An Asymptotically Nonadaptive Algorithm for Conflict Resolution in Multiple-Access Channels

JÁNOS KOMLÓS AND ALBERT G. GREENBERG, MEMBER, IEEE

***Abstract*—A basic problem in the decentralized control of a multiple access channel is to resolve the conflicts that arise when several stations transmit simultaneously to the channel. Capetanakis, Hayes, and Tsybakov and Mikhailov found a deterministic *tree algorithm* that resolves conflicts among k stations from an ensemble of n in time $\Theta(k + k\log(n/k))$ in the worst case. In this algorithm, at each step, the choice of which stations to enable to transmit depends crucially on feedback information provided by the channel. We show that if k is given *a priori* then such conflicts can be resolved in time $\Theta(k + k\log(n/k))$ using an algorithm in which the corresponding choices do not depend on feedback.**

I. Introduction

A MULTIPLE-ACCESS channel provides a low cost means for a large number of geographically dispersed computing stations to communicate. Several such channels have been proposed and some have been implemented, based on coaxial cable, fiber optic, packet radio, or satellite transmission media. A well-known example is the ETHERNET [4], [16], which uses a coaxial cable of up to 1.5 km in length to connect up to 1024 stations.

We consider the following model commonly taken as the basis of mathematical studies of the multiple-access channel [2], [3], [8]–[11], [15], [17]. Let n be the total number of stations tapped into the channel. At *steps* numbered $1, 2, 3, \cdots$, any station can transmit a packet of data to the channel. There is no central control. If k stations transmit simultaneously, the result depends on k as follows:

- If $k = 0$ then of course no packets are transmitted.
- If $k = 1$ then the packet is broadcast to every station, an event called a successful transmission.
- If $k \geq 2$ then all the packets are lost because the transmissions interfere destructively, an event called a collision.

All stations receive the feedback 0, 1, or $2+$, indicating that k is 0, 1, or ≥ 2, respectively before the next step.

Central to some recently proposed schemes for controlling access to the channel is an algorithm that, to *resolve* a transmission conflict, schedules retransmissions so that, with certainty, each of the conflicting stations eventually transmits singly to the channel [2], [3], [11], [15], [17]. Just those stations involved in the initial conflict participate. At each step of its execution, the algorithm gives transmission rights to a subset of the stations, called the query set. A station transmits if it

- is in the query set,
- is one of the stations that caused the initial collision, and
- is, as of this step, still unsuccessful in its attempts to transmit singly.

A conflict resolution algorithm may be used to coordinate access to the channel in the following way [2], [3], [11], [15], [17]. Access alternates between intervals in which access is unrestricted and intervals in which access is restricted to resolve conflicts. Initially access is unrestricted, and stations are permitted to transmit packets upon receipt. When a collision arises, the stations involved execute an algorithm to resolve it, and the other stations defer. All stations detect the algorithm's termination on the basis of channel feedback, at which point access is again unrestricted.

Capetanakis [2], [3], Hayes [11], and Tsybakov and Mikhailov [17] (independently) found a deterministic *tree algorithm* that resolves conflicts among k stations ($2 \leq k \leq n$) in $\Theta(k + k\log(n/k))$ time in the worst case.[1] (Worst case refers to the maximum over all possible choices of the k stations; time is measured as the number of steps used.) The algorithm works without any *a priori* information about k, other than $k \geq 2$. The algorithm is *adaptive* in the sense that, at each step, the choice of query set depends on the feedback elicited at previous steps. Greenberg and Winograd [10] showed that all deterministic conflict resolution algorithms must use time $\Omega(k(\log n)/(\log k))$ in the worst case, for all k and n ($2 \leq k \leq n$).

[1] $f(n) = O(g(n))$ means $|f(n)| \leq c|g(n)|$ for some constant $c > 0$ and all sufficiently large n.
$f(n) = \Omega(g(n))$ means $|f(n)| \geq c|g(n)|$ for some constant $c > 0$ and all sufficiently large n.
$f(n) = \Theta(g(n))$ means $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$. Bounds like $\Theta(k + k\log(n/k))$ hold uniformly in k (the same constant c works for all k) unless stated otherwise.

