution Algorithms in Random-Access Systems with Free or Blocked Channel-Access", IEEE Transactions on Information Theory, Vol IT-31, No 2, pp 217-243, March 1985.
- [5] V.A. Mikhailov and B.S. Tsybakov, "Upper Bound for the Capacity of a Random Multiple Access System", Problemy Peredachi Informatsii, Vol. 17, No. 1, pp. 90-95, Jan. 1981.

- [6] J. Mosely and P.A. Humblet, "A Class of Efficient Contention Resolution Algorithm for Multiple Access", IEEE Transactions on Communication, Vol. Com-33, Number 2, Feb. 1985, pp 145-152
- [7] D. Ryter, "A Conflict Resolution Algorithm for Noisy Multiaccess Channels", MIT Laboratory for Information and Decision Systems, Report Th 1007, May 1980.
- [8] P. Studer and H. Pletscher "Q-ary Part-and-Try Algorithm for Packet Conflict Resolution", Institute of Telecommunications, Swiss Federal Institute of Technology, Zurich, Switzerland, October 11, 1984.
- [9] B.S. Tsybakov and N.D. Vvedenskaya, "Random Multiple-Access Stack Algorithm", Probl. Peredachi Inf., Vol 16, No 3, pp. 80-94, July-Sept 1980.
- [10] B.S. Tsybakov and V.A. Mikhailov, "Random Multiple Packet Access: Part-and-Try Algorithm", Probl. Peredachi Inf., Vol 16, No 4, pp. 65-79, Oct-Dec 1980.
- [11] B.S. Tsybakov, "Some New Random Access Protocols", presented at the 1985 IEEE International Symposium on Information Theory, Brighton, England, June 24-28 1985.

TABLE 1

ACHIEVABLE THROUGHPUT AS A FUNCTION OF N.
EQUAL OUTCOME DURATIONS.
SPLITTING INTERVALS IN EQUAL PARTS.

N	Achievable throughput	E(number of transmissions in T)
2	.4494	1.159
3	.4793	1.247
4	.4853	1.262
5	.4867	1.265
6	.4870	1.266
>6	.4871	1.266

FIGURE CAPTIONS:

Figure 1: Phase transition diagram

Figure 2: Examples of set labeling of the time axis

Figure 3: The Markov chain for throughput computation

TABLE CAPTIONS:

TABLE 1

ACHIEVABLE THROUGHPUT AS A FUNCTION OF N.
EQUAL OUTCOME DURATIONS.
SPLITTING INTERVALS IN EQUAL PARTS.

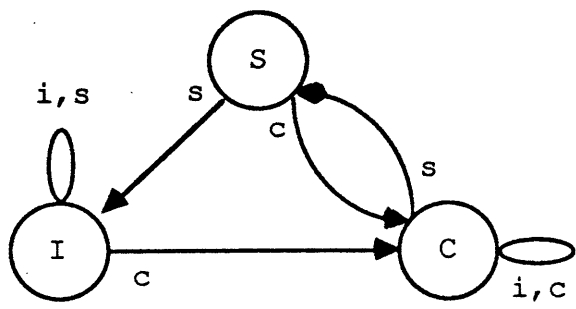


Fig 1

```

xxxxxxx.....xxxxx....RRRL..xxxxLLLLxxxxxxx...WWWW    -> time
0                                                         |
                                                         current time

```

(a)

```

xxxxxxx.....xxxxx....xxxx.TxxxxxxxxxxxxxxxxTTTTTTTW  -> time
0                                                         |
                                                         current time

```

(b)

