Estimating the multiplicities of conflicts in multiple access channels

Albert Greenberg, Philippe Flajolet, Richard E. Ladner

IRIA
CENTRE DE ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tel (3) 954 90 20

Rapports de Recherche

N° 333

ESTIMATING THE MULTIPLICITIES OF CONFLICTS IN MULTIPLE ACCESS CHANNELS

Albert GREENBERG
Philippe FLAJOLET
Richard E. LADNER

Septembre 1984

ESTIMATING THE MULTIPLICITIES OF CONFLICTS IN MULTIPLE ACCESS CHANNELS

A. Greenberg, P. Flajolet, R. Ladner

Resumé: Cet article propose de nouveaux algorithmes permettant de régler l'accès à un canal à accès multiple partagé par des stations géographiquement dispersées. On présente d'abord des algorithmes qui permettent aux stations entrant en conflit d'estimer stochastiquement de manière coopérative et distribuée la multiplicité du conflit. La combinaison de nos méthodes d'estimation avec un protocole en arbre conduit à des algorithmes hybrides de résolution de conflits sur le canal. La combinaison la plus efficace conduit à une méthode 20% plus rapide que les autres algorithmes comparables connus.

Abstract: We propose new, improved algorithms for regulating access to a multiple access channel, a common channel shared by many geographically distributed computing stations. We first present and analyze algorithms that allow the conflicting stations to cooperatively compute a stochastic estimate of the multiplicity of conflict. Combining one of our estimation algorithms with a tree algorithm then leads to a hybrid algorithm for conflict resolution. Several efficient combinations are possible, the most efficient of which resolves conflicts about 20% faster than any of the comparable algorithms reported to date.

ESTIMATING THE MULTIPLICITIES OF CONFLICTS IN MULTIPLE ACCESS CHANNELS

Albert G. Greenberg

AT&T Bell Laboratories
600 Mountain Avenue
Murray Hill, NJ 07974

Philippe Flajolet

I.N.R.I.A.
78150 - Rocquencourt, France

Richard E. Ladner

Department of Computer Science FR-35
University of Washington
Seattle, WA 98195

May, 1984

ABSTRACT

We propose new, improved algorithms for regulating access to a multiple access channel, a common channel shared by many geographically distributed computing stations. A conflict of *multiplicity* n occurs when n stations transmit simultaneously to the channel. As a result, all stations receive feedback indicating whether n is 0, 1, or ≥ 2. If $n = 1$ the transmission succeeds, whereas if $n \geq 2$ all the transmissions fail. We first present and analyze algorithms that allow the conflicting stations to cooperatively compute a stochastic estimate n^* of n, at small cost, as a function of the feedback elicited during its execution. An algorithm to *resolve* a conflict among two or more stations controls the retransmissions of the conflicting stations so that each eventually transmits singly to the channel. Combining one of our estimation algorithms with a tree algorithm (of Capetanakis, Hayes, and Tsybakov and Mikhailov) then leads to a hybrid algorithm for conflict resolution. Several efficient combinations are possible, the most efficient of which resolves conflicts about 20% faster on average than any of the comparable algorithms reported to date.

1. 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 either co-axial cable, fiber optic, packet radio, or satellite transmission media. A well known example is the Ethernet [5,22], which uses a co-axial cable of up to 1.5 km in length to connect stations numbering up to 1024.

We consider the following model commonly taken as the basis of mathematical studies of the multiple access channel [2-3,6-8,12-18,21,27]. At *steps* numbered 1, 2, 3,... any station can transmit a packet of data to the channel. There is no central control. If n stations transmit simultaneously, the result depends on n as follows:

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

Prior to the next step all stations receive the feedback 0, 1, or 2+ indicating whether the *multiplicity of conflict* n is 0, 1, or ≥ 2, respectively. In general no *a priori* knowledge of n is available, such as its probability distribution.

CONFLICT RESOLUTION ALGORITHMS

In the late 1970's, Capetanakis [2,3], Hayes [16], and Tsybakov and Mikhailov [27] independently introduced novel schemes for coordinating access to the channel, which have some remarkable properties either not known to exist or provably absent in earlier proposals (see e.g. reference [21]). At the heart of these schemes is an algorithm to solve the problem of *conflict resolution*, which is the main problem this paper addresses. This

problem is closely related to the problem of supporting a high rate of packet arrivals, as described below.

An algorithm to *resolve* a conflict schedules retransmissions so that each of the conflicting stations eventually transmits singly to the channel. Just the n stations involved in the initial conflict participate. However, all stations monitoring the channel are able to detect the algorithm's termination on the basis of channel feedback. Capetanakis and Tsybakov and Mikhailov proposed probabilistic *tree algorithms* for conflict resolution, whose analysis and refinement have received considerable research attention [6-8,17,21]. Using expected running time (measured as the number of steps used) as a measure of merit, the best of these is the modified, biased tree algorithm, which Fayolle and Hofri [6,17] show has expected running time approximately 2.662 n for large n.

Our main innovation is a probabilistic algorithm that at small cost produces a stochastic estimate of n, the conflict multiplicity. We construct a *hybrid algorithm* to resolve conflicts, using as subroutines

- an estimation algorithm (there are several versions of the algorithm), and
- a binary tree algorithm (again, there are several versions).

The details of the construction depend on results obtained from the precise asymptotic analysis of the two component algorithms. With expected running time as the measure of merit, the best hybrid algorithm resolves conflicts of multiplicity n in expected time approximately 2.134 n for large n.

We present our results in two stages. We first give simple versions of the algorithms, and a precise asymptotic analysis of their behavior. We then present improved versions of the algorithms, and explain how to adapt the analysis to capture the improvements. The key technical device at work in the analysis is the Mellin integral transform, which allows

us to obtain precise asymptotic results with relatively little effort.

Our first version of the estimation algorithm controls transmission to the channel using coin tosses with bias 2^{-i} for integer i. We call this the *base 2 estimation algorithm*. In expected time $\log_2 n + O(1)$ it produces an estimate of n with mean approximately $0.914\,n$ and standard deviation approximately $0.630\,n$ (Theorems 2 and 3). We consider a generalization to an arbitrary numerical base $a > 1$ and, in particular, consider choosing a close to 1. Varying a gives a certain time/accuracy trade-off. The base a algorithm has expected running time $\log_a n + O(1)$. Asymptotically, that is, for a close to 1 and n large, the algorithm produces a sharp, unbiased estimate of n in time that remains $0(\log n)$ (Theorems 8 and 9).

As mentioned above, the hybrid algorithm uses an estimation algorithm and a binary tree algorithm as subroutines. Our first version of the algorithm uses

1. the base 2 estimation algorithm, and
2. the simple tree algorithm of Capetanakis [2,3] and Tsybakov and Mikhailov [27].

We show that this algorithm resolves conflicts of multiplicity n in expected time approximately $2.490\,n$, for large n (Theorem 5). We later generalize the algorithm to use

1. the base a estimation algorithm, in conjunction with
2. one of the following tree algorithms
   - the simple tree algorithm,
   - the modified tree algorithm of Massey [2,3,21] and Tsybakov and Mikhailov [27], or
   - the modified, biased tree algorithm of Fayolle and Hofri [6,17].

For each of the three choices of the underlying tree algorithm, our results indicate that the smaller a the better the asymptotic (large n) performance of the hybrid algorithm. Using the base $1 + 10^{-4}$ estimation algorithm, the expected running time of the hybrid algorithm is $cn + O(n^{\epsilon})$, where $0 < \epsilon < 1$ and apart from negligible fluctuations c is as described in

Table 1. (This data comes from evaluating formulas presented in section 5.2.)

TABLE 1. Expected Running Time of the Hybrid Algorithm

tree algorithm	c
simple tree	2.3356
modified tree	2.1699
modified, biased tree	2.1403

As the base a goes to 1 the expected running time of the hybrid algorithm goes to that of a certain idealized algorithm, which *obtains at no cost* the exact value of n instead of an estimate.

APPLICATIONS OF CONFLICT RESOLUTION ALGORITHMS

Conflict resolution algorithms are the building blocks of the following *channel access protocols* proposed by Capetanakis [2,3], Hayes [16], and Tsybakov and Mikhailov [27]. Initially access to the channel is unrestricted. Thereafter, channel activity alternates between intervals in which access is unrestricted and intervals in which access is restricted to resolve conflicts. During an unrestricted interval a station holding a packet simply transmits it without delay. When a collision arises the conflicting stations execute an algorithm to resolve the conflict, while all other stations defer. When the algorithm terminates access is again unrestricted.

The behavior of such protocols is typically studied under the assumption that new packets arise from an infinite population of stations according to a Poisson process at a fixed rate of λ packets per step ($0 < \lambda < 1$). Following Fayolle and Hofri [5,15], we say the protocol is *stable* if the expected length of each conflict resolution interval is uniformly bounded. On general grounds, one can show that if a conflict resolution algorithm has

expected running time $T(n)$, then the corresponding channel access protocol is stable for all $\lambda < \liminf \frac{n}{T(n)}$ and is unstable for all $\lambda > \limsup \frac{n}{T(n)}$ [6,17,21]. It follows from our analysis of $T(n)$ for the hybrid algorithm that

- the hybrid algorithm using base 2 estimation and the simple tree algorithm is stable for all λ up to 0.4025, and
- the hybrid algorithm using base a estimation for a close to 1 and the modified, biased tree algorithm is stable for all λ up to 0.4686.

Channel access protocols are described in more detail in section 6, where we also discuss measures of the stability and adaptivity of the protocols.

In addition, algorithms to resolve conflicts are needed to implement certain network protocols for distributed computations, as described in [12,13]. These protocols proceed in rounds of information exchange, where each round gives rise to a burst of packets potentially leading to a conflict, which must be resolved before the next round can begin. The time efficiency of the protocol depends in part on that of the underlying algorithm for resolving conflicts.

DETERMINISTIC CONFLICT RESOLUTION ALGORITHMS

In this paper we concentrate on probabilistic algorithms for conflict resolution. However, it is possible to define deterministic algorithms if each station is uniquely identified by a whole number between 1 and N, where N is the total number of stations, and the algorithm can use the station numbers. At each step, a set of stations called the query set is enabled to transmit, chosen as a function of channel feedback. A measure of merit is the time needed in the worst case to resolve a conflict among n stations (worst case refers to the maximum over all subsets of size n), measured as the number of steps used. Capetanakis [2,3], Hayes [16], and Tsybakov and Mikhailov [27] independently proposed a deterministic tree algorithm, which uses channel feedback to guide a search for the n

conflicting stations. This algorithm runs in time $\Theta(n + n\log \frac{N}{n})$ in the worst case for all n and N, $(2 \leq n \leq N)$. We know of no algorithm attaining an asymptotically smaller worst case running time. If n is given *a priori* then the $O(n + n\log \frac{N}{n})$ bound can be achieved non-adaptively, that is, by a sequence of query sets determined as a function of n and N, independent of channel feedback [18]. No deterministic algorithm (adaptive or non-adaptive) can do substantially better: With respect to a general model of deterministic algorithms to resolve conflicts, $\Omega(n(\log N)/(\log n))$ time is needed in the worst case to resolve a conflict of multiplicity n, for all n and N $(2 \leq n \leq N)$ [14]. It can be shown further that the worst case running time of every divide and conquer algorithm, which includes the deterministic tree algorithm, is $\Omega(n + n\log \frac{N}{n})$ [13]. These lower bounds on deterministic algorithms prove that the probabilistic algorithms we consider, which run in expected time linear in the multiplicity n for any n, are better than deterministic algorithms, whose running time is a function of n and N. (When $n = 2$, $\Omega(\log N)$ time is required.)

OUTLINE

In section 2 we first describe and motivate the model of a multiple access channel. In subsections 2.1 and 2.2, the simple versions of the estimation and hybrid algorithms are presented. In section 3, we present a detailed analysis of the first two moments of n^*, the estimate of n provided by the simple estimation algorithm. In section 4, we present a comparable analysis of the first moment of the running time of the simple hybrid algorithm. In section 5, we present the improved versions of the algorithms, and the corresponding analyses in subsections 5.1 and 5.2. Finally, in section 6 we address the stability and adaptivity of channel access protocols based on the hybrid and related conflict resolution algorithms.

Our analyses produce tight asymptotic (large n) bounds. At appropriate points, we include corroborative results obtained from simulations. Preliminary results of the research reported here appeared in references [13] and [15].

2. ALGORITHMS FOR A TIME SLOTTED CHANNEL

Underlying the formal model of a multiple access channel given in the Introduction, is the more general model of a *time slotted channel* [26]. Figure 1 depicts an example of the time line with slots demarcated, and each labeled with 0, 1, or 2+ according to whether the slot corresponds to the simultaneous transmission of 0, 1, or $\geq$ 2 stations. It is assumed that stations can detect slot boundaries, and are synchronized so that transmissions are initiated only at slot boundaries.

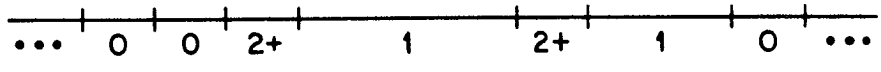


Figure 1. Example illustrating activity on the channel over 7 slots.

The length of an empty slot (marked 0) depends on the physical characteristics of the system such as the maximum transmission propagation delay between stations. The length of a slot corresponding to a collision (marked 2+) depends on related physical characteristics of the system. In local networks, stations involved in a collision rapidly receive feedback indicating this, and can abort the transmissions after having transmitted only a small fraction of a packet. The length of a slot corresponding to a successful transmission (marked 1) may depend on the length of message carried in the packet.

In local networks, there is a trend towards driving the channel at higher and higher bit rates. (Transmission on the prototype Ethernet proceeded at 3 million bits/second [20].

