ON ACHIEVING A THROUGHPUT OF ONE FOR A RANDOM ACCESS CHANNEL WITH COLLISIONS OF KNOWN MULTIPLICITY*

Shivendra S. Panwar
Polytechnic University, Brooklyn, NY
Walter Grote
UTFSM, Valparaiso, Chile

January 25, 1995

Abstract- We consider the problem of a group of users sharing a common communication channel. The channel is slotted and the slot size corresponds to a packet length. At the end of each slot, every user is informed whether the slot was idle, contained a successful packet transmission, or experienced a collision of multiplicity k, $k \geq 2$. Throughput is defined to be the expected fraction of slots with successful transmissions. It has been shown, via a non-constructive proof, that a throughput of one can be achieved. However, the best throughput achieved thus far is approximately 0.532. We will describe a collision resolution algorithm (CRA) which achieves, in the limit, a throughput of one.

Index Terms- Random Access Communications, Collision Resolution Algorithms

*This work was supported by the National Science Foundation under Grants NCR-8909719 and NCR-9115864, and by the New York State Foundation for Science and Technology as part of its Centers for Advanced Technology Program. This work was presented in part at the IEEE International Symposium on Information Theory, Budapest, Hungary, June 1991.

1 Introduction

The problem of a group of users sharing a common channel has been widely investigated [8]. Much of the work assumes that, in the event of two or more packets simultaneously transmitting on the channel, the only information available to the users is that a collision has occured. It may be possible to determine the multiplicity of the collision by means of an energy level detector [1-7]. This paper will show that this additional information can be used to achieve a throughput approaching one in the limit.

We will briefly describe the model next. The channel is slotted and the duration of each slot corresponds to a packet length. Packet transmissions are timed to correspond to slot durations on the channel. At the end of each slot, every user is informed whether the slot was idle (contained no transmission), had a successful transmission (contained one packet), or experienced a collision of multiplicity k, $k \geq 2$ (contained k packets). The packet arrivals from all the users is assumed to be Poisson with rate λ. Throughput is defined to be the expected fraction of slots with successful transmissions. A collision resolution algorithm (CRA) is used to control retransmissions of colliding packets until they are successfully transmitted. The objective of a CRA is to maximize the throughput of the channel while maintaining stability.

It has been shown by Pippenger [1], via a non-constructive proof, that a throughput of one can be achieved. However, the best throughput achieved thus far has achieved a throughput of approximately 0.532 [2]. This CRA used a nested scheme to successfully transmit all packets in a group of colliding packets.

2 Description of the Collision Resolution Algorithm

Consider two time axes, the packet arrival time axis and the slotted channel time axis. The packet arrival time axis records the arrival of packets for transmissions from all the users. An *epoch* will be defined as the total length of time, which in general may consist of non-contiguous time intervals, as seen on the channel and measured in channel slots, required to successfully transmit all packets that arrived in an interval of length $\triangle$ on the arrival time axis.For this CRA we will choose $\triangle$ such that $\lambda\triangle = (N-1)/2$, where N is an integer, $N > 2$. Let us first consider how this algorithm operates after a time interval $(0, t)$, when all arrivals have been successfully transmitted. Let the current time be CT. Then an epoch is the number of slots the CRA requires to transmit successfully all arrivals in the interval $(t, min(t+\triangle, CT))$. In the next slot, $(CT, CT+1)$ on the channel time axis, all the packets in the interval $(t, min(t+\triangle, CT))$ are transmitted or *enabled.* Intervals are enabled in first-come first-served order, i.e. no interval of length $\triangle$ is enabled before intervals on the arrival axis before it have not already been enabled. If the channel is idle or a successful packet is transmitted as a result of there being none or exactly one packet arrival in the interval $(t, min(t+\triangle, CT))$, respectively, we have successfully resolved the interval and the epoch ends. Let the expected length of an epoch when an interval of length $\triangle$ contains k packet arrivals be L_k slots. Thus $L_0 = L_1 = 1$. If a collision of multiplicity $k, k \geq N$, occurs, then use a nested scheme, as described in [2], to successfully retransmit all k colliding packets. If a collision of multiplicity $k, 2 \leq k < N$, occurs then set this interval aside and enable the next interval of length $\triangle$. Continue this process until we have set aside $F_N(m)$ such intervals. Here $F_N(m)$ is an integer-valued function of N and m, as defined in [9]. Let us define a binary "detecting" matrix $A_{m,N}$ [9], of size $m \times F_N(m)$. Also, split each of the $F_N(m)$ intervals into two sub-intervals, as described in [2], and denote the number of arrivals in the first sub-interval of each by $x_i, 1 \leq i \leq F_N(m)$. Now each row of the binary matrix $A_{m,N}$ can be considered an enabling, where a one or zero is equivalent to including or leaving out the first sub-interval of the corresponding interval from the set of sub-intervals enabled. The feedback from a set of m enablings, which we will call a *pass*, can be shown to be unique for every distinct set $\{x_i, 1 \leq i \leq F_N(m), x_i = 0, 1, 2, \ldots, N-1\}$. Since $F_N(m) > m$ for sufficiently large m, this leads to considerable savings in the number of slots needed to determine the value the x_i's. Continue using a nested procedure [2] on the two sub-intervals, until every interval of length $\triangle$ can be resolved into sub-intervals which contain either one or no arrivals. After each pass, any sub-intervals known to contain only one arrival may be enabled for successful transmission. During this process, the original $F_N(m)$ intervals will be subdivided into several smaller sub-intervals. In each pass of the detecting matrix, we need $F_N(m)$ such sub-intervals, each of which have between 2 and $N-1$ arrivals. We shall assume that sub-intervals are chosen for a pass in first-come first-served order. As these sub-intervals are resolved, there will be a time when less than $F_N(m)$ sub-intervals will be available for a pass. At this time a new interval or intervals of length $\triangle$ will be enabled until we can accumulate a total of $F_N(m)$ intervals with $k, 2 \leq k < N$, arrivals in each one of them. Note that if any of the new intervals have k arrivals, where $k = 0, 1$ or $k > N-1$, it will be resolved as before. Thus at any arbitrary time, a new interval of length $\triangle$ is enabled only if either all earlier arrivals have been successfully transmitted or if less than $F_N(m)$ unresolved sub-intervals are available for a pass, which needs to be"filled out".