Manuscript received May 30, 1984; revised December 14, 1984.
J. Komlós is with the Mathematical Institute of the Hungarian Academy of Sciences, Budapest, and the Mathematics Department, University of California at San Diego, La Jolla, CA 92093, USA.
A. G. Greenberg is with the Mathematics and Statistics Research Center, AT & T Bell Laboratories, Room 2D-101, 600 Mountain Avenue, Murray Hill, NJ 07974, USA.

0018-9448/85/0300-0302$01.00 ©1985 IEEE

Here we consider *nonadaptive* conflict resolution algorithms. These are algorithms that, given both k and n, generate the same sequence of queries, irrespective of the feedback elicited. Specifically, a nonadaptive algorithm generates a list of queries, $Q_1, Q_2, \cdots, Q_t$, as a function of k and n. On inspection of this list, any conflicting station x may find the subsequence $Q_{i_1}, Q_{i_2}, \cdots, Q_{i_s}$ $(1 \le i_1 < i_2 \cdots < i_s \le t)$ of those queries enabling station x to transmit. At step i_j $(1 \le j \le s)$, station x transmits, provided that none of its earlier transmissions succeeded. Thus, a station must monitor feedback and reschedule its next transmission only at steps at which it transmits to the channel. In contrast, in the tree algorithm (which is adaptive), a station must monitor feedback and accordingly reschedule its next transmission at every step of execution.

Our main result is to show that there is a nonadaptive algorithm that, for all k and n, generates just $\Theta(k + k\log(n/k))$ queries, the same number (to within constant factors) as the tree algorithm. (A preliminary version of this result appeared in [9].) The proof is non-constructive, and is obtained using probabilistic methods.

The assumption that k is known *a priori* does not necessarily limit the practicality of nonadaptive algorithms. Any nonadaptive algorithm that works for given k and n also works for k' and n, provided $k' \le k$. Consider the following procedure for using a nonadaptive algorithm to construct a particularly simple adaptive algorithm to resolve a conflict of unknown multiplicity k. Try the algorithm for $k = 2, 4, 8, \cdots$, and, following each try, query the whole ensemble $\{1, 2, \cdots, n\}$. Terminate if this query elicits no transmissions. Suppose the running time of the underlying nonadaptive algorithm is $O(k + k\log(n/k))$. Then this procedure resolves conflicts of unknown multiplicity k $(2 \le k \le n)$ in time on the order

$$\sum_{i=1}^{p} 2^i\left(1 + \log\frac{n}{2^i}\right) = O\left(k + k\log\frac{n}{k}\right),$$

where $p = \lceil \log_2 k \rceil$. We note that an even faster nonadaptive algorithm could lead to an even faster adaptive one. Suppose the running time of the underlying nonadaptive algorithm of the procedure above is $O(k(\log n)/(\log k))$. Then the procedure resolves conflicts of unknown multiplicity k in time on the order

$$\sum_{i=1}^{p} (2^i/i)\log n = O\big(k(\log n)/(\log k)\big).$$

A related problem, *group testing*, deserves some mention. In *group testing*, the goal is to determine from the feedback to queries the identities of the k members of the input (a subset of size k from $\{1, 2, 3, \cdots, n\}$), under a model in which a query produces feedback 0 if the query holds no members of the input and feedback 1+ otherwise. In one sense, group testing is harder than conflict resolution, because in conflict resolution the algorithm need not determine the identities of the conflicting stations. In another sense, conflict resolution is harder than group testing, because in group testing the feedback is less informative. Group testing can be accomplished adaptively in $\Theta(k + k\log(n/k))$ time in the worst case [12], and nonadaptively

- in $\Theta(k + k\log(n/k))$ time, if the algorithm is allowed to make errors (with small probability) and k grows slower than any polynomial in n [7], or
- in $O(\log n)$ time, if k is fixed [13].

Under our definition of a nonadaptive algorithm, the queries are generated with no dependence on feedback. However, as described above, a station must adapt to the feedback produced when it actually transmits. One can study even more restricted algorithms, where transmissions are not influenced by any feedback whatsoever. Then the problem is to produce, given k and n $(2 \le k \le n)$, the shortest sequence of queries such that the following holds: For every possible set I of k conflicting stations the query sequence is such that, for all x in I, the sequence contains a query Q with $I \cap Q = \{x\}$. Formally, this is the problem of constructing minimum length $(k-1, n, 1)$ superimposed codes [1], [5], [14]. An interesting result of Bassalygo [5] is that serially querying individual stations is optimal when $k \ge \sqrt{2n} + 1$. A straightforward counting argument shows that $O(k^2 + k^2\log(n/k))$ queries suffice, for all k and n.