On a more recent version, transmission proceeds at 10 million bits/second [5]. Plans for constructing related networks supporting rates of 200 million bits/second have been reported [20].) As this trend continues, the difference in the lengths of the three types of slots decreases, as about the same time is needed to transmit a packet, sense the channel idle, or sense a collision. Thus, our formal model of the channel, which accounts for slots as equal cost *steps*, grows closer to the real situation along with this trend. Also, the analytic methods presented here can be easily extended to cover cases where collision slots, empty slots, and transmission slots have different (but fixed) durations.

Although analyses of multiple access channels nearly universally start with the assumption that time is slotted, achieving the required synchronization is a non-trivial engineering problem. Time slotted satellite channels have been built, such as the common signaling channel used in the Comsat Intelsat Spade system [26]. In channels for local networks, like the Ethernet [5] and the Hyperchannel [10], synchronization may be partial. A tone can be put out on the channel to aid synchronization [20]. An apparent difficulty is in coping with local clock drift during a long interval of consecutive empty slots, say, one of length $\Omega(N)$, where N is the total number of stations. A nice feature of the fast probabilistic algorithms described here is that such long intervals of consecutive empty slots are extremely unlikely during a conflict resolution interval. In the Ethernet, stations use similar clocks and use the notion of a slot to schedule retransmissions, but need not agree on slot boundaries. In the Hyperchannel, stations must agree on slot boundaries during the interval of time following a collision up to a successful transmission. ($N-1$ empty slots may elapse during this interval).

Recently, Molle [23] proposed a way to map low level algorithms designed under a time slotted model (like those presented here) into algorithms that work under a model in which

stations have similar clocks but need not agree on slot boundaries (like the Ethernet). Roughly speaking, his simulation results indicate that efficient algorithms under the time slotted model map into efficient algorithms in the unslotted setting. These results are encouraging in that they show that the rich theory of the slotted channel may lead to more efficient use of channels like the Ethernet without major changes in their engineering.

2.1 A SIMPLE ALGORITHM FOR ESTIMATING THE MULTIPLICITY CONFLICT

Suppose that n stations simultaneously transmit to the channel, and that $n \geq 2$. We wish to compute a random function n^* of n whose value gives an indication of the value of n itself, while not using many steps in doing so.

Our strategy is to search for a power of 2 that is close to n with high probability. Suppose for the sake of argument there is some evidence that $n \geq 2^i$ for some integer $i > 0$. A probabilistic test of the hypothesis that $n > 2^{i+1}$ can be arranged as follows. Have each of the n conflicting stations either transmit or not transmit in accordance with whether the outcome of a toss of a biased binary coin is 0 or 1. The coin is biased to turn up 0 with probability $2^{-(i+1)}$ and 1 with the complementary probability. Since the expected number of transmitters is $2^{-(i+1)}n$, feedback 2+ supports the hypothesis that $n > 2^{i+1}$. Figure 2 describes the *base 2 estimation algorithm* motivated by this observation. Each of the conflicting stations executes the algorithm, resulting in a string of collisions whose length determines n^*. We assume that the random decisions involved are independent. In this way, n^* is computed in time $1 + \log_2 n^*$.

Figure 2. Base 2 Estimation Algorithm

```
i := 0
repeat
    i := i+1
    With probability 2^{-i}, transmit to the channel
until no collision occurs
n* := 2^i
```

Consider the example given by $n = 2$. At the first step of the loop, each station transmits with probability 1/2. As a result, with probability $(1 - 1/2)^2 + 2(1 - 1/2)(1/2) = 3/4$, zero or one stations transmit, causing the algorithm to halt, and yielding $n^* = 2$. With the complementary probability, 1/4, both stations transmit and the algorithm proceeds to the second step of the loop. At that step, each station transmits with probability 1/4. Thus, with probability $(1 - 1/4)^2 + 2(1 - 1/4)(1/4) = 15/16$, zero or one stations transmit, causing the algorithm to halt, and yielding $n^* = 4$. With the complementary probability, 1/16, the algorithm proceeds to the third step of the loop, and so forth.

As i increases, the probability that at most one station transmits increases monotonically, and approaches 1 extremely rapidly as i increases past $\log_2 n$. As a result, one would suspect that the algorithm proceeds until i is close to $\log_2 n$ and then terminates producing n^* close to n. It is not hard to confirm this suspicion in the sense that the $m-th$ moment of $\log_2 n^*$ is $\Theta(\log_2^m n)$ and the $m-th$ moment of n^* is $\Theta(n^m)$. We can pinpoint the values of these moments much more precisely. Of particular interest are the first two moments of n^*, $E(n^*)$ and $E((n^*)^2)$:

$$E(n^*) = n(\phi + U(\log_2 n)) + 0(n^{\epsilon}),$$

$$E((n^*)^2) = n^2(\Phi + V(\log_2 n)) + 0(n^{1+\epsilon}),$$

where

$$\phi = \frac{1}{\log 2} \int_0^\infty e^{-x}(1+x) \prod_{k=1}^{\infty} (1 - e^{-2^k x} (1 + 2^k x)) / x^2 \, dx = 0.91422...,$$

$$\Phi = \frac{1}{\log 2} \int_0^\infty e^{-x}(1+x) \prod_{k=1}^{\infty} (1 - e^{-2^k x} (1 + 2^k x)) / x^3 \, dx = 1.23278 ...,$$

U and V are periodic functions having mean 0 and very small amplitude, and ϵ is an arbitrary constant with $0 < \epsilon < 1$. The proofs of these results are given in the next section, where in passing we present precise asymptotic information about the distribution of n^*. The two integrals were evaluated using double precision numerical quadrature routines from the Port software library (a licensed product of AT&T Bell Laboratories).

The fact that for large n, $E(n^*) \approx \phi n$, suggests treating $n^+ = \frac{n^*}{\phi}$ as an estimate of n. Owing to the contribution of the periodic function U, n^+ is not an unbiased estimate of n, in the sense that $\frac{E(n^+)}{n}$ is not 1. Fortunately, the amplitude of $U(z)$ turns out to be less than $2 \cdot 10^{-5}$, so this bias is negligible for all practical purposes.

Interestingly, a simple variant of the estimation algorithm has provably disastrous performance. Consider the algorithm where each station involved in the initial collision transmits to the channel with probability 1/2. If this causes another collision then those that just transmitted again transmit with probability 1/2. The others drop out. This continues, with the stations continuing to try to transmit always being a subset of those that just transmitted, until there is no collision. Take 2^i as the estimate of the multiplicity of conflict where i is number of the steps preceding the first step at which there is no collision. It can be shown that the second and all higher moments of this estimate are infinite.

2.2 A SIMPLE HYBRID ALGORITHM FOR CONFLICT RESOLUTION

We begin this section with a review of the simple tree algorithm for conflict resolution. We then discuss a way to improve it, due to Capetanakis, for a special case in which the stations have *a priori* information specifying a particular probability distribution for the multiplicity of conflict n [2,3]. Lastly we define the hybrid algorithm for conflict resolution, which achieves a related improvement, without using *a priori* information about n.

Following Massey [21], we present the simple tree algorithm in a recursive fashion. Suppose that n stations transmit simultaneously to the channel. If the resulting feedback indicates n is 0 or 1 then the conflict is already resolved. Otherwise, each of the conflicting stations tosses an unbiased binary coin, at which point:

1. Stations whose coins turned up 0 transmit to the channel, which results in a conflict that these stations now resolve.
2. Next, stations whose coins turned up 1 transmit to the channel, which results in a conflict that these stations now resolve.

Using a local counter c, each conflicting station can keep track of when it should transmit [21]. Until a station has transmitted successfully, the station checks c just before each step, and proceeds to transmit only if c equals 0. Prior to the next step, c is updated according to the following rules:

1. If feedback indicates a collision, then
   - if the station participated in the collision, c is set equal to outcome of an unbiased coin toss (0 or 1),
   - whereas if the station did not participate, c is set equal to $c + 1$.
2. If feedback indicates 0 or 1 station transmitted, then c is set equal to $c - 1$.

Table 2 summarizes the rules for updating c. By an analogous technique, stations not involved in the conflict resolution can detect its termination.

TABLE 2. Tree Algorithm Implementation (toss represents the outcome (0 or 1) of the toss of an unbiased binary coin).

	2+	1	0
$c = 0$	$c :=$ toss	success	(impossible)
$c \geq 0$	$c := c + 1$	$c := c - 1$	$c := c - 1$

An execution of the algorithm determines a binary tree in which internal vertices correspond to collisions and leaves to either successful transmissions or no transmissions. Figure 3 illustrates a possible execution on a conflict involving three stations, say A, B, and C, which we take to occur at step 1. All three stations independently toss coins after the initial conflict. Two stations, say A and B, obtain 0's and so collide at step 2. After a step of no transmissions the two collide again at the fourth step, and then transmit successfully at steps 5 and 6. Station C transmits successfully at step 7, which marks the completion of execution. Table 3 illustrates how the local variables c evolve in accordance with channel feedback.

The algorithm runs in expected time approximately 2.885 n, for large n [15].

Figure 3. Binary Tree Determined by Execution

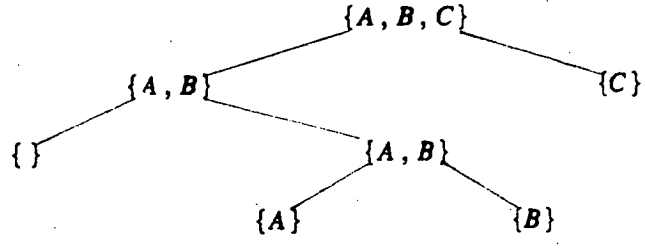


TABLE 3. Trace of Local Counters

Step	c for A	c for B	c for C	Feedback
1	0	0	0	2+
2	0	0	1	2+
3	1	1	2	0
4	0	0	1	2+
5	0	1	2	1
6	-	0	1	1
7	-	-	0	1

Capetanakis suggested the following improvement, motivated by the fact that the tree algorithm is most efficient for conflicts of small multiplicity [2,3]. Assume the initial conflict has multiplicity $n \geq 2$. All stations involved pick a number in the range 1, 2, ..., m uniformly and independently at random, where m is a global parameter. This divides the conflicting stations into m groups, which are then processed serially using the tree algorithm: First, those that picked 1 transmit, and the resulting conflict is resolved using the tree algorithm. Second, those that picked 2 transmit, and that conflict is resolved using the tree algorithm, and so forth. Capetanakis defined a *dynamic tree protocol* exploiting this idea, under the assumption that the multiplicity of conflict is a Poisson distributed random variable whose mean λ is known *a priori*.

An appropriate choice of m is crucial. If $\frac{m}{n}$ is too small then the algorithm will behave essentially the same as the tree algorithm. If $\frac{m}{n}$ is too large then the algorithm will behave poorly because many of the m groups will be empty and steps will be wasted.

In section 5.2, we present a method for finding the best setting of $\frac{m}{n}$. It turns out that to minimize the expected running time $\frac{m}{n}$ should be close to 0.9. The fact that $E(n^*) \approx 0.9n$ suggests the following simple *hybrid algorithm* for conflict resolution, which works without any *a priori* information n.

Figure 4. Simple Hybrid Algorithm

1. Compute n^* using the simple estimation algorithm, and let $m = n^*$.
2. Divide the conflicting stations into m groups by having each pick a number uniformly at random between 1 and m. Process the m groups serially using the tree algorithm.

Note that the algorithm can be implemented from only the knowledge of channel feedback.

Later, we will generalize the algorithm to force $\frac{m}{n}$ closer to the optimal value, with high probability. A small detail involves handling the case in which the estimation phase ends with one station transmitting successfully. To ease the analysis we assume that nevertheless this station participates in the second phase. This increases the expected running time slightly over the case where this station does not participate but does not affect the asymptotic results.

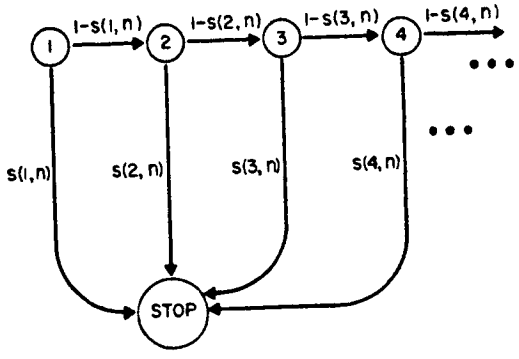
3. ANALYSIS OF THE SIMPLE ESTIMATION ALGORITHM

In this section we obtain precise asymptotic bounds on the first two moments of n^*, $E(n^*)$ and $E((n^*)^2)$. We present the analysis leading to a bound for $E(n^*)$ in detail. We give a brief outline of the corresponding analysis for $E((n^*)^2)$, because the details are much the same as those for $E(n^*)$. Higher moments of n^* can be analyzed in a similar manner. Two techniques are used over and over in the analyses: a certain exponential approximation, and an approach to the asymptotic of sums that involves the Mellin integral transform. Knuth and de Bruijn pioneered the application of these techniques, in treating

key properties of radix exchange sort and trie data structures ([19], pages 123-139, 481-505.) Knuth refers to this style of analysis as the gamma function approach, but this is indeed an application of Mellin transform techniques, similar to those found in analytic number theory.

The estimation algorithm corresponds to the transition diagram of Figure 5.