3 Stability of the CRA

For the CRA to be stable, the expected epoch length L should be less than $\triangle$, i.e.

$$L = E[L_k] \leq \triangle.$$

As mentioned in the last section, $L_0 = L_1 = 1$. Let us now consider $L_k, 2/leqk < N$. Excluding the initial enabling of an interval of length $\triangle$ and the k slots needed for successful transmissions, the expected number of slots needed to resove a collision of multiplicity k using the approach desribed in [2] can be bounded by $1.8784k, k \geq 2$. Excluding the initial enabling and successful transmissions, the number of passes required to resolve an interval corresponds exactly to the number of slots required in [2]. However, in this CRA, each such slot in [2] corresponds to $m/F_N(m)$ slots instead of one slot. This is due to the fact that the m slots required for a pass using the detecting matrix gives us the same information as would have $F_N(m)$ slots using the approach in [2] Therefore the corresponding expected number of slots needed can be bounded by $(m/F_N(m))1.8784k$. Thus, after taking into consideration the initial enabling and successful transmissions, $L_k < 1 + (m/F_N(m))1.8784k + k, N > k \geq 2$. For $k \geq N$, we may use the results from [2]. Thus $L_k < 1 + 1.8784k, k \geq N$.

Thus we can define L_k^{ub}, an upper bound to $L_k, k \geq 0$, as follows

$$L_k^{ub} = 1 + (m/F_N(m))1.8784k + k, N > k \geq 0,$$

and

$$L_k^{ub} = 1 + 1.8784k, k \geq N.$$

Also, recall that $\lambda\triangle = (N-1)/2$. Therefore

$$L^{ub} = E[L_k^{ub}] = \sum_{k=0}^{\infty} \frac{e^{-(N-1)/2}(\frac{N-1}{2})^k}{k!} L_k^{ub}$$

$$= \sum_{k=0}^{N-1} \frac{e^{-(N-1)/2}(\frac{N-1}{2})^k}{k!}(1+(m/F_N(m))1.8784k+k) + \sum_{k=N}^{\infty} \frac{e^{-(N-1)/2}(\frac{N-1}{2})^k}{k!}(1+1.8784k)$$

$$< 1 + (N-1)/2 + 0.8784 + 1.8784(m/F_N(m))(N-1)/2$$

where we have applied Chebyshev's inequality to bound the tail of a Poisson distribution. We can also upper bound $F_N(m)$ by using results from [9,10] as follows

$$F_N(m) > \frac{\frac{m+1}{2}\log(m+1) - \frac{m+1}{2}\log 4/3 - m}{\log N}$$

where all logarithms are base 2. Therefore the throughput, $\lambda_{m,N}$, can be upper-bounded as follows, by using the following sufficient condition for stability:

$$1.8784+(N-1)/2+1.8784(N-1)/2(\frac{m\log N}{\frac{m+1}{2}\log(m+1)-\frac{m+1}{2}\log 4/3-m}) < \triangle = (N-1)/2\lambda_{m,N}$$

which gives us

$$\lambda_{m,N} < \frac{(N-1)/2}{1.8784+(N-1)/2+1.8784(N-1)/2(\frac{m\log N}{\frac{m+1}{2}\log(m+1)-\frac{m+1}{2}\log 4/3-m})}$$

If we use very large, effectively infinite, size detecting matrices, then

$$\lim_{m\to\infty}\lambda_{m,N} < \frac{(N-1)/2}{1.8784+(N-1)/2}$$

The expression on the right achieves a maximum of one when N (or, equivalently, $\triangle$) is infinite. Since r.h.s $\to 1$ as $N \to \infty$, this CRA is stable as long as $\lambda < 1$. We will state this result as a theorem.