II. A Fast Nonadaptive Algorithm

Let $[n]$ denote the set of station identifiers $\{1, 2, \cdots, n\}$. We refer to the set of k conflicting stations as the *input* I ($|I| = k$ and $I \subseteq [n]$). Consider a list of queries $Q_1, Q_2, \cdots, Q_t$, where each $Q_j \subseteq [n]$. We define a corresponding list $I_1, I_2, \cdots, I_t$, where each $I_j \subseteq I_0 = I$, as follows:

$$\begin{aligned} I_j &= I_{j-1} - Q_j, \qquad \text{if } |Q_j \cap I_{j-1}| = 1 \\ &= I_{j-1}, \qquad \text{otherwise.} \end{aligned}$$

We allow $Q_1, Q_2, \cdots, Q_t$ to depend on k and n, but not on the choice of I. When $Q_j \cap I_{j-1} = \{x\}$, we say Q_j isolates x. The list of queries is said to be (α, m, n) *universal* if, for every input I with $|I| \le m$, the number of members of I that the queries fail to isolate is at most $(1-\alpha)m$, where $0 \le \alpha \le 1$ and $0 \le m \le n$.

Our goal is to prove that for all k and n there is a $(1, k, n)$ universal list $Q_1, Q_2, \cdots, Q_t$ of length $t = \Theta(k + k\log(n/k))$. Such a list represents a nonadaptive algorithm in which query Q_i is generated at the ith step of execution. A particular station, say station x, will transmit with success at that step j where Q_j isolates x.

For example, suppose $k = 2$, and the list $Q_1, Q_2, \cdots, Q_m$ is $(1, 2, n)$ universal. How large must m be? The $m \times n$ matrix whose ith row is the incidence vector of Q_i must have distinct columns. (If the two stations indexed by matching columns are the active ones, then the two always collide.) Thus, we must have $n \le 2^m$, i.e., $m = \lceil \log_2 n \rceil + 1$ and no fewer queries suffice if $k = 2$.

Our plan is as follows. We consider a list of $k/2$ queries chosen randomly, and prove that, with overwhelming probability, the list isolates at least a constant fraction of any input of size k (Lemma 2). This leads to a proof that (c, k, n) universal lists of length $\Theta(k + k\log(n/k))$ must exist for some constant c $(0 < c < 1)$ (Theorem 1). It is then a simple matter to build $(1, k, n)$ universal lists from the (c, k, n) universal lists (Theorem 2).

At this point, let us suppose that k divides n. We will drop this supposition when we get to Theorems 1 and 2. Consider the list $Q_1, Q_2, \cdots, Q_p$, where $p = \max\{1, \lfloor k/2 \rfloor\}$ and each query Q_i is of size n/k and is chosen uniformly at random from the $\binom{n}{n/k}$ possibilities.

Lemma 1: For all j $(1 \le j \le p)$, Q_j isolates some member of the input with probability greater than $1/2e^2$, no matter what the result of $Q_1, Q_2, \cdots, Q_{j-1}$.

Proof: The lemma holds trivially if $k = n$ or $k = 1$. Suppose $2 \le k < n$, which means $2 \le k \le n/2$, as k divides n. Consider any Q_j with $1 \le j \le p$. Let x denote the size of I_{j-1}, so $k - p < x \le k$. For given x, the probability that Q_j isolates a member of I_{j-1} is

$$\frac{x\binom{n-x}{n/k-1}}{\binom{n}{n/k}} = \frac{x}{k}\,\frac{(n-n/k)(n-n/k-1)\cdots(n-n/k-x+2)}{(n-1)(n-2)\cdots(n-x+1)}$$
$$= \frac{x}{k}\left(1 - \frac{n/k-1}{n-1}\right)\left(1 - \frac{n/k-1}{n-2}\right)\cdots \left(1 - \frac{n/k-1}{n-x+1}\right).$$