Figure 5. Transition Diagram for the Base 2 Estimation Algorithm



State i of the diagram corresponds to the $i-th$ iteration of the loop of Figure 2. With probability $1 - s(i,n)$ the algorithm reaches the next state (iteration), and with probability $s(i,n)$ the algorithm stops, yielding $n^* = 2^i$. Here $s(i,n)$ represents the probability of 0 or 1 transmissions when n stations toss coins with bias 2^{-i} to decide whether or not to transmit:

$$s(i,n) = (1 - 2^{-i})^n + (1 - 2^{-i})^{n-1} 2^{-i} n.$$

Let $q(i,n)$ denote the probability that the algorithm reaches state i, that is, $q(i,n) = Prob(n^* \geq 2^i)$. Then $q(1,n) = 1$, and if $i > 1$ then

$$q(i,n) = (1 - s(i-1,n))(1 - s(i-2,n)) \cdots (1 - s(1,n)),$$

so

$$E(n^*) = \sum_{i=1}^{\infty} (q(i,n) - q(i+1,n))\, 2^i \,. \qquad (1)$$

Our analysis hinges on the fact that this sum is closely approximated by a certain *harmonic sum*, that is, by a sum of the form $\sum_i b_i \Psi(a_i x)$ where Ψ is a real valued function of its argument, and the a_i and b_i are two sequences of real numbers. Harmonic sums arise often in the analysis of algorithms and, in particular, in the analysis of algorithms for a multiple access channel. An approach to the asymptotics of harmonic sums involving the Mellin integral transform often succeeds (cf. reference [9], pages 141-158). We will see that such an approach succeeds here.

The harmonic sum approximating $E(n^*)$ arises from the exponential approximation: $(1 - x)^n \approx e^{-x\,n}$ if $0 < x < 1$. Specifically, we use

$$e^{-2^{-i}n} + e^{-2^{-i}n}\, 2^{-i}\, n$$

as an approximation for

$$s(i,n) = (1 - 2^{-i})^n + (1 - 2^{-i})^{n-1}\, 2^{-i}\, n,$$

which leads to $\psi(\frac{n}{2^i})$ as approximation for $q(i+1,n)$, where

$$\psi(x) = \prod_{j=0}^{\infty} (1 - e^{-2^j x}\,(1 + 2^j x)).$$

(The infinite product converges for all $x \geq 0$, going rapidly to 0 as x goes to 0, and going rapidly to 1 as x goes to ∞. Notice that $\psi(0) = 0$.) The first i terms of the product for $\psi(\frac{n}{2^i})$ are the exponential counterparts to the i terms in the product for $q(i+1,n)$. The other terms are included to simplify the dependence on i though, being extremely close to 1, these terms have little influence. Substituting $\psi(\frac{n}{2^i})$ for $q(i+1,n)$ in equation (1) leads to

$$F(n) = \sum_{i=0}^{\infty} \Psi(\frac{n}{2^i})\, 2^i \qquad (2)$$

where

$$\Psi(x) = \psi(2x) - \psi(x) = e^{-x}(1 + x) \prod_{j=1}^{\infty} (1 - e^{-2^j x}\ (1 + 2^j x)).$$

Lemma 1: The expectation $E(n^*)$ satisfies

$$E(n^*) = F(n) + 0(n^{\epsilon})$$

for all $n \geq 2$, where ϵ is any constant with $0 < \epsilon < 1$.

Proof: A simple rearrangement indicates

$$E(n^*) = \sum_{i=1}^{\infty} q(i,n)\, 2^i - \sum_{i=1}^{\infty} q(1+1, n)\, 2^i = 1 + \sum_{i=0}^{\infty} q(i+1, n)\, 2^i .$$

Similarly,

$$F(n) = \psi(2n) + \sum_{i=0}^{\infty} \psi(\frac{n}{2^i})\, 2^i .$$

Thus, it suffices to show

$$\sum_{i=0}^{\infty} \psi(\frac{n}{2^i})\, 2^i - \sum_{i=0}^{\infty} q(i+1, n)\, 2^i = 0(n^{\epsilon}).$$

Let $p = \lceil \log_2 n \rceil$. We need two facts, which are proved in the Appendix:

1. $\psi(\frac{n}{2^i})$ and $q(i+1, n)$ are each less than $\frac{1}{2^{(i-p)(i-p+1)}}$, and
2. $\psi(\frac{n}{2^i}) - q(i+1, n) = 0(\frac{\log n}{n})$, uniformly in i.

Using the bound for $\psi(\frac{n}{2^i}) - q(i+1, n)$ we obtain

$$\sum_{i \le p+\sqrt{p}} [\psi(\frac{n}{2^i}) - q(i+1,n)]\, 2^i = 0(\frac{\log n}{n}) \sum_{i \le p+\sqrt{p}} 2^i$$

$$= 0(\frac{\log n}{n}) 0(n 2^{\sqrt{\log n}}) = 0(n^\epsilon) .$$

Using the fact about the rapid decline of $\psi(\frac{n}{2^i})$ and $q(i+1,n)$ with i, we obtain

$$\sum_{i > p+\sqrt{p}} [\psi(\frac{n}{2^i}) - q(i+1,n)]\, 2^i < \sum_{i > p+\sqrt{p}} \frac{2^i}{2^{(i-p)(i-p+1)}}$$

$$= 2^p \sum_{i > \sqrt{p}} 2^{-i^2} = 0(n)\, 0(\frac{1}{n}) = 0(1).$$

■

Following Davies [4], we define the *Mellin transform* of a real function $f(x)$ as

$$M(f(x), s) = \int_0^\infty f(x)\, x^{s-1}\, dx,$$

where s is a complex variable. As a notational convenience, we will write $M(f(x), s)$ as $f^*(s)$. Thus, the Mellin transform of $F(x)$, for F as described in equation (2), is

$$F^*(s) = \int_0^\infty [\sum_{k=0}^\infty 2^k\, \Psi(\frac{x}{2^k})]\, x^{s-1}\, dx.$$

By the change of variable, $y = \frac{x}{2^k}$,

$$F^*(s) = \int_0^\infty [\sum_{k=0}^\infty 2^{k(s+1)}\, \Psi(y)] y^{s-1}\, dy .$$

Assume $\mathrm{Re}(s) < -1$ so the geometric sum converges, and

$$F^*(s) = \frac{1}{1 - 2^{s+1}} \int_0^\infty \Psi(y)\, y^{s-1} dy .$$

It is not hard to show that $\Psi(y) = 0(y^d)$ as $y \to 0$ and $\Psi(y) = 0(y^{-d})$ as $y \to \infty$ for any

$d > 1$. Using these bounds, it can be verified that

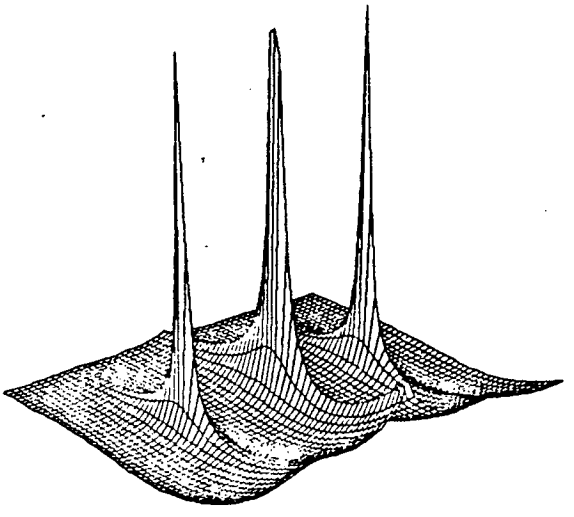
$$\Psi^*(s) = \int_0^\infty \Psi(y) y^{s-1} dy$$

is entire (analytic for all s). Thus $F^*(s)$, as given by the integral representation, is analytic for $\mathrm{Re}(s) < -1$ and the expression

$$F^*(s) = \frac{\Psi^*(s)}{1 - 2^{s+1}},$$

shows that it is meromorphic in the whole of the complex plane, with simple poles at $s = x_k = -1 + \frac{2\pi i k}{\log 2}$. (These are the roots of $1 - 2^{s+1}$.) Figure 6 gives a rendition of $|F^*(s)|$ plotted versus s on the neighborhood of the poles at x_{-1}, x_0, and x_1.

Figure 6. $|F^*(s)|$



By the inverse Mellin transform theorem,

$$F(x) = \frac{1}{2\pi i} \int_{-\frac{3}{2}-i\infty}^{-\frac{3}{2}+i\infty} F^*(s)\, x^{-s}\, ds = \frac{1}{2\pi i} \int_{-\frac{3}{2}-i\infty}^{-\frac{3}{2}+i\infty} \frac{\Psi^*(s)}{1 - 2^{s+1}}\, x^{-s}\, ds.$$

(The constant $-\frac{3}{2}$ appearing in the limits of the integral could be replaced with any

constant less than -1.) The analysis is nearly done, because it turns out to be possible to derive a precise asymptotic bound for $F(x)$ from this integral representation.

Theorem 2: For any constant ϵ with $0 < \epsilon < 1$, and all $n \geq 2$,

$$E(n^*) = n \frac{1}{\log 2}(\Psi^*(-1) + U(\log_2 n)) + 0(n^{\epsilon}),$$

where

$$\frac{\Psi^*(-1)}{\log 2} = \frac{1}{\log 2} \int_0^{\infty} e^{-x}(1+x) \prod_{k=1}^{\infty} (1 - e^{-2^k x}\ (1 + 2^k\ x)) / x^2\, dx$$
$$= 0.91422...,$$

and $U(z)$ is the Fourier series

$$U(z) = \sum_{k \neq 0} \Psi^*(-1 + 2\pi i k / \log 2) \exp(-2\pi i k z),$$

which has mean value 0 (k ranges over the integers). (The amplitude of U is approximately $2 \cdot 10^{-5}$.)

Proof: We intend to show that for all $x \geq 2$

$$F(x) = x \frac{1}{\log 2} (\Psi^*(-1) + U(\log_2 x)) + 0(1)\ .$$

This is sufficient since, by Lemma 1, $E(n^*) = F(n) + 0(n^{\epsilon})$. Cauchy's residue theorem provides a relationship between the integral

$$F(x) = \frac{1}{2\pi i} \int_{-\frac{3}{2}-i\infty}^{-\frac{3}{2}+i\infty} \frac{\Psi^*(s)}{1 - 2^{s+1}}\ x^{-s}\ ds$$

and the singularities of its integrand. As we remarked above, these singularities are simple poles at $s = x_k = -1 + \frac{2\pi i k}{\log 2}$, for $k = 0, \pm 1, \pm 2, ...$ Consider the contour integral

$$\int_{\Gamma_N} \frac{\Psi^*(s)}{1-2^{s+1}}\, x^{-s}\, ds$$

where Γ_N is described in Figure 7.

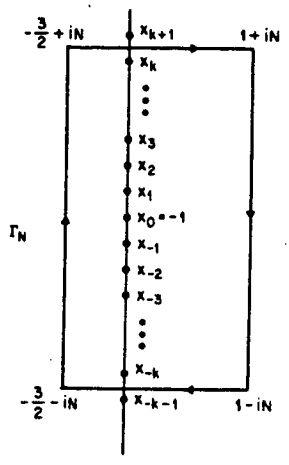


Figure 7. The contour Γ_N

N is chosen so that the top and bottom segments of Γ_N pass halfway between successive poles.

It happens that for large N, the value of this integral is concentrated along the vertical segment. Specifically, we claim (and defer the proof)

$$\lim_{N\to\infty} \int_{\Gamma_N} \frac{\Psi^*(s)}{1-2^{s+1}}\, x^{-s}\, ds = \int_{-\frac{3}{2}-i\infty}^{-\frac{3}{2}+i\infty} \frac{\Psi^*(s)}{1-2^{s+1}}\, x^{-s}\, ds + 0(1)\,. \tag{3}$$

By the residue theorem, the integral along Γ_N is $-2\pi i$ times the sum of the integrand's residues inside Γ_N. Thus,

$$\lim_{N\to\infty} \int_{\Gamma_N} \frac{\Psi^*(s)}{1-2^{s+1}}\, x^{-s}\, ds = 2\pi i \sum_k - \text{Res}\left(\frac{\Psi^*(s)}{1-2^{s+1}}\, x^{-s}, x_k\right),$$

summing over all integers k, so

$$F(x) = \sum_k -Res\left(\frac{\Psi^*(s)}{1-2^{s+1}}\, x^{-s},\, x_k\right) + 0(1)\ .$$

The residue at $x_0 = -1$ is given by

$$\begin{aligned}\text{Res}\left(\frac{\Psi^*(s)}{1-2^{s+1}}\, x^{-s}\ ,\, -1\right) &= \lim_{s\to -1}(s+1)\,\frac{\Psi^*(s)}{1-2^{s+1}}\, x^{-s} \\ &= x\ \Psi^*(-1)\lim_{s\to -1}\frac{s+1}{1-2^{s+1}} \\ &= -\,x\frac{\Psi^*(-1)}{log\,2}\ .\end{aligned}$$

Similarly, the residue at $x_k = -1 + \dfrac{2\pi i k}{\log 2}$ is given by

$$-x\ \frac{\Psi^*(-1 + 2\pi i k/\log 2)}{\log 2}\ \exp(-2\pi i k \log_2 x)\ .$$

Hence

$$F(x) = \frac{x}{\log 2}\left(\Psi^*(-1) + \sum_{k\neq 0}\Psi^*(-1 + 2\pi i k/\log 2)\ \exp(-2\pi i k\ \log_2 x)\right) + 0(1),$$

the desired result.