Theorem: The CRA described above for the random access channel with collisions of known multiplicity can achieve a throughput arbitrarily close to one.

This result can be extended to the Bernoulli model [1,6], where arrivals occur only at discrete time instants with probability p, by using Bernoulli probabilities instead of the Poisson distribution.

4 Concluding Remarks

We have shown a CRA that achieves a throughput approaching unity for the random access channel with collisions of known multiplicity. This is done at the cost of high delays. Further results giving the throughputs and a lower bound on the delay for the case when only finite and small collision multiplicities are detectable is reported in [11]. The CRA used here made use of "detecting" matrices, the construction of which was given by Lindström [9]. Note that the work in [1,7] used a different type of detecting matrix, which has not yet been constructed. Also, since neither of these matrices use the feedback information from each enabling, but only at the end of each pass, it may be possible to construct CRA's with lower delays than CRA's using a purely detecting matrix approach.

References

[1] N. Pippenger, "Bounds on the performance of protocols for a multiple-access broadcast channel," *IEEE Trans. on Information Theory*, vol IT-27, March 1981.

[2] L. Georgadis and P. Papantoni-Kazakos, "A Collision resolution Protocol for Random Access Channels with Energy Detectors," *IEEE Trans. on Communications*, vol COM-30, November 1982.

[3] B.S. Tsybakov, "Resolution of a conflict of known multiplicity," *Prob. Inform. Transmission*, vol 16, no. 2, translated from *Problemy Peredachi Informatsii.*

[4] M. Georgiopoulos and P. Papantoni-Kazakos, "Collision resolution protocols utilizing absorptions and collision multiplicities," *IEEE Trans. on Communications*, vol. COM-33, July 1985.

[5] N. Mehravari, " On Poisson Contention Resolution Problem with Feedback Based on Conflict Intensity," *IEEE Trans. on Communications*, vol COM-36, April 1988.

[6] N. Mehravari, " Random Multiple-Access Communication with Feedback Based on Conflict Multiplicity," *Proc. of the 1984 Conf. on Information Sciences and Systems*, Princeton University, March 1984.

[7] Roy D. Yates, "Methods of Multiple Access Communications with Energy Detectors," M.S. Thesis, LIDS - TH - 1557, M.I.T., May 1986.

[8] Special Issue on Random-Access Communications, James L. Massey, Guest Editor, *IEEE Trans. on Information Theory*, vol. IT-31, March 1985.

[9] B. Lindström, "On a Combinatorial Problem in Number Theory", *Canadian Mathematical Bulletin*, vol. 8, no. 4, June 1965.

[10] G.F. Clements and B. Lindström, "A sequence of (± 1) - determinants with large values, *Proc. Amer. Math. Soc.*, June 1965.

[11] Walter Grote, "Collision Resolution Algorithms for Random Access Channels", Ph.D. Dissertation, Department of Electrical Engineering, Polytechnic University, Brooklyn, NY, January 1992.

Explanation of changes made in response to reviewer's comments

The following are the two main changes that have been made to the paper:

(i) The operation of the collision resolution algorithm has been clarified.

The precise order in which intervals are enabled and processed by the detecting matrix was not clear in the previous version. The new version makes clear that
(a) Sub-intervals generated in the nested search are processed by the detecting matrix in FCFS order.
(b) New intervals are enabled in FCFS order and only when either previous intervals have all been resolved or if new intervals are needed to accumulate $F_N(m)$ intervals for the detecting matrix. This ensures that the queue of unresolved intervals cannot grow without bound. Indeed, it can be shown that the number of such intervals has a finite bound for finite N. Thus, even for values of λ that the system is unstable, the instability will lead a backlog of time intervals that are never enabled, not of unresolved sub-intervals.

(ii) A proof of stability is included.

Using the standard argument of negative drift (e.g. see [2]), we use a sufficient condition for stability to show that the CRA is stable, in the limit, for all arrival rates less than 1.

I have continued to refer to L_k as an epoch length, since even though it does not give a direct indication of delay as is the case in CRA's which resolve one interval at a time, since it does indicate the number of slots needed to resolve an interval with k arrivals, even if these slots, or fractions of slots (when the detecting matrix is used) are spread out in time.