But $(n/k - 1)/(n - i) \le 1/2$ for all i with $1 \le i \le x - 1$. Since $1 - u > \exp(-2u)$ if $0 < u \le 1/2$,

$$\frac{x\binom{n-x}{n/k-1}}{\binom{n}{n/k}} > \frac{x}{k}\exp\left(-2(n/k-1)\cdot\left(\frac{1}{n-1} + \frac{1}{n-2} + \cdots + \frac{1}{n-x+1}\right)\right).$$

Note that

$$\log\frac{n}{n-x} = \int_{n-x}^{n}\frac{1}{u}\,du > \frac{1}{n-1} + \frac{1}{n-2} + \cdots + \frac{1}{n-x+1},$$

so

$$\frac{x\binom{n-x}{n/k-1}}{\binom{n}{n/k}} > \frac{x}{k}\exp\left(-2(n/k-1)\log\left(\frac{n}{n-x}\right)\right) = \frac{x}{k}\left(1 - \frac{x}{n}\right)^{(n/k-1)2} > \frac{1}{2}\left(1 - \frac{k}{n}\right)^{(n/k-1)2} > \frac{1}{2e^2},$$

as was to be proved.

Lemma 2: $Q_1, Q_2, \cdots, Q_p$ isolate at least ck members of the input, with probability greater than $1 - 1/e^{bk}$, where c and b are constants $(0 < c < 1,\ b > 0)$.

Proof: Let $B(m, M, f)$ denote the probability of m or more failures in M independent trials, each of which fails with probability f:

$$B(m, M, f) = \sum_{m \le i \le M}\binom{M}{i} f^i (1-f)^{M-i}.$$

Let us consider Q_i a success if it isolates a member of the input and a failure otherwise. Let $f = 1 - 1/(2e^2)$. By Lemma 1, each Q_i fails with probability less than f, irrespective of the success or failure of any previous query. Hence

$$\Pr\{m \text{ or more } Q_i \text{ fail}\} \le B(m, M, f).$$

The list $Q_1, Q_2, \cdots, Q_p$ isolates at least $M - m$ members of I, that is, at least $M - m$ of the Q_i succeed, with probability at least

$$1 - \Pr\{m \text{ or more } Q_i \text{ fail}\} \ge 1 - B(m, M, f).$$

We want to find constants c and b $(0 < c < 1,\ b > 0)$ such that, when $M - m \ge ck$, the right hand side of the last inequality is at least $1 - 1/e^{bk}$.

It follows from a result of Chernoff [6, p. 17] that, for any constants $\beta\,(0 < \beta < 1)$ and $f\,(0 < f < 1)$,

$$B((1+\beta)Mf, M, f) \le 1/e^{(\beta^2 Mf/3)}.$$

(The probability is extremely small, $1/e^{\Omega(M)}$, that the number of failures is more than a constant factor greater than Mf, the expected number.) To complete the proof, let $M = p$ (for concreteness), $\beta = 1/(2e^2)$, and $m = (1 + \beta)pf$, so that $M - m = p/(4e^4) \ge kc$ and

$$1/e^{(\beta^2 Mf/3)} < 1/e^{bk},$$

where $c = 1/9e^4$ and $b = \beta^2 f/12$. Combining these inequalities gives the lemma.

The next order of business is to convert the lemma about probabilities into a theorem about certainties.

Theorem 1: For all k and n $(2 \le k \le n)$, there is a (c, k, n) universal list of length $\Theta(k + k\log(n/k))$, where c is the constant of Lemma 2.

Proof: Suppose, for the moment, that k divides n. Consider a list $Q_1, Q_2, \cdots, Q_t$ of t queries, each of size n/k and each chosen uniformly at random from the $\binom{n}{n/k}$ possibilities. Associate with each input I of size k the random variable

$$X_I = \begin{cases} 0, & \text{if } Q_1, Q_2, \cdots, Q_t \text{ leaves } \le (1-c)k \\ & \text{unisolated members of } I \\ 1, & \text{otherwise} \end{cases}$$

and let $X = \sum X_I$, summing over all such I. (X is the number of inputs I such that the queries fail to isolate at least ck members of I.) Suppose t is so large that the expected value of X_I, $E\{X_I\}$, is less than $1/\binom{n}{k}$ for every I. Then $E\{X\} = \Sigma E\{X_I\}$ is less than 1, which means that there is a query list $Q_1, Q_2, \cdots, Q_t$ under which $X_I = 0$ for every I, that is, $Q_1, Q_2, \cdots, Q_t$ is (c, k, n) universal. We need only show that, for $t = \Theta(k + k\log(n/k))$, we can ensure that $E\{X_I\} = \Pr(X_I = 1) < 1/\binom{n}{k}$.