It remains to justify equation (3). Break the integral along Γ_N into four parts:

$$\int_{\Gamma_N} = \int_{-\frac{3}{2}-iN}^{-\frac{3}{2}+iN} + \int_{-\frac{3}{2}+iN}^{1+iN} + \int_{1+iN}^{1-iN} + \int_{1-iN}^{-\frac{3}{2}-iN}\ .$$

It suffices to show that the last three integrals are $0(1)$, uniformly in N. To show this we need a crude bound for $\Psi^*(s)$:

$$\Psi^*(s) = 0\left(\frac{1}{s^2}\right)\ ,$$

for all s on Γ_N, uniformly in N as $N \to \infty$. The proof of this bound is similar in spirit to the proof of the Reimann-Lesbegue lemma. Integrating by parts twice we obtain

$$\Psi^*(s) = \int_0^\infty \Psi(x)\, x^{s-1}\, dx = \frac{1}{s(s+1)} \int_0^\infty \Psi''(x)\, x^{s+1}\, dx \ ,$$

where $\Psi''(x)$ is the second derivative of $\Psi(x)$. (The terms not involving the integral in the integration by parts formula turn out to be 0.) Taking the modulus of both sides and using simple estimates ($\Psi''(x) = 0(x^{-d})$ as $x \to 0$ and $\psi''(x) = 0(x^{-d})$ as $x \to \infty$, for any $d > 1$) produces the bound.

Now, consider the integral along the top, horizontal segment of Γ_N:

$$\int_{-\frac{3}{2}+iN}^{1+iN} \frac{\Psi^*(s)}{1-2^{s+1}}\, x^{-s}\, ds = \int_{-\frac{3}{2}}^{1} \frac{\Psi^*(t+iN)}{1-2^{t+iN+1}}\, x^{-(t+iN)}\, dt \ ,$$

so

$$\left| \int_{-\frac{3}{2}}^{1} \frac{\Psi^*(t+iN)}{1-2^{t+iN+1}}\, x^{-(t+iN)} dt \right| \le \int_{-\frac{3}{2}}^{1} \left| \frac{\Psi^*(t+iN)}{1-2^{t+iN+1}} \right| \left| x^{-(t+iN)} \right| dt$$

$$= \int_{-\frac{3}{2}}^{1} 0\left(\frac{1}{N^2}\right) x^{-t}\, dt = 0\left(\frac{1}{N^2}\right) \ ,$$

which vanishes as $N \to \infty$ (x is fixed at this point). Similarly, the integral along the bottom, horizontal segment of Γ_N vanishes as $N \to \infty$. Lastly, consider the integral along the right, vertical segment of Γ_N:

$$\int_{1+iN}^{1-iN} \frac{\Psi^*(s)}{1-2^{s+1}}\, x^{-s}\, ds = \int_{+N}^{-N} \frac{\Psi^*(1+it)}{1-2^{2+it}}\, x^{-(1+it)}\, i dt \ .$$

Taking the modulus and making some simplifications, we find

$$\left| \int_{N}^{-N} \frac{\Psi^*(1+it)}{1-2^{2+it}} x^{-(1+it)}\, dt \right| \leq \frac{1}{x} \int_{-\infty}^{+\infty} |\Psi^*(1+it)|\, dt$$

$$= 0(1) .$$

The last step follows from the fact that $x \geq 2$, and the fact that the integral converges in view of $\Psi^*(s) = 0(\frac{1}{s^2})$. ■

This completes the analysis of $E(n^*)$. We now outline the key steps in a similar analysis that leads to a bound for the second moment of n^*, $E((n^*)^2)$. Our point of departure is the equation

$$E((n^*)^2) = \sum_{k=1}^{\infty} 4^k \, (q(k,n) - q(k+1,n)) .$$

The exponential approximation, $(1-x)^n \approx e^{-x\,n}$, suggests that

$$G(n) = \sum_{k=0}^{\infty} 4^k \Psi(\frac{n}{2^k})$$

should be close to $E((n^*)^2)$, and indeed it turns out that

$$E((n^*)^2) = G(n) + O(n^{1+\epsilon}),$$

where ϵ is any constant with $0 < \epsilon < 1$. The Mellin transform of G is related to that of Ψ by

$$G^*(s) = \frac{\Psi^*(s)}{1-2^{s+2}} ,$$

which defines $G^*(s)$ as an analytic function of all s, excluding the points $s = x_k = -2 + \frac{2\pi i k}{\log 2}$, which are simple poles of $\frac{\Psi^*(s)}{1-2^{s+2}}$, for $k = 0, \pm 1, \pm 2, \ldots$. By the inverse Mellin transform theorem,

$$G(x) = \frac{1}{2\pi i} \int_{\frac{-5}{2} - i\infty}^{\frac{-5}{2} + i\infty} \frac{\Psi^*(s)}{1 - 2^{s+2}} x^{-s}\, ds.$$

Using the residue theorem (as in the proof of Theorem 2) to relate the integral to the singularities of its integrand, we find that, for any $x \geq 2$,

$$G(x) = \frac{x^2}{\log 2}(\Psi^*(-2) + V(\log_2 x)) + 0(1)$$

where

$$\frac{\Psi^*(-2)}{\log 2} = \frac{1}{\log 2} \int_0^\infty e^{-x}(1 + x) \prod_{k=1}^{\infty} (1 - e^{-2^k x}(1 + 2^k x)) / x^3\, dx$$

$$= 1.23278...$$

and $V(z)$ is the Fourier series

$$V(z) = \sum_{k \neq 0} \Psi^*(-2 + 2\pi i k / \log 2) \exp(-2\pi i k z),$$

which has mean value 0. Numerical calculations indicate that the amplitude of V is $\approx 10^{-4}$. In sum, we have

Theorem 3: For any constant ϵ with $0 < \epsilon < 1$, and all $n \geq 2$,

$$E((n^*)^2) = n^2 \frac{1}{\log 2}(\Psi^*(-2) + V(\log_2 n)) + O(n^{1+\epsilon}),$$

where Ψ^* and V are as described above.

4. ANALYSIS OF THE SIMPLE HYBRID ALGORITHM

In this section we present a precise asymptotic bound for the expected running time of the simple hybrid algorithm, described in section 2.2. Recall that the algorithm works in two stages:

1. An estimate n^* of n is computed, which leads to a partition of the stations into n^* groups.
2. The groups are processed serially using the tree algorithm.

It is not hard to show that $\log_2 n + O(1)$ expected time elapses in stage 1 (the additive constant can be pinpointed up to a $o(1)$ error term), and $\Theta(n)$ expected time in stage 2. Because the cost of stage 2 swamps that of stage 1, we will not present a precise treatment of the cost of stage 1. Let $T(n)$ denote the expected time used in stage 2, and let $T(n \mid 2^k)$ denote this expectation conditioned on $n^* = 2^k$. Then

$$T(n) = \sum_{k=1}^{\infty} T(n \mid 2^k)\, Prob(n^* = 2^k). \tag{4}$$

Our aim is to find a precise asymptotic bound for this sum.

To begin, note that $T(n \mid 2^k)$ represents the expected time used in 2^k invocations of the simple tree algorithm. The number of stations involved in each invocation is a binomially distributed random variable, with parameters n and 2^{-k}. By symmetry, $T(n \mid 2^k) = 2^k\, C(n \mid 2^k)$, where $C(n \mid 2^k)$ is the expected time used in any one of the 2^k invocations. We claim

$$C(n \mid 2^k) = 1 + 2 \sum_{j=0}^{\infty} 2^j \left(1 - (1 - 2^{-k-j})^n - (1 - 2^{-k-j})^{n-1}\, 2^{-k-j}\, n\right). \tag{5}$$

To see this, consider the binary tree associated with one of the invocations (cf. Figure 3). Each node in the tree corresponds to a step in the course of the execution. An internal node, which corresponds to a collision, occurs at depth j with probability

$$1 - (1 - 2^{-k-j})^n - (1 - 2^{-k-j})^{n-1}\, 2^{-k-j}\, n,$$

since this is the probability of two or more stations joining the same group (out of the 2^k equally likely possibilities) and then agreeing on their first j coin tosses. As there are 2^j possible internal nodes at depth j, the sum in equation (5) represents the expected number

of internal nodes in the tree. Because the total number of nodes, which is identical to the running time, is one greater than twice the number of internal nodes, $C(n \mid 2^k)$ represents the expected running time.

We now develop a suitable approximation for $T(n)$. By the exponential approximation, $(1-x)^n \approx e^{-x\,n}$, we expect $C(n \mid 2^k)$ to be close to $c(\frac{n}{2^k})$ where

$$c(x) = 1 + 2\sum_{j=0}^{\infty} 2^j\,(1 - e^{-2^{-j}\,x}(1 + 2^{-j}\,x)).$$

Indeed, substituting $c(\frac{n}{2^k})$ for $C(n \mid 2^k)$ and accounting for the error incurred leads to

$$T(n \mid 2^k) = 2^k c(\frac{n}{2^k}) + O(n^{\epsilon}).$$

This holds uniformly in k, for any constant ϵ with $0 < \epsilon < 1$. (See reference [19], pages 132-133, for an essentially identical result.) As in section 5 we use $\Psi(\frac{n}{2^i})$ to approximate $Prob(n^* = 2^i)$, where

$$\Psi(x) = \psi(2x) - \psi(x) = e^{-x}\,(1 + x)\prod_{k=1}^{\infty}(1 - e^{-2^k\,x}(1 + 2^k x)).$$

Guided by these approximations and equation (4), we define the following approximation for $T(n)$:

$$t(n) = \sum_{k=0}^{\infty} 2^k\; c(\frac{n}{2^k})\;\Psi(\frac{n}{2^k}). \tag{6}$$

Lemma 4: The expectation $T(n)$ satisfies

$$T(n) = t(n) + 0(n^{\epsilon}),$$

for all $n \geq 2$, where ϵ is any constant with $0 < \epsilon < 1$.

Proof: As noted above, $T(n \mid 2^k) = 2^k\, c(\frac{n}{2^k}) + O(n^\epsilon)$, so

$$\begin{aligned} T(n) &= \sum_{k=1}^{\infty} T(n \mid 2^k)\, Prob(n^* = 2^k) \\ &= \sum_{k=1}^{\infty} 2^k\, c(\frac{n}{2^k})\, Prob(n^* = 2^k) + O(n^\epsilon). \end{aligned}$$

Thus, it suffices to show

$$\sum_{k=1}^{\infty} 2^k c(\frac{n}{2^k})\, Prob\ (n^* = 2^k) - \sum_{k=0}^{\infty} 2^k\, c(\frac{n}{2^k})\, \Psi(\frac{n}{2^k}) = 0(n^\epsilon)\ .$$

Let us recall some notation from section 3, namely $q(k, n) = \text{Prob}\ (n^* \geq 2^k)$ and $\Psi(\frac{n}{2^k}) = \psi(\frac{2n}{2^k}) - \psi(\frac{n}{2^k})$. In these terms, the left hand side of the last equation is

$$\sum_{k=1}^{\infty} 2^k\, c(\frac{n}{2^k})\, [q(k, n) - \psi(\frac{2n}{2^k})] - \sum_{k=1}^{\infty} 2^k\, c(\frac{n}{2^k})\, [q(k+1, n) - \psi(\frac{n}{2^k})] - c(n)\Psi(n)\ .$$

We will prove that the first term is $O(n^\epsilon)$. The same holds for the other two terms; and the proofs are similar.

A crude bound for $c(x)$ will be helpful. Note

$$\sum_{2^j < x} 2^j\, (1 - e^{-2^{-j} x}(1 + 2^{-j} x)) < \sum_{2^j < x} 2^j < 2\, x,$$

and

$$\sum_{2^j \geq x} 2^j\, (1 - e^{-2^{-j} x}(1 + 2^{-j} x)) < \sum_{2^j \geq x} 2^j\, (\frac{x}{2^j})^2 < x \sum_{j \geq 0} 2^{-j} = 2\, x.$$

Here we used the fact that $e^{-z} \geq 1 - z$ for all real z. Thus,

$$\begin{aligned} c(x) &= 1 + 2 \sum_{j=0}^{\infty} 2^j\, (1 - e^{-2^{-j} x}(1 + 2^{-j} x)) \\ &\leq 1 + 8x. \end{aligned}$$

Now let $p = \lceil \log_2 n \rceil$. By Lemma A2 of the Appendix, $q(k, n) - \psi(\frac{2n}{2^k}) = O(\frac{\log n}{n})$,

uniformly in k, so

$$\sum_{k \le p+\sqrt{p}} 2^k \, c(\frac{n}{2^k}) \, [q(k,n) - \psi(\frac{2n}{2^k})] = O(\frac{\log n}{n}) \sum_{k \le p+\sqrt{p}} 2^k \, c(\frac{n}{2^k})$$

$$\le O(\frac{\log n}{n}) \sum_{k \le p+\sqrt{p}} (2^k + 8n)$$

$$= O((\log n) \, 2^{\sqrt{\log n}}) = O(n^\epsilon),$$

for any ϵ with $0 < \epsilon < 1$. It remains to show that the complementary sum,

$$\sum_{k > p+\sqrt{p}} 2^k \, c(\frac{n}{2^k}) \, [q(k,n) - \psi(\frac{2n}{2^k})],$$

is at least as small. By Lemma A1 of the Appendix, both $q(p+k,n)$ and $\psi(\frac{2n}{2^{p+k}})$ are less than $2^{-k(k-1)}$, so

$$\sum_{k > p+\sqrt{p}} 2^k \, c(\frac{n}{2^k}) \, [q(k,n) - \psi(\frac{2n}{2^k})] < \sum_{k > \sqrt{p}} 2^{p+k} \, (1 + 8\frac{n}{2^{p+k}}) \, 2^{-k(k-1)}$$

$$= O(n) \sum_{k > \sqrt{p}} 2^{-k^2+2k}$$

$$= 0(2^{\sqrt{\log n}}) = 0(n^\epsilon) .$$

■

To get a more informative asymptotic bound for $T(n)$, we once again call on the Mellin transform. Let

$$\alpha(x) = c(x) \, \Psi(x),$$

so we may rewrite equation (6) as

$$t(x) = \sum_{k=0}^{\infty} 2^k \, \alpha(\frac{x}{2^k}) .$$

The Mellin transform of $t(x)$ is

$$t^*(s) = \int_0^\infty t(x)\, x^{s-1}\, dx = \int_0^\infty \sum_{k=0}^\infty 2^k\, \alpha(\frac{x}{2^k})\, x^{s-1}\, dx$$

$$= \int_0^\infty \sum_{k=0}^\infty 2^{k(s+1)}\, \alpha(x)\, x^{s-1}\, dx$$

$$= \frac{1}{1-2^{s+1}} \int_0^\infty \alpha(x)\, x^{s-1}\, dx$$

$$= \frac{\alpha^*(s)}{1-2^{s+1}}, \qquad (7)$$

provided $\mathrm{Re}(s) < -1$ so that the sum converges. It is not hard to show that $\alpha(x) = O(x^d)$ as $x \to 0$ and $\alpha(x) = O(x^{-d})$ as $x \to \infty$, for any $d > 1$. Using these bounds, one can show $\alpha^*(s)$ is analytic for all s. It follows by the principle of analytic continuation that $t^*(s)$, as given by equation (7), is analytic for all s, excluding the points $s = x_k = -1 + \frac{2\pi i k}{\log 2}$, which are the simple poles of $\frac{\alpha^*(s)}{1+2^{s+1}}$, $k = 0, \pm 1, \pm 2, \ldots$. By the inverse Mellin transform theorem,

$$t(x) = \frac{1}{2\pi i} \int_{\frac{-3}{2}-i\infty}^{\frac{-3}{2}+i\infty} \frac{\alpha^*(s)}{1-2^{s+1}}\, x^{-s}\, ds.$$

An asymptotic analysis of this integral leads to the following result.