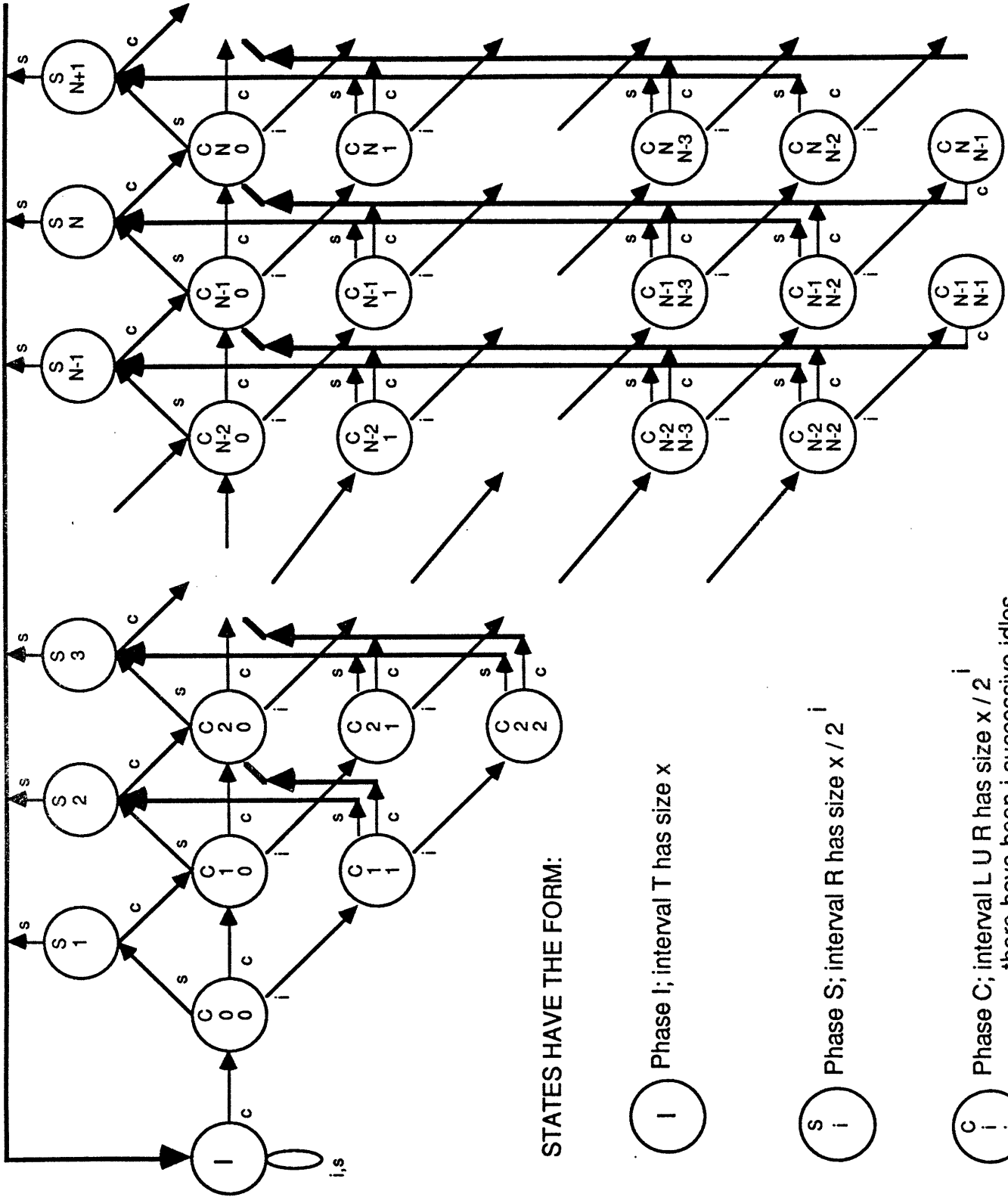
```

packets generated in WWWWW are synchronizing (with a priori stat.)
"      "      " ..... are synchronized (with a priori stat.)
"      "      " LLLLL form the L subset (phase C)
"      "      " RRRRR form the R subset (phases C and S)
"      "      " xxxxx have been successfully transmitted

```

The sets LL and RR appear above in keeping with the Last In First Out spirit. This feature is by no means necessary.

Fig 4



STATES HAVE THE FORM:

1 Phase I; interval T has size x

S i Phase S; interval R has size $x / 2$

C i j Phase C; interval L U R has size $x / 2$
there have been j successive idles