Suppose the number of random queries $t = mp$, where $p = \max\{1, \lfloor k/2 \rfloor\}$. Suppose, for the moment, that we restore the input just after steps $p, 2p, 3p, \cdots, (m-1)p$. This means the query list $Q_{jp+1}, Q_{jp+2}, \cdots, Q_{(j+1)p}$ acts on the original input I (for each j, $0 \le j \le m$), instead of the reduced input I_{jp}. By Lemma 2, with probability less than e^{-bk}, $Q_{jp+1}, \cdots, Q_{(j+1)p}$ leaves more than $(1-c)k$ unisolated members of the input. Since the queries are chosen independently, all m groups leave more than $(1-c)k$ unisolated members of the input with probability less than e^{-bmk}. Now, drop the supposition that the input is restored after steps $p, 2p, \cdots, (m-1)p$. This can only lessen the aforementioned probability, so $\Pr(X_I = 1) < e^{-bmk}$. In order to make e^{-bmk} less than $1/\binom{n}{k}$, put

$$m = \left\lceil \left(\ln\binom{n}{k}\right) \Big/ (bk) \right\rceil + 1.$$

As a result, the list length $t = mp$ is $\Theta(k + k\log(n/k))$.

We now consider the case where k does not divide n. Add up to k dummy stations to get a total of $n' \le n + k$ stations, where k divides n'. Using the argument given above, produce a (c, k, n') universal list of length $\Theta(k + k\log(n'/k)) = \Theta(k + k\log(n/k))$, and strike the dummy stations from each query to obtain a (c, k, n) universal list of the desired length. This proves the theorem.

We now put together $(1, k, n)$ universal lists using the (c, k, n) universal lists of Theorem 1 as building blocks.

Theorem 2: For all k and n $(2 \le k \le n)$, there is a $(1, k, n)$ universal list of length $\Theta(k + k\log(n/k))$.

Proof: Theorem 1 guarantees the existence of (c, k, n) universal lists of length $\Theta(k + k\log(n/k))$, where c $(0 < c < 1)$ is the constant of Lemma 2. Let

$$p = \lceil (\log k)/(\log 1/(1-c)) \rceil + 1.$$

Apply Theorem 1 to produce p query lists $L_0, L_1, \cdots, L_{p-1}$ such that L_i is $(c, (1-c)^i k, n)$ universal and has length

$$\Theta\left(k(1-c)^i + k(1-c)^i \log\left(\frac{n}{k}(1-c)^{-i}\right)\right).$$

Concatenate the L_i to form a list $L = L_0 L_1 \cdots L_{p-1}$, which has length $\Theta(k + k\log(n/k))$. Notice that L_0 fails to isolate at most $(1-c)k$ members of any given input I. $L_0 L_1$ fails to isolate at most $(1-c)^2 k$ members of I. $L_0 L_1 L_2$ fails to isolate at most $(1-c)^3 k$ members of I, and so forth. Thus, L fails to isolate at most $(1-c)^p k$ members of I and, by the definition of p,

$$(1-c)^p k \le (1-c) < 1.$$

Thus, L successfully isolates all k members of I, meaning L is $(1, k, n)$ universal, as was to be shown.

III. Directions for Further Research

An adaptive conflict resolution algorithm is allowed to determine each query as a function of the feedback from its previous queries, where feedback indicates whether the query elicited 0, 1, or ≥ 2 transmission attempts. With respect to worst case running time, the fastest adaptive algorithm reported so far (the tree algorithm [2], [3], [11], [17]) takes time $\Theta(k + k\log(n/k))$ to resolve a conflict among k stations, for any k and n $(2 \le k \le n)$. In this paper, we gave a nonconstructive proof that there is a nonadaptive algorithm that meets the same time bound. Our counting methods were not refined enough to make it worthwhile to optimize the constants implicit in the $\Theta(k + k\log(n/k))$ bound. It would be of interest to do a tighter analysis aimed at getting small constants. It would also be of interest to get a constructive proof, one that produces an algorithm instead of verifying its existence.