Theorem 5: For any ϵ with $0 < \epsilon < 1$, and all $n \geq 2$,

$$T(n) = n\, \frac{1}{\log 2}(\alpha^*(-1) + W(\log_2 n)) + O(n^\epsilon)$$

where

$$\frac{\alpha^*(-1)}{\log 2} = \frac{1}{\log 2} \int_0^\infty \left[1 + 2 \sum_{k=0}^{\infty} 2^k \ (1 - e^{-2^k x} (1 + 2^{-k} x))\right]$$

$$\left[e^{-x} (1 + x) \prod_{k=1}^{\infty} (1 - e^{-2^k x}(1 + 2^k))\right] x^{-2} \, dx$$

$$= 2.49035...,$$

and $W(z)$ is the Fourier series

$$\sum_{k \neq 0} \alpha^*(-1 + 2\pi i k/\log 2) \exp(-2\pi i k z)$$

which has mean value 0 (k ranges over the integers). (The amplitude of W is approximately 10^{-5}.)

Proof: By the previous lemma, $T(n) - t(n) = 0(n^\epsilon)$, so it suffices to show that, for all $x \geq 2$,

$$t(x) = x \frac{1}{\log 2} (\alpha^*(-1) + W(\log_2 x)) + 0(1) .$$

Our proof of this bound is similar to the proof of Theorem 2. Consider the contour integral

$$\int_{\Gamma_N} \frac{\alpha^*(s)}{1 - 2^{s+1}} x^{-s} \, ds$$

where Γ_N is as described in Figure 7. By the residue theorem,

$$\lim_{N \to \infty} \int_{\Gamma_N} \frac{\alpha^*(s)}{1 - 2^{s+1}} x^{-s} \, ds = 2\pi i \sum_k - Res\left(\frac{\alpha^*(s)}{1 - 2^{s+1}} x^{-s}, x_k\right),$$

where k ranges over the integers. After evaluating the residues (as in the the proof of Theorem 2), we obtain

$$\lim_{N\to\infty} \int_{\Gamma_N} \frac{\alpha^*(s)}{1-2^{s+1}}\, x^{-s}\, ds = x\, \frac{2\pi i}{\log 2}\, (\alpha^*(-1) + W(\log_2 x)),$$

where $W(z)$ is as described in the statement of the Theorem. Moreover, the integral along Γ_N is concentrated along the left vertical segment, that is,

$$\lim_{N\to\infty} \int_{\Gamma_N} \frac{\alpha^*(s)}{1-2^{s+1}}\, x^{-s}\, ds = \int_{\frac{-3}{2}-i\infty}^{\frac{-3}{2}+i\infty} \frac{\alpha^*(s)}{1-2^{s+1}}\, x^{-s}\, ds + 0(1).$$

To prove this, we express the integral along Γ_N as the sum of the four integrals along the four segments of Γ_N. Using a crude bound for $\alpha^*(s)$ ($\alpha^*(s) = 0(\frac{1}{s^2})$ works), we can show that the contributions of the top, bottom, and right vertical segments are $0(1)$. ■

5. IMPROVED ALGORITHMS

Consider the estimate of n provided by the base 2 algorithm, $n^+ = \frac{n^*}{\phi}$, where $\phi = \frac{\Psi^*(-1)}{\log 2} \approx 0.9142$ (cf. Theorem 2). Even though the expected value of n^+ is quite close to n, n^+ is likely to differ from n by a factor of two. A small generalization of the base 2 algorithm remedies this defect, providing an estimate whose mean is close to n but whose distribution peaks more sharply about the mean. Figure 8 describes the *base a estimation algorithm*, which differs from the base 2 algorithm only in that it uses coin tosses with bias a^{-i} instead of 2^{-i}, where a is an arbitrary constant greater than 1. The improvement alluded to above comes from picking a close to 1.

Figure 8. Base a Estimation Algorithm

$i := 0$
repeat
$i := i + 1$
With probability a^{-i}, transmit to the channel
until no collision occurs
$n^* := a^i$

Next, consider the complementary generalization of the hybrid algorithm of section 2.2:

Figure 9. General Hybrid Algorithm

1. Compute $n^*(a)$ using the base a estimation algorithm and let $m = \max\{2, \lfloor s\, n^*(a) \rfloor\}$, where $a > 1$ and the *slide parameter* $s > 0$ are constants.
2. Divide the conflicting stations into m groups by having each pick a number uniformly at random between 1 and m. Process the m groups serially using a conflict resolution algorithm.

Of course the conflict resolution algorithm used at stage 2 should be an efficient one, especially on conflicts of small multiplicity. We will consider three choices for this algorithm:

- the simple tree algorithm [2,3,27],
- the modified tree algorithm [2,3,21,27], and
- the modified, biased tree algorithm [6,17].

As noted earlier, the ratio $\frac{m}{n}$ has a major impact on the performance of the hybrid algorithm. The role of the slide parmeter s is to steer $\frac{m}{n}$ close to the optimal values. In section 5.2, we develop a precise asymptotic formula for the expected running time expressed as a function of $\frac{m}{n}$. This formula coupled with one describing the expected value of $n^*(a)$ (Theorem 6, section 5.1) leads to a good heuristic for choosing s. Our results indicate that under this heuristic, as a tends to 1 the expected running time for large n tends to that of the idealized algorithm in which m is fixed *a priori* so that $\frac{m}{n}$ is at

the optimal value.

We now give a brief description of the alternate tree algorithms under consideration. Let us reconsider the example in section 2.2 of the execution of the simple tree algorithm on a conflict of multiplicity 3. At step 1 there is a collision, at step 2 there is another collision, and at step 3 there are no transmissions at all. From the feedback these three steps produce, all stations could infer that step 4 is doomed to hold a collision. To avoid the sure collision and save a step, Massey and Tsybakov and Mikhailov [2,3,21,27] suggested a *modified tree algorithm* that uses slightly different rules governing how a station updates its local counter c, which in turn governs transmission. In addition to c, each station must maintain a local variable d, which assumes the value 0 when the situation just described arises and the value 1 at all other times. Table 4 describes the new rules.

TABLE 4. Modified Tree Algorithm Implementation (A station maintains two local variables c and d, updating them in response to feedback as indicated below.)

	2+	1	0
$c = 0$	$c :=$ toss	success	(impossible)
$c \geq 1$	$c := c + 1$	$c := c - 1$	if $d = 0$ then $c :=$ toss else $c := c - 1$
$c \geq 2$	$c := c + 1$	$c := c - 1$	$c := c - d$

2+	1	0
$d := 0$	$d := 1$	$d := d$

Under these rules, the algorithm resolves conflicts of multiplicity n in expected time approximately 2.667 n, for large n, an improvement of about 8% over the simple tree algorithm [21,27].

However, the new rules lead to a certain imbalance: After a collision the expected time

to resolve the conflict among those that toss 0 slightly exceeds the expected time to resolve the one among those that toss 1. It happens that using a coin that is slightly less likely to come up 0 than 1 reduces the expected time to complete the whole tree. Fayolle and Hofri found that biasing the coin to turn up 0 with probability .4175 minimizes the expected running time for large n [6,17]. (Under this bias, the expected running time is approximately 2.662 n for large n). What's even more important for our purposes is that .4175 is close to the optimal bias for small n as well. We will be content here to fix the bias at .4175, instead of treating it as another parameter.

5.1 ANALYSIS OF THE IMPROVED ESTIMATION ALGORITHM

The results of this section have the flavor: As $a \to 1$, the base a estimation algorithm produces an increasingly sharp estimate of n. We note that the cost of the algorithm, measured as its expected running time, also increases as $a \to 1$. It is not hard to show that this cost is $\log_a n + O(1)$. However, in our main application -- conflict resolution using the hybrid algorithm -- this cost figures in as a low order term, while the attendant increased sharpness lessens the high order term.

The choice of base did not play a crucial role in the analysis of estimation algorithm in section 5, so that analysis generalizes readily. The first moment of $n^*(a)$, is given by

$$E(n^*(a)) = F(n) + O(n^{\epsilon}),$$

where ϵ is an arbitrary constant with $0 < \epsilon < 1$,

$$F(x) = \sum_{k=0}^{\infty} a^k \, \Psi(\frac{x}{a^k}),$$

and

$$\Psi(x) = e^{-x}\,(1+x)\prod_{k=1}^{\infty}(1 - e^{-a^k x}(1+a^k x)).$$

As in the base 2 case, the Mellin transforms $F^*(s)$ and $\Psi^*(s)$ of $F(x)$ and $\Psi(x)$ are simply related:

$$F^*(s) = \frac{\Psi^*(s)}{1-a^{s+1}}.$$

Inverting $F^*(s)$ leads to an integral representation of $F(x)$. Using contour integration to determine the asymptotics of the integral, we arrive at the following counterpart of Theorem 2.

Theorem 6: For any constant ϵ with $0 < \epsilon < 1$, and all $n \geq 2$,

$$E(n^*(a)) = n\,\frac{1}{\log a}(\Psi^*(-1) + U_a(\log_a n)) + O(n^{\epsilon}),$$

where

$$\Psi^*(-1) = \int_0^{\infty} e^{-x}(1+x)\prod_{k=1}^{\infty}(1 - e^{-a^k x}(1+a^k x))/x^2\,dx,$$

and $U_a(z)$ is the Fourier series

$$U_a(z) = \sum_{k\neq 0} \Psi^*(-1 + 2\pi i k/\log a)\exp(-2\pi i k z),$$

which has mean 0 and amplitude tending to 0 as a tends to 1.

(The fact that the amplitude of the U_a tends to 0 is proven below.) A similar analysis for the second moment of $n^*(a)$ leads to the following counterpart of Theorem 3.

Theorem 7: For any constant ϵ with $0 < \epsilon < 1$, and all $n \geq 2$,

$$E((n^*(a))^2) = n^2\,\frac{1}{\log a}(\Psi^*(-2 + 2\pi i k/\log a) + V_a(\log_a n)) + O(n^{1+\epsilon}),$$

where

$$\Psi^*(-2) = \int_0^\infty e^{-x}(1+x) \prod_{k=0}^{\infty} (1 - e^{-a^k x}(1 + a^k x)) \,/\, x^3 \, dx,$$

and $V_a(z)$ is the Fourier series

$$V_a(z) = \sum_{k \neq 0} \Psi^*(2 + 2\pi i k / \log a) \exp(-2\pi i k z),$$

which has mean value 0 and amplitude tending to 0 as a tends to 1.

We now turn to the problem of constructing a sharp estimate of n using $n^*(a)$. Since $n^*(a) \approx \phi(a) n$ where $\phi(a) = \dfrac{\Psi^*(-1)}{\log a}$, a natural choice for an estimate of n is

$$n^+(a) = \frac{n^*(a)}{\phi(a)} .$$

It turns out that the bias of $n^+(a)$, as measured by $|1 - \dfrac{E(n^+(a))}{n}|$, is close to 0 for small a, and sufficiently large n.

Theorem 8: For all n greater than some constant $n_0(a)$,

$$|\frac{E(n^+(a))}{n} - 1| < \epsilon(a) ,$$

where $\epsilon(a) \to 0$ as $a \to 1$.

Proof: It suffices to show that the Fourier series

$$U_a(z) = \sum_{k \neq 0} \Psi^*(-1 + 2\pi i k / \log a) \exp(-2\pi i k z)$$

tends to 0 as a tends to 1, uniformly for all $z \geq 0$. To do so it is convenient to work with

$$\psi(x) = \prod_{k=0}^{\infty} (1 - e^{-a^k x}(1 + a^k x)) ,$$

rather than $\Psi(x) = \psi(ax) - \psi(x)$. The transforms of the two are related by

Theorem 9: For all n greater than some constant $n_0(a)$,

$$\frac{\sigma(n^+(a))}{n} < \epsilon(a) ,$$

where $\epsilon(a) \to 0$ as $a \to 1$.

Proof: As in the previous proof it is more convenient to work with $\psi(x)$ and its transform $\psi^*(s)$ than $\Psi(x)$ and $\Psi^*(s)$. To emphasize the dependence on a, we write $\psi^*(s)$ as

$$\psi_a^*(s) = \int_0^\infty \prod_{j=0}^{\infty} (1 - e^{-a^j x} (1 + a^j x)) \, x^{s-1} \, dx .$$

By Theorems 6 and 7 and the correspondence $\psi_a^*(s) = (a^{-s+1} - 1) \Psi_a^*(s)$, we have

$$\frac{E((n^+(a))^2) - (E(n^+(a)))^2}{n^2} = K_a + W_a(\log n) + o(1)$$

where $W_a(u)$ is a periodic function of u that tends to 0 uniformly as $a \to 1$, and

$$K_a = \frac{(a+1)\log a}{a-1} \frac{\psi_a^*(-2)}{(\psi_a^*(-1))^2} - 1 .$$

Thus all we have to prove is that K_a tends to 0 as a tends to 1. This last fact follows directly from Lemma A3 of the Appendix, which states

$$\psi_a^*(s) \sim -\frac{1}{s} (\log \frac{1}{a-1}))^s . \quad \blacksquare$$

To complement these asymptotic results, we present results from a simulation study, which suggest that the trends described by Theorems 8 and 9 hold for small n as well. Specifically, we simulated the base a algorithm and computed the sample mean (Table 5a) and standard deviation (Table 5b) of $n^+(a)$, averaging over 10^5 trials. We found from these experiments that a small variation on the definition of $n^+(a)$,

$$n^+(a) = \frac{n^*(a) - 1}{\phi(a)}$$

works slightly better in estimating n than does $n^+(a)$ as defined above. We used this definition in the experiments. (The theorems hold for this definition as well.) It follows from the proof of Theorem 9 that, apart from negligible fluctuations and o(1) contributions, $\frac{\sigma(n^+(a))}{n}$ is

$$\left[\frac{(a+1)\log a}{(a-1)} \frac{\Psi^*(-2)}{(\Psi^*(-1))^2} - 1\right]^{\frac{1}{2}},$$

which is .6892 when $a=2$, .3438 when $a=1.1$, and .2127 when $a=1.01$. We note that these numerical values agree well with the simulation data of Table 5b.

TABLE 5. Simulation Results

5a) Expected Value of $n^+(a)$

	n=10	n=100	n=1000
a=2.00	9.552	99.965	987.212
a=1.10	11.310	101.876	995.864
a=1.01	12.762	102.725	1002.467

b) Standard Deviation of $n^+(a)$

	n=10	n=100	n=1000
a=2.00	6.886	68.432	666.990
a=1.10	3.440	35.055	341.695
a=1.01	2.107	21.359	211.823

5.2 ANALYSIS OF THE IMPROVED HYBRID ALGORITHM

In this subsection we generalize the analysis of section 4 to obtain a precise asymptotic bound for the expected running time of the improved hybrid algorithm. The dominant term in this expectation is the expected time $T(n)$ that elapses in processing the n stations using a tree algorithm on each of the $m = \max\{2, \lfloor sn^*(a) \rfloor\}$ groups:

$$T(n) = \sum_{k=1}^{\infty} T(n \mid \max\{2, \lfloor sa^k \rfloor\}) \text{ Prob } (n^*(a) = a^k),$$

where $T(n \mid m)$ denotes the same expectation conditioned on m. Naturally, the analysis

$$\Psi^*(s) = (a^{-s+1} - 1)\,\psi^*(s),$$

so

$$|U_a(z)| = 0(1) \sum_{k \neq 0} |\psi^*(-1 + 2\pi ik/\log a)| .$$

Let us consider the asymptotic behavior of

$$\psi^*(s) = \int_0^\infty \psi(x)\, x^{s-1}\, dx .$$

Integrate twice by parts to get

$$\psi^*(s) = \frac{1}{s(s+1)} \int_0^\infty \psi''(x)\, x^{s+1}\, dx ,$$

where ψ'' denotes the second derivative of $\psi(x)$. Thus,

$$|\psi^*(-1 + 2\pi ik/\log a| = \frac{0(\log^2 a)}{k^2} \int_0^\infty |\psi''(x)|\, dx$$

and

$$U_a(z) = 0(\log^2 a) \int_0^\infty |\psi''(x)|\, dx .$$

We will show $\log^2 a \int_0^\infty |\psi''(x)|dx$ tends to 0 as a tends to 1.

Let $p_j(x) = 1 - e^{-a^j x}\,(1 + a^j x)$, so $\psi(x) = \prod_{j \geq 0} p_j(x)$. The first two derivatives of $\psi(x)$ are

$$\psi'(x) = \sum_{j=0}^{\infty} \frac{p_j'(x)}{p_j(x)}\, \psi(x)$$

and

$$\psi''(x) = \psi(x)\left[\sum_{j=0}^{\infty}\left(\frac{p_j'(x)'}{p_j(x)} - \frac{p_j'(x)}{(p_j'(x))^2}\right) + \left(\sum_{j=0}^{\infty}\frac{p_j'(x)}{p_j(x)}\right)^2\right],$$

where p_j' and p_j'' denote the first two derivatives of p_j. A straightforward calculation indicates that for some constant $c > 0$, $|\psi''(x)| < c$ if $0 < x \leq 1$ and

$$|\psi''(x)| \leq c(\sum_{j=0}^{\infty} p_j(x))^2\,\psi(x)$$

if $x > 1$. Applying the Euler-McLauren summation formula, we find

$$\sum_{j=0}^{\infty} p_j(x) = \frac{1}{\log a}\int_0^{\infty} a^u\, x\, e^{-a^u x}du + 0(1)$$

$$= \frac{0(xe^{-x})}{\log a}$$

if $x \geq 1$. Thus,

$$\log^2 a \int_0^{\infty} |\psi''(x)|dx = 0(\log^2 a) + \int_1^{\infty} \psi(x)\, 0(x^2e^{-2x})\, dx\ .$$

Of course, the first term goes to 0 as a goes to 1. To see this for the second term, note $x^2e^{-2x} = 0(x^{-2})$ if $x \geq 1$, and

$$\int_1^{\infty} \frac{\psi(x)}{x^2}\, dx \to 0$$

as $a \to 1$ by Lemma A3 of the Appendix.

A similar argument indicates the Fourier series $V_a(z)$ that arises in the asymptotic analysis of $E((n^*(a))^2)$ (cf. Theorem 7) tends to 0 as a tends to 1. ■

Next, we show that the distribution of $n^+(a)$ becomes more peaked as $a \to 1$. Let $\sigma(n^+(a))$ denote the standard deviation of $n^+(a)$. For every a, $\sigma(n^+(a))$ is $\Theta(n)$, but the implicit constant goes to 0 as a goes to 1.

begins with $T(n \mid m)$. A spinoff of the analysis is a successful hueristic for deciding the value of the slide parameter.

Let us first assume that the underlying tree algorithm is the simple tree algorithm described in section 4.

In section 4 we gave formulas (equations (4) and (5)) describing $T(n \mid 2^k)$. These generalize easily to

$$T(n \mid m) = m\ C(n \mid m)$$

and

$$C(n \mid m) = 1 + 2\sum_{j=0}^{\infty} 2^j \left(1 - \left(1 - \frac{1}{2^j m}\right)^n - \left(1 - \frac{1}{2^j m}\right)^{n-1} \frac{n}{2^j m}\right),$$

for any integers $n, m \geq 2$. We use $c(\frac{n}{m})$ to approximate $C(n \mid m)$, where

$$c(x) = 1 + 2\sum_{j=0}^{\infty} 2^j \left(1 - e^{-2^{-j}x}(1 + 2^{-j}x)\right),$$

and can show

$$T(n \mid m) = m\ c(\frac{n}{m}) + 0(n^{\epsilon})$$

$$= (\frac{m}{n}\, c(\frac{n}{m}))\, n + 0(n^{\epsilon}).$$

where ϵ is any constant with $0 < \epsilon < 1$.

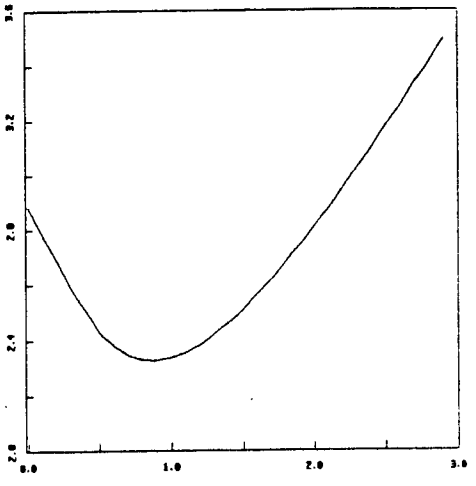
A nice property of the last equation is that the coefficient of n depends just on $\frac{m}{n}$. In the hybrid algorithm, m is a random variable. Specifically, $m = \max\{2, \lfloor sn^*(a) \rfloor\}$. Of

course we would like $x = \frac{m}{n}$ to minimize $x\, c(\frac{1}{x})$. Figure 6 depicts $x\, c(\frac{1}{x})$ plotted vs. x. The numerical results reflected in the plot indicate that $x\, c(\frac{1}{x})$ attains a global minimum of ≈ 2.3282 at $x = x_0 \approx 0.8710$. By Theorems 8 and 9, $\frac{m}{n} \approx s\, \frac{\Psi^*(-1)}{\log a}$, for small a and large n. This suggests the hueristic of setting the slide parameter s as

$$s = \frac{\log a}{\Psi^*(-1)}\, x_0,$$

where x_0 minimizes $x\, c(\frac{1}{x})$. Henceforth, we assume that s is fixed in this way.

Figure 10. Plot of $x\, c(\frac{1}{x})$



We now return to the analysis of $T(n)$, and define the exponential approximation

$$t(n) = \sum_{k=0}^{\infty} s a^k \alpha_s(\frac{n}{a^k}),$$

where $\alpha_s(x) = c(\frac{x}{s})\Psi(x)$. The approximation works in the sense that

$$T(n) = t(n) + 0(n^{\epsilon})$$

for any constant ϵ with $0 < \epsilon < 1$. The Mellin transform of $t(x)$ is

$$t^*(z) = \frac{s\alpha_s^*(z)}{1 - a^{z+1}}$$

where $\alpha_s^*(z)$, the Mellin transform of $\alpha_s(z)$, is

$$\alpha^*(z) = \int_0^{\infty} \alpha_s(x) x^{z-1}\, dx \ .$$

It can be verified that $\alpha^*(z)$ is entire and $t^*(z)$ as given above is analytic for all z excluding the points $z = x_k = -1 + \frac{2\pi i k}{\log a}$, which are simple poles of $\frac{s\alpha_s^*(z)}{1 - a^{z+1}}$ for $k = 0, \pm 1, \pm 2, \ldots$. By the inverse Mellin transform theorem,

$$t(x) = \int_{\frac{-3}{2} - i\infty}^{\frac{-3}{2} + i\infty} \frac{s\alpha_s^*(z)}{1 - a^{z+1}}\, x^{-z}\, dz \ .$$

Using contour integration to evaluate this integral, we obtain the following bound for $T(n)$. The proof is essentially the same as the proof of Theorem 5.

Theorem 10: For all $n \geq 2$,

$$T(n) = n\frac{s}{\log a}\left(\alpha_s^*(-1) + W_a(\log_a n)\right) + 0(n^{\epsilon}),$$

where $W_a(z)$ is the Fourier series

$$W_a(z) = \sum_{k \neq 0} \alpha_s^*(-1 + 2\pi i k/\log a) \exp(-2\pi i k z),$$

which has mean value 0 and amplitude tending to 0 as a tends to 1.

Thus, for small a and large n, $T(n) \approx n\frac{s\ \alpha_s^*(-1)}{\log a}$. Table 6 describes $\frac{s\ \alpha_s^*(-1)}{\log a}$ for $a = 2$, 1.1, 1.01, 1.001, and 1.0001 and s chosen according to the hueristic described above. Numerical results indicate that for these choices of s and a the amplitude of $W_a(\log_a n)$ is less than 10^{-4}.

The analysis is readily adapted to handle other choices for the underlying tree algorithm. The choice of tree algorithm affects the formulas for $C(n \mid m)$ and $c(x)$ only. Suppose that the underlying algorithm is the modified tree algorithm. A small variation on the combinatorial method of Fayolle and Hofri [6,17] then leads to

$$C(n \mid m) = 1 + 2\sum_{j=0}^{\infty} 2^j \left(1 - \left(1 - \frac{1}{2^j m}\right)^n - \left(1 - \frac{1}{2^j m}\right)^{n-1} \frac{n}{2^j m}\right)$$

$$-\sum_{j=0}^{\infty} 2^j\left[\left(1 - \frac{1}{2^{j+1}m}\right)^n - \left(1 - \frac{1}{2^j m}\right)^n - \left(1 - \frac{1}{2^j m}\right)^{n-1} \frac{n}{2^{j+1}m}\right].$$

The first line is the same as the old formula. The sum on the second line accounts for the expected number of steps saved by avoiding sure collisions. The corresponding exponential approximation $c(\frac{n}{m})$ for $C(n \mid m)$ is given by

$$c(x) = 1 + \sum_{j=0}^{\infty} 2^j \left[2(1 - e^{-2^{-j}x}(1 + 2^{-j}x)) - e^{-2^{-j+1}x} + e^{-2^{-j}x}\ (1 + 2^{-j-1}x)\right].$$

Theorems 9 and 10 hold under these substitutions. Numerical calculations indicate that $x\ c(\frac{1}{x})$ has the same shape as before, and has a global minimum of 2.1632 at $x = x_0 = .8000$. Again, we set the slide parameters to be $\frac{\log a}{\Psi^*(-1)}\ x_0$. Table 6 describes $T(n) \approx n\ s\ \frac{\alpha_s^*(-1)}{\log a}$ under these substitutions, and shows that the substitutions improve

performance by about 7 or 8%.