We note that a simple extension of our argument indicates that, for some constant $c > 0$, a random list of $c(k + k\log(n/k))$ queries resolves the conflict with overwhelming probability. Thus, there is a particularly simple, constructive probabilistic algorithm for resolving conflicts whose queries are chosen independent of feedback.

All deterministic algorithms for conflict resolution, adaptive or nonadaptive, must have worst case running time $\Omega(k(\log n)/(\log k))$ [10]. It would be of interest to narrow the gap between this lower bound and the order $k + k\log(n/k)$ upper bound, at least for nonadaptive algorithms.

Acknowledgment

We are grateful to Philippe Jacquet for finding an error in the manuscript, and to a referee for pointing out the connection with superimposed coding and for several helpful suggestions for improving the exposition.

References

[1] L. A. Bassalygo and M. S. Pinsker, "Restricted asynchronous multiple access," *Problemy Peredachi Informatsii*, vol. 9, no. 4, pp. 92–96, 1983.
[2] J. Capetanakis, "Generalized TDMA: The multi-accessing tree protocol," *IEEE Trans. Commun.*, vol. COM-27, pp. 1479–1484, Oct. 1979.
[3] ——, "Tree algorithms for packet broadcast channels," *IEEE Trans. Inform. Theory*, vol. IT-25, pp. 505–515, Sept. 1979.
[4] Digital-Intel-Xerox, "The ethernet data link layer and physical layer specifications 1.0," Sept. 1980.
[5] A. G. Dyachov and V. V. Rykov, "A survey of superimposed code theory," *Probs. Contr. Intro. Th.*, vol. 12, no. 4, pp. 1–33, 1983.
[6] P. Erdös and J. Spencer, *Probabilistic Methods in Combinatorics*. New York: Academic, 1974.
[7] V. Freidlina, "On a design problem for screening experiments," *Theory Prob. Appl.*, vol. 20, no. 1, pp. 102–115, 1975.
[8] A. G. Greenberg and R. E. Ladner, "Estimating the multiplicities of conflicts in multiple access channels," in *Proc. 24th Ann. Symp. Foundations of Computer Science*, Tucson, AZ, pp. 383–392, 1983.

[9] A. G. Greenberg, "Efficient algorithms for multiple access channels," Ph.D. dissertation, Univ. of Washington, Seattle, 1983.

[10] A. G. Greenberg and S. Winograd, "A lower bound on the time needed to resolve conflicts deterministically in multiple access channels," *J. Assoc. Comput. Mach.*, vol. 32, no. 3, July 1985.

[11] J. F. Hayes, "An adaptive technique for local distribution," *IEEE Trans. Commun.*, vol. COM-26, pp. 1178–1186, Aug. 1978.

[12] F. Hwang, "A method for detecting all defective members in a population by group testing," *J. Amer. Statist. Assoc.*, vol. 67, pp. 605–608, 1972.

[13] F. Hwang and V. Sós, "Nonadaptive group testing procedures," *Studia Math. Hungarica*, to be published.

[14] W. H. Kautz and R. C. Singleton, "Nonrandom binary superimposed codes," *IEEE Trans. Inform. Theory*, vol. IT-10, pp. 363–377, Oct. 1964.

[15] J. L. Massey, "Collision-resolution algorithms and random-access communications," in *Multi-User Communication Systems*, G. Longo, Ed. (CISM Courses and Lectures No. 265). New York: Springer-Verlag, 1981.

[16] R. Metcalfe and D. Boggs, "Ethernet: distributed packet switching for local computer networks," *Commun. Assoc. Comput. Mach.*, vol. 19, pp. 395–404, July 1976.

[17] B. S. Tsybakov and V. A. Mikhailov, "Free synchronous packet access in a broadcast channel with feedback," *Prob. Inform. Trans.*, vol. 14, no. 4, pp. 259–280, April 1979. (Translated from Russian original in *Problemy Peredachi Informatsii*, vol. 16, no. 3 (Oct.–Dec. 1978), pp. 32–59.)