Lastly, suppose the underlying tree algorithm is the modified, biased tree algorithm. The new formula for $C(n \mid m)$ is once again easy to derive by the combinatorial method of Fayolle and Hofri [5,15], but is quite complicated. We present only the exponential approximation

$$c(x) = 1 + 2 \sum_{j=0}^{\infty} \sum_{k=0}^{j} \binom{j}{k} [1 - e^{-p^j(1-p)^{k-j}x} (1 + p^j(1-p)^{k-j}x)$$

$$- \sum_{j=0}^{\infty} \sum_{k=0}^{j} \binom{j}{k} [e^{-p^j(1-p)^{k-j+1}x} + e^{-p^j(1-p)^{k-j}} (-1 + p^j(1-p)^{k-j+1}x)] ,$$

where $p = .41737$ is the bias. Now $x\, c(\frac{1}{x})$ attains a global minimum of 2.1338 at $x = .7900$. We see from Table 6 that using the biased, modified tree algorithm results in the best performance (through the gain over the unbiased version is quite small).

TABLE 6. $\frac{s\alpha_s^*(-1)}{\log a}$ is given for $a = 2$, 1.1, 1.01, 1.001, and 1.0001, and the underlying tree algorithm being the simple tree, the modified tree, and the modified, biased tree.

	simple tree	modified tree	modified, biased tree
$a = 2$	2.4842	2.3034	2.2725
$a = 1.1$	2.3798	2.2095	2.1791
$a = 1.01$	2.3495	2.1830	2.1532
$a = 1.001$	2.3393	2.1732	2.1436
$a = 1.0001$	2.3356	2.1699	2.1403

Recall that the idea behind the setting of the slide parameter s is to steer $\frac{m}{n}$ close to the optimal value. If somehow we could fix $\frac{m}{n}$ at the optimal value, then $T(n)$ would be

$\approx 2.3282n$, or $2.1663n$, or $2.1338n$, depending on whether we use the simple, the modified, or the biased, modified tree algorithm (respectively). (These values are the global minima of the appropriate functions $x\ c(\frac{1}{x})$.) Table 6 shows that as a tends to 1 performance tends to this ideal level.

6. CHANNEL ACCESS PROTOCOLS

Capetanakis [2,3], Hayes [16], and Tsybakov and Mikhailov [27] introduced *channel access protocols* that in an appealing simple way coordinate access to the channel using algorithms for conflict resolution.

Let us review the relevant definitions and introduce some terminology. A channel access protocol proceeds as a sequence of *sessions*. At the beginning of every session, all stations holding packets transmit. If as a result 0 or 1 stations transmit, then the next sessions begins at the next step. Otherwise, the conflicting stations execute a conflict resolution algorithm, causing each of the packets involved in the conflict to be successfully transmitted, while the other stations defer. All stations sense the algorithm's termination, at which point the next session begins.

Capetanakis [2] and Tsybakov and Mikhailov [27] introduced a probabilistic model describing packet arrivals, which we call the *Poisson model*. It is assumed that an infinite number of stations share the channel. At a given moment, a station is either *active*, having a single packet ready for transmission, or *inactive*, having none. New packets arise among the inactive stations according to a stationary Poisson Process with intensity (mean number of packet arrivals per step) λ: At each step, for all $i > 0$, i new packets arrive with probability $\lambda^i e^{-\lambda}/i!$, each to a different (previously) inactive station. The packets are ready to be transmitted at the same step.

In the remainder of the section we examine the performance of channel access protocols, including ones based on tree algorithms and the hybrid algorithm, with respect to measures of long term *stability* in the face of a steady stream of packet arrivals, and short term *adaptivity* to sharp bursts of packet arrivals.

6.1 STABILITY

Following Fayolle and Hofri [6,17], we say a channel access protocol is *stable* if the expected length of each session is uniformly bounded. A simple criterion for stability at arrival rate λ (cf. [6,27]) is

$$\lambda < \liminf \frac{n}{T(n)}$$

where $T(n)$ denotes the expected time used in resolving a conflict of multiplicity n. Similarly, the protocol is unstable at arrival rate λ if

$$\lambda > \limsup \frac{n}{T(n)} .$$

(In the case of the hybrid and tree algorithms, the liminf and the limsup are extremely close but not equal, due to the contribution of a periodic term in the asymptotic formula for $T(n)$.) By the analysis of section 5.2, $T(n)$ for the hybrid algorithm is given by

$$T(n) = n(C_a + U_a(\log n)) + O(n^{\epsilon})$$

where the constant C_a is described in Table 6 (section 5.2) and the periodic term $U_a(\log n)$ has amplitude less than 10^{-4} for the values of a we considered ($a = 2, 1.1, 1.01, 1.001$, and 1.0001). The data of Table 6 indicates, in particular, that

- the hybrid algorithm using base 2 estimation and the simple tree algorithm is stable for all λ up to approximately 0.4025
- the hybrid algorithm using base 1.001 estimation and the modified, biased tree algorithm is stable for all λ up to approximately 0.4672.

It is possible to raise the degree of stability higher if the statistical nature of the Poisson model is taken into account. Several procedures have been proposed for controlling access to the channel, some of which attain stability at intensities up to 0.488, which do not fall in the category of channel access protocols just defined [1,11,23,25]. As an example of a scheme tuned to take advantage of the Poisson model, we briefly describe Capetanakis' dynamic tree protocol [2,3], which, as we mentioned earlier, inspired the discovery of the hybrid algorithm. Instead of computing m by running the estimation algorithm, m is computed as a function of the packet arrival intensity λ and the length L of the previous session: $m = 1$ if $\lambda L \leq 1.70$, and $m = \lceil(\lambda L - 0.55)/1.15\rceil$ otherwise. Capetanakis showed that under the Poisson model the dynamic tree protocol (using the modified tree algorithm at stage 2) is stable for all λ up to 0.462.

6.2 ADAPTIVITY TO BURSTY TRAFFIC

Consider a number of sessions in the execution of a random access scheme, or the dynamic tree protocol. We define the *throughput* during the sessions as the ratio of the expected number of packets that arrived to the expected length (number of steps) of the sessions. If the packets arrive according to the Poisson model at intensity λ then the throughput converges to λ, as the number of sessions increases, provided of course that the scheme is stable at intensity λ.

However if packet arrivals do not suit the Poisson model then this need not be the case. In this section we consider the throughput achieved over a small number of sessions in processing a sharp burst of packet arrivals. Specifically, we present simulation data contrasting, under this throughput measure,

- the channel access protocol based on the hybrid algorithm using base 2 estimation and the tree algorithm,

- the channel access protocol based on the simple tree algorithm, and
- the dynamic tree protocol using the simple tree algorithm. For concreteness, we assume the algorithm is tuned for an intensity λ of 0.4.

It turns out that the scheme based on the hybrid algorithm adapts best to large bursts of packets, followed by the scheme based on the tree algorithm, followed by the dynamic tree protocol.

Consider three sessions in the operation of a channel access protocol, immediately following a session involving no transmissions. At the beginning of the first session, assume exactly 2 stations transmit packets, causing a collision. In the resulting conflict resolution, b new packets arrive, where b is a parameter. During the second session, the b packets are successfully transmitted, while 2 additional packets arrive, which are in turn successfully transmitted during the third session. A measure of merit is the throughput the access control scheme achieves during the three sessions, $(4 + b)/E(L_1 + L_2 + L_3)$ where L_i denotes the number of steps consumed in the $i-th$ session. Intuitively, the closer this measure is to 1, the better the scheme adapts to the offered traffic.

Table 7 depicts throughput data collected by a simulation, averaging over 64,000 trials.

TABLE 7. Throughput Data

b	simple tree	simple hybrid	dynamic tree (tuned to $\lambda = 0.4$)
2	0.402	0.363	0.347
10	0.369	0.367	0.310
100	0.352	0.392	0.263
1000	0.346	0.401	0.258

As b gets large, the throughput of a channel access protocol whose underlying conflict resolution algorithm works in expected time $\approx c\,n + o(n)$ (for some constant c) will achieve a throughput of $\approx 1/c$. Thus, the scheme based on the tree algorithm has throughput $\approx 1/2.885 = 0.347$ for large b, and the scheme based on the hybrid algorithm

has throughput $\approx 1/2.490 = 0.402$ for large b. In contrast the dynamic tree protocol adapts poorly to the burst of arrivals, either predicting two few packets will belong to a session (in session 2), or predicting two many will belong (in session 3). For large b, using the fact that the simple tree algorithm runs in expected time 5 for conflicts of multiplicity 2 and $\approx 2.885\ b$ for conflicts of multiplicity b, throughput for the dynamic tree protocol is

$$\approx (4 + b)/(5 + (2.885\ b) + (2.885\ b\ \lambda/1.15))$$

which is ≈ 0.257 for $\lambda = 0.4$.

7. EXPECTED DELAY

In this paper we presented new algorithms for estimating the multiplicities of transmission conflicts, new algorithms for resolving conflicts, and precise asymptotic analyses of the algorithms' performance. Molle and Massey have recommended the simple tree algorithm as the conflict resolution algorithm of choice in practice [24, 19]. Some virtues of this algorithm include: simplicity, analytic tractabiltiy, robustness, and stability at arrival rates $\lambda < 0.3466$. To a large degree the simple hybrid algorithm presented here shares the first three traits, and it maintains stability at arrival rates $\lambda < 0.4025$, more than a 10% inprovement. In practice, an equally important performance measure is the mean packet delay, which is a function of the arrival rate. A packet that arrives at step t and is transmitted successfully at step s is said to have experienced delay $s-t-1$. Pinpointing the mean packet delay appears to be much harder analytically than finding the range of stability. (Using a method of Massey [21] and Tsybakov and Mikhailov [27] it is possible to develop certain lower and upper bounds for the mean packet delay. But tight bounds for, in particular, the channel access protocols based on the simple tree or simple hybrid algorithms are not yet available.) We simulated, various channel access protocols, and computed the following statistics on mean packet delay. (In these hybrid algorithm

simulations, if the estimation phase ends with one station transmitting successfully, then this station does not enter into the second phase.)

TABLE 8. Expected Delay Data

	D_{h2}	$D_{h1.75}$	$D_{h1.5}$	D_c	D_∞
λ = 0.100	0.42	0.42	0.42	0.41	0.47
0.200	1.45	1.43	1.50	1.41	1.83
0.300	7.41	7.12	8.35	6.46	10.93
0.350	27.71	27.92	35.56	--	--
0.375	80.11	79.05	96.23	--	--

D_{h2}, $D_{h1.75}$, $D_{h1.5}$ and D_c refer to the mean packet delay of the channel access protocols based on the simple hybrid, of the improved hybrid using base 1.75 estimation, the improved hybrid using base 1.50 estimation, and the simple tree algorithms (respectively). The two improved hybrid algorithms have the simple tree as the underlying tree algorithm. D_∞ refers to the mean packet delay of the continuous entry version of the simple tree algorithm [7]. (We took this column of data from the precise analytical results of [7], not from simulation.

As a final conclusion, use of the hybrid algorithm with a base in the range 2-1.5 results in improved throughput without appreciable increase in delay for small arrival rates. It thus appears to be the method of choice among the class of collision resolution algorithms that are based on the tree algorithm idea and resolve collisions in sessions. Continuous entry schemes are easier to implement but the price to be paid is both lower capacity and increased delay.

REFERENCES

[1] Berger, T., "The Poisson Multiple-Access Conflict Resolution Problem," School of Electrical Engineering, Cornell University (1979).

[2] Capetanakis, J., "Tree Algorithms for Packet Broadcast Channels," *IEEE Transactions on Information Theory* IT-25, 5 (September 1979), 505-515.

[3] Capetanakis, J., "Generalized TDMA: The Multi-Accessing Tree Protocol," *IEEE Transactions on Communications* COM-27, 10 (October 1979), 1479-1484.

[4] Davies, B., *Integral Transforms and Their Applications*, Springer, Berlin (1978).

[5] Digital-Intel-Xerox, "The Ethernet Data Link Layer and Physical Layer Specifications 1.0" (September, 1980).

[6] Fayolle, G. and Hofri, M., "On the Capacity of a Collision Channel Under Stack-Based Collision Resolution Algorithms," preprint (April, 1983).

[7] Fayolle, G., Flajolet, P, and Hofri, M., "On A Functional Equation Arising in the Analysis of a Protocol for a Multi-Access Broadcast Channel," INRIA Report 131 (April, 1982).

[8] Fayolle, G., Flajolet, P, Hofri, M., and Jacquet, P., "The Evaluation of Packet Transmission Characteristics in a Multi-Access Channel with Stack Resolution Protocol," INRIA Report 245 (October, 1983).

[9] Flajolet, P., "Methods in the Analysis of Algorithms: Evaluations of a Recursive Partitioning Process," *Proceedings of the 1983 International Conference on Foundations of Computation Theory*, Lecture Notes in Computer Science, vol. 158 (1983), Springer, Berlin.

[10] Franta, W. R. and Chlamtac, I. *Local Networks*, Lexington Books, Lexington, Mass. (1981).

[11] Gallagher, R.G., "Conflict Resolution in Random Access Broadcast Networks," *Proceedings of the AFOSR Workshop in Communication Theory and Applications* (September 1978), 74-76.

[12] Greenberg, A.G., "On the Time Complexity of Broadcast Communication Schemes," *Proceedings of the 14-th Annual ACM Symposium on Theory of Computing*, San Francisco, CA (1982), 354-364.

[13] Greenberg, A.G., "Efficient Algorithms for Multiple Access Channels," Ph.D. Thesis, University of Washington, Seattle (1983).

[14] Greenberg, A.G. and Winograd, S., "On the Time Needed To Resolve Conflicts Deterministically in Multiple Access Channels," preprint (1983).

[15] Greenberg, A. G. and Ladner, R. E. "Estimating the Multiplicities of Conflicts in Multiple Access Channels (preliminary version)," *Proceedings of the 24th Annual Symposium on Foundations of Computer Science*, Tuscon (November, 83).

[16] Hayes, J.F., "An Adaptive Technique for Local Distribution," *IEEE Transactions on Communications*, COM-26 (August 1978), 1178-1186.

[17] Hofri, M., "Stack Algorithms for Collision-Detecting Channels and Their Analysis: A Limited Survey," Technical Report 266, Israel Institute of Technology, Department of Computer Science (February, 1983).

[18] Komlos, J. and Greenberg, A.G. "A Fast Non-Adaptive Algorithm for Conflict Resolution in Multiple Access Channels," preprint (March, 1984).

[19] Knuth, D.E., *The Art of Computer Programming*, volume 3, Addison Wesley, Reading, Mass. (1973).

[20] Limb, J., "High Speed Operation Broadcast Local Networks," *IEEE International Conference on Communications '82* (June 1982), 6.C.1.1-6.C.1.5.

[21] Massey, J.L., "Collision-Resolution Algorithms and Random-Access Communications," *Multiuser Communication Systems* CISM Courses and Lectures No. 265, G,. Longo Ed., Springer-Verlag, Wien and New York (1981).

[22] Metcalfe, R. and Boggs, D., "Ethernet: Distributed Packet Switching for Local Computer Networks," *Communications of the ACM*, 19, 17 (July 1976), 395-404.

[23] Molle, M., "A Simulation Study of Retransmission Strategies for the Asynchronous Virtual Time CSMA Protocol," *Performance 83* (May, 1983).

[24] Molle, M., "Unifications and Extensions of the Multiple Access Communications Problem," Ph.D. Thesis, University of California, Los Angeles (July 1981).

[25] Mosely, J., "An Efficient Contention Resolution Algorithm for Multiple Access Channels," Technical Report LIDS-TH-918, Laboratory for Information an Decision Systems, MIT (June, 1979).

[26] Tanenbaum, A., *Computer Networks*, Prentice Hall, Englewood Cliffs, N.J. (1981).

[27] Tsybakov, B.S. and Mikhailov, V.A., "Free Synchronous Packet Access in a Broadcast Channel with Feedback," *Problems of Information Transmission* 14, 4 (April 1978) 259-280 (translated from Russian Original in *Prob. Peredach. Inform.*, Oct-Dec 1977).

[28] Tsybakov, B.S. and Mikhailov, V.A., "Random Multiple Packet Access: Part and Try Algorithm," *Problems of Information Transmission*, 16, 4 (April 1981) 305-317 (translated from Russian Original in *Prob. Peredach. Inform.*, Oct-Dec 1980).

APPENDIX

We first give two technical lemmas about the functions

$$\psi(x) = \prod_{j=0}^{\infty} 1 - e^{-2^j x}\ (1 + 2^j x))$$

and

$$q(i+1, x) = \prod_{j=1}^{i} (1 - (1 - 2^{-j})^x - (1 - 2^{-j})^{x-1} 2^{-j} x)\ ,$$

which arose in the analysis of the simple estimation and hybrid algorithms. Let n be any integer, $n \geq 2$, and let $p = \lceil \log_2 n \rceil$.

Lemma A1:

Both $q(i+1, n)$ and $\psi(\frac{n}{2^i})$ are less than $2^{-(i-p)(i-p+1)}$.

Proof:

The j-th term in the product for $q(i+1, n)$ is

$$\begin{aligned} 1 - (1 - 2^{-j})^n - (1 - 2^{-j})^{n-1} 2^{-j} n &< 1 - (1 - 2^{-j})^n (1 + 2^{-j} n) \\ &< 1 - (1 - 2^{-j} n)\ (1 + 2^{-j} n) \\ &< 2^{-2(j-p)}\ . \end{aligned}$$

Therefore,

$$\begin{aligned} q(i+1, n) &< \prod_{j=p}^{i} (1 - (1 - 2^{-j})^n - (1 - 2^{-j})^{n-1} 2^{-j} n) \\ &< \prod_{j=p}^{i} 2^{-2(j-p)} = 2^{-(i-p)(i-p+1)}, \end{aligned}$$

as desired. Similarly, the typical term in the product for $\psi(\frac{n}{2^i})$,

$$1 - e^{-2^{-j}n} - e^{-2^{-j}n}\, 2^{-j}n < 2^{-2(j-p)},$$

which leads to the same bound for $\psi(\frac{n}{2^i})$. ■

Lemma A2:

The bound, $\psi(\frac{n}{2^i}) - q(i+1,n) = 0(\frac{\log n}{n})$, holds uniformly in i.

Proof:

If $i \geq 2p$ then

$$\psi(\frac{n}{2^i}) - q(i+1, n) < 2 \cdot 2^{-p(p+1)} = 0(\frac{\log n}{n}),$$

by Lemma A1. Suppose $i < 2p$. Further, suppose $\psi(\frac{n}{2^i}) > q(i+1, n)$. (A minor change to the argument is needed if $\psi(\frac{n}{2^i}) < q(i+1, n)$.) Break the product for $\psi(\frac{n}{2^i})$ in two:

$$\psi(\frac{n}{2^i}) = \prod_{j=1}^{i} (1 - e^{-2^{-j}n}(1 + 2^{-j}n)) \prod_{j=0}^{\infty} (1 - e^{-2^{-j}n}\, (1 + 2^{j}n)).$$

It is not hard to prove the crude estimate for the second product:

$$\prod_{j=0}^{\infty} (1 - e^{-2^{j}n}\, (1 + 2^{j}n)) = 1 + 0(e^{-n/2}).$$

By this bound, it suffices to simply prove

$$\prod_{j=1}^{i} (1 - e^{-2^{-j}n} - e^{-2^{-j}n}\, 2^{-j}n) - \prod_{j=1}^{i} (1 - (1 - 2^{-j})^n - (1 - 2^{-j})^{n-1}\, 2^{-j}n) = 0(\frac{\log n}{n}) .$$

It happens that corresponding terms in these last two products differ by less than $\frac{2}{n}$. Specifically,

$$|e^{-2^{-j}n} - (1 - 2^{-j})^n| + |e^{-2^{-j}n}\, 2^{-j}n - (1 - 2^{-j})^{n-1}\, 2^{-j}n| < \frac{2}{n} .$$

We prove this for the first term; the proof for the second is similar. Note

$$(1 - 2^{-j})^n < e^{-2^{-j}n}$$

and

$$(1 - 2^{-j})^n = e^{n \log(1 - 2^{-j})} > e^{-n(2^{-j} + 4^{-j})}$$

so

$$\begin{aligned} |e^{-2^{-j}n} - (1 - 2^{-j})^n| &< e^{-2^{-j}n}(1 - e^{-4^{-j}n}) \\ &< e^{-2^{-j}n}\, 4^{-j}n < \frac{1}{n} \end{aligned}$$

since $e^{-z} < z^{-2}$ for all $z > 0$. (Here $2^{-j}n$ plays the role of z.)

Returning to the main argument, these observations imply

$$\begin{aligned} &\prod_{j=1}^{i} (1 - e^{-2^{-j}n}\, (1 + 2^{-j}n)) - \prod_{j=1}^{i} (1 - s(j, n)) \\ &< \prod_{j=1}^{i} (1 - s(j, n) + \frac{2}{n}) - \prod_{j=1}^{i} (1 - s(j, n)) , \end{aligned}$$

where $s(j, n) = (1 - 2^{-j})^n - (1 - 2^{-j})\, 2^{-j}n$. Factoring out $\prod_{j=1}^{i} (1 - s(j, n))$, which is

less than 1, we obtain

$$\prod_{j=1}^{i}(1 - s(j,n) + \frac{2}{n}) - \prod_{j=1}^{i}(1 - s(j,n)) < (\prod_{j=1}^{i}(1 + \frac{2}{n(1-s(j,n))})) - 1$$

$$< (1 + \frac{2}{n(1-s(2p,n))})^{2p} - 1$$

$$= 0(\frac{p}{n}) = 0(\frac{\log n}{n}) ,$$

as desired. ■

The next lemma concerns the behavior of the integral

$$\psi_a^*(s) = \int_0^\infty \psi_a(x)\, x^{s-1}\, dx$$

$$= \int_0^\infty \prod_{k=0}^{\infty}(1 - e^{-a^k x}\,(1 + a^k x))\, x^{s-1}\, dx$$

as a tends to 1.

Lemma A3:

For any fixed real $s < 0$, as $a \to 1$, one has

$$\psi_a^*(s) \sim -\frac{1}{s}(\log \frac{1}{a-1})^s .$$

Proof:

We shall only briefly sketch the main steps of the evaluation of $\psi_a(s)$.

(i) The contribution of the integral between 0 and 1 is exponentially small as $a \to 1$:

$$\psi_a^*(s) = \int_1^\infty \prod_{k=0}^{\infty}(1 - e^{-a^k x}\,(1 + a^k x))\, x^{s-1}\, dx + 0(e^{-c/(a-1)})$$

for some constant $c > 0$. This follows easily from the fact that for $x < 1$, the infinite

product comprises $0(\frac{1}{a-1})$ factors each bounded above by $t(2)$ where $t(u)$ is the increasing function $(1 - e^{-u}(1+u))$. Hence the product is exponentially small in $1/(a-1)$ for any fixed x in $[0, 1]$.

(ii) Towards evaluating the integrand for $x \geq 1$ consider

$$-\log \psi_a(x) = \sum_{j=0}^{\infty} \lambda(xa^j) \qquad \textbf{(A1)}$$

where $\lambda(u) = -\log(1 - (1+u)\, e^{-u})$, a function which is positive when $u > 0$. Setting $a = e^t$, as $a \to 1$ (*i.e.* $t \to 0$), $-\log \psi_a(x)$ can be approximated by means of the Euler-McLauren summation formula, so that

$$-\log \psi_a(x) = \frac{1}{t} \int_0^{\infty} \lambda(xe^u)\, du + \frac{1}{2}\lambda(x) + 0(t)$$

where the error term is uniform in x for $x \geq 1$, being of the form $0(e^{-x}t)$. Thus,

$$\psi_a^*(s) = \int_1^{\infty} e^{\frac{-\Lambda(x)}{t} - 1/2\lambda(x) + O(e^{-x}t)}\, x^{s-1} dx + O(e^{-K/(a-1)}) \qquad \textbf{(A2)}$$

where

$$\Lambda(x) = \int_0^{\infty} \lambda(xe^u)\, du = \int_x^{\infty} \lambda(u)\, \frac{du}{u},$$

and K is a constant ($K > 0$). Taking out the error terms in (A2), we obtain

$$\psi_a^*(s) = J_a(s)\,(1 + 0(\log a)) + 0(e^{-K/(a-1)})$$

where

$$J_a(s) = \int_1^{\infty} e^{-\frac{\Lambda(x)}{t} - 1/2\,\lambda(x)}\, x^{s-1}\, dx\,. \qquad \textbf{(A3)}$$

The integral in (A3) is of the type classically evaluated by means of the Laplace method. We set $v = \Lambda(x)$, and find:

$$J_a(s) = -\int_0^{\Lambda(1)} e^{-v/t}\, w(v)\, dv \qquad \text{(A4)}$$

where

$$w(v) = \frac{e^{-\frac{1}{2}\lambda(\Lambda^{(-1)}(v))}}{\Lambda'(\Lambda^{(-1)}(v))} (\Lambda^{(-1)}(v))^{s-1} .$$

The main contribution in (A4) comes in the neighborhood of 0, since for any fixed v the integrand in (A4) decreases as $e^{-v/t}$, that is, exponentially fast in t as $t \to 0$. Splitting the integration interval, we find that

$$J_a(s) = -\int_0^{t \log^2 t} e^{v/t}\, w(v)\, dv + 0(e^{-\log^2 t}) \qquad \text{(A5)}$$

Now the integral in (A5) can be evaluated by expanding $w(v)$ in the vicinity of 0; from the expansion

$$w(v) = \frac{1}{v} (-\log v)^{s-1} (1 + o(1))$$

one finds that

$$J_a(s) \sim \int_0^{t \log^2 t} e^{-v/t} (-\log v)^{s-1} \frac{dv}{v}$$

$$\sim \int_0^{\log^2 t} e^{-y} (-\log t - \log y)^{s-1}\, dy$$

which through integration by parts leads to

$$J_a(s) \sim -\frac{1}{s} (-\log t)^s \int_0^{\log^2 t} e^{-y} (1 + \frac{\log y}{\log t})^s\, dy.$$

The last integral has its dominant contribution concentrated in the interval $[\sqrt{t}, \log^2 t]$, where, expanding $(1 + \frac{\log y}{\log t})^s$ in powers of $\frac{\log y}{\log t}$, one finds

$$J_a(s) \sim \frac{1}{s} (-\log t)^s \int_{\sqrt{t}}^{\log^2 t} e^{-y}\, dy \sim \frac{1}{s} (-\log t)^s . \qquad \textbf{(A6)}$$

Since $t = \log a$, we have $t \sim a - 1$ as $a \to 1$ and (A6) is equivalent to the statement of the lemma. ■

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique