Research Report 285

Contention Resolution with Bounded Delay

MS Paterson and A Srinivasan

RR285

When many distributed processes contend for a single shared resource that can service at most one process per time slot, the key problem is devising a good *distributed* protocol for contention resolution. This has been studied in the context of multiple-access channels (*e.g.*, ALOHA, Ethernet), and recently for PRAM emulation and routing in optical computers. Under a stochastic model of continuous request generation from a set of n synchronous processes, Raghavan & Upfal have recently shown a protocol which is stable if the request rate is at most λ_0 for some fixed $\lambda_0 < 1$; their main result is that for any given resource request, its *expected delay* (expected time to get serviced) *is* $O(\log n)$. Assuming further that the initial clock times of the processes are within a known bound B of each other, we present a stable protocol, again for some fixed positive request rate λ_1, $0 < \lambda_1 < 1$, wherein the *expected delay for each request is* $O(1)$, *independent of* n. We derive this by showing an analogous result for an infinite number of processes, assuming that all processes agree on the time; this is the first such result. We also present tail bounds which show that for every given resource request, it is unlikely to remain unserviced for much longer than expected, and extend our results to other classes of input distributions.

Department of Computer Science
University of Warwick
Coventry CV4 7AL
United Kingdom

April 1995

Contention Resolution with Bounded Delay*

Mike Paterson † Aravind Srinivasan‡

Abstract

When many distributed processes contend for a single shared resource that can service at most one process per time slot, the key problem is devising a good *distributed* protocol for contention resolution. This has been studied in the context of multiple-access channels (*e.g.*, ALOHA, Ethernet), and recently for PRAM emulation and routing in optical computers. Under a stochastic model of continuous request generation from a set of n synchronous processes, Raghavan & Upfal have recently shown a protocol which is stable if the request rate is at most λ_0 for some fixed $\lambda_0 < 1$; their main result is that for any given resource request, its *expected delay* (expected time to get serviced) *is* $O(\log n)$. Assuming further that the initial clock times of the processes are within a known bound B of each other, we present a stable protocol, again for some fixed positive request rate λ_1, $0 < \lambda_1 < 1$, wherein the *expected delay for each request is* $O(1)$, *independent of* n. We derive this by showing an analogous result for an infinite number of processes, assuming that all processes agree on the time; this is the first such result. We also present tail bounds which show that for every given resource request, it is unlikely to remain unserviced for much longer than expected, and extend our results to other classes of input distributions.

1 Introduction

In scenarios where a set of distributed processes have a single shared resource that can service at most one process per time slot, the main problem is devising a "good" *distributed* protocol for resolving contention for the resource by the processes. This has traditionally been studied in the context of multiple-access channels (*e.g.*, ALOHA) and for Ethernet protocols, and more recently for PRAM emulation and for routing in optical computers. Assuming a stochastic model of continuous request generation from a set of n synchronous processes (see Section 1.1 for the formal definition), Raghavan & Upfal have very recently shown a protocol which is stable as long as the request rate is at most λ_0 for some fixed $\lambda_0 < 1$ [18]; their main result is that for any given resource request, its *expected delay* (expected time until it is serviced) *is* $O(\log n)$. Assuming further that the initial clock times of the processes are within a known bound B of each other, we present a stable protocol again for some fixed positive request rate λ_1, wherein the *expected delay for each request is* $O(1)$, *independent of* n. We derive this by showing an analogous result for an infinite number of processes (which is a model for processes entering and leaving dynamically), assuming that all processes agree on the time; this is the first such result. We also present tail bounds which show that for every given resource request, it is unlikely to remain unserviced for much longer than expected, and extend our results to various classes of input distributions.

*Supported in part by the ESPRIT Basic Research Action Programme of the EC under contract No. 7141 (project ALCOM II).

†Department of Computer Science, University of Warwick, Coventry CV4 7AL, England (msp@dcs.warwick.ac.uk).

‡Max-Planck-Institut für Informatik, Im Stadtwald, 66123 Saarbrücken, Germany (srinivas@mpi-sb.mpg.de). Most of the work was done while this author was visiting the Department of Computer Science, University of Warwick, Coventry CV4 7AL, England. Part of this work was done while visiting the Max-Planck-Institut für Informatik.

1.1 Model and motivation

In multiple-access channels (MACs), there is one channel (resource) shared by a finite or infinite number of synchronized senders (*i.e.*, each sender's local clock ticks at the same rate as the others' clocks). Time is slotted into units of time starting at 0, and in each unit of time, each sender may receive some packets according to some distribution. Each sender which has packets will have to send its packets (one at a time) to the channel, but if more than one sender attempts to transmit at the same time slot, the packets collide and are not received by the sender. Otherwise if exactly one packet was sent in a slot, it is received by the channel which sends the corresponding sender an acknowledgement. Thus if a sender did not receive an acknowledgement for a packet sent, it knows that there was a collision, and must try again; it is natural to expect randomized protocols to play a key role in this. This model was initiated by work on ALOHA, a multi-user communication system based on radio-wave communication (Abramson [1]), and a similar situation arises in Ethernet protocols (Metcalf & Boggs [15]). Much research on MACs was spurred by ALOHA, especially in the information theory community; see, *e.g.*, the special issue of *IEEE Trans. Info. Theory* on this topic [12]. Recently such resource-allocation problems have arisen again, in the context of PRAM emulation–running PRAM algorithms on more realistic models of parallel computation–and in message routing in optical computers. These parallel models include optical networks (Anderson & Miller [4], Geréb-Graus & Tsantilas [7], Goldberg, Jerrum, Leighton & Rao [8]), DMM models (Dietzfelbinger & Meyer auf der Heide [6]), and Valiant's S*PRAM model [20]; see MacKenzie, Plaxton and Rajaraman [14] for details. In addition, MACs provide a good model to study the abstract problem of distributed contention resolution for a common shared resource. All these definitions can easily be extended to the case of more than one channel (shared resource).

Rather than the static case (see below), we will be interested in the dynamic scenario of packets arriving into the system at every time step, according to some distribution. There are two important parameters for a MAC protocol–the *arrival rate* λ of packets into the system (the expected number of new arrivals per unit time), and *stability*. To define stability, suppose $W(P)$ is the random variable measuring the amount of time a packet P spends in the system (before being sent successfully to the channel). Then let the random variable W_{ave} be the limit as $i \to \infty$ of the arithmetic mean of $W(P)$ for the first i packets arriving into the system. Similarly, we may define the random variable L_{ave} as the limit as $i \to \infty$ of $1/i$ times the sum of the number of waiting packets in the first i steps. Finally, we may define T_{ret} to be the time taken to have all sender queues empty, if we start from an arbitrary state of the system (weighted by the probability of being in such a state). Unifying several previous definitions, Håstad, Leighton & Rogoff define a protocol to be stable if and only if all three of $E[W_{ave}]$, $E[L_{ave}]$, and $E[T_{ret}]$ are finite [9]. Actually, $L_{ave} = \lambda W_{ave}$ with probability one and similarly, the *throughput rate* (average rate of successful transmissions) equals λ with probability one, for a stable protocol [9].

We must distinguish a few models when defining the problem further. First, we might have a finite or infinite number of senders. In the former case, there are n senders into which there is a continuous influx of packets; at most one packet arrives per sender, in any given time step. The usual assumption is that these arrivals are independent across different time steps and across different senders, and that the expected total arrival per time step is at most λ. The infinite case is a natural extension of this, with a random number of packets arriving with a Poisson distribution of mean λ, independently at each step. Here, each packet may be regarded as a sender in itself.

The next key feature is the type of acknowledgement sent by the channel to the senders. A popular model used in the information theory literature for this is that of *ternary feedback*: at the end of each time slot, *each sender* receives information on whether zero, one, or more than one packets were sent to the channel at that time step. In this case, stable protocols are known for $\lambda \leq 0.4878...$ (Vvedenskaya & Pinsker [21]), and there is no stable protocol for $\lambda \geq 0.587...$ (Mikhailov & Tsybakov [16]); but if the stronger feedback of the exact number of packets that tried at the current step is sent to each sender, then there is a stable protocol for all $\lambda < 1$ (Pippenger [17]). A weaker feedback model which is more realistic for the purposes of PRAM emulation and optical routing is *acknowledgement-based*, wherein the only information known to each sender which attempted to send a packet, is whether it succeeded or not; idle senders get no information. The acknowledgement-based feedback model is thus a minimal-information model, and we follow [9, 14, 18] in focussing on this henceforth.

The above classifications dealt with the dynamic situation of packet arrivals at every step. Alternatively, we may consider a *static* scenario where at most h of n senders have a packet each to send to the channel; the problem then is to design a distributed protocol (wherein each sender only knows the value h, and whether it has a packet to send or not) for this. Assuming acknowledgement-based feedback, the work of [14], among other things, improves on previous work to provide near-optimal bounds for various problems relating to the static version; in the optical routing case, a similar problem is termed *h-relation routing*, for which the best known bounds are due to [8].

Since the static case is fairly well-understood, we focus only on the dynamic case in this work.

1.2 Previous work

For our model of interest–the dynamic setting with acknowledgement-based protocols, only negative results were known in the *infinite senders* case: while Kelly showed that a large class of protocols (including "polynomial backoff") are unstable for *any* $\lambda > 0$ [13], Aldous extended this to the case of binary exponential backoff, the Ethernet protocol [2]. Also, any stable protocol in the infinite case must have $\lambda < 0.587...$ [16].

In striking contrast to Kelly's result, the important work of [9] showed, among other things, that in the *finite senders* case, most polynomial backoff protocols are stable, in fact *for all* $\lambda < 1$. However, their proven upper bound for $E[W_{ave}]$ is $2^{f(n)}$, where $f(n) = n^{O(1)}$. With applications such as high-speed communications in mind where the average delay $E[W_{ave}]$ needs to be kept small, the very recent work of [18] presented a protocol for the finite case, which is stable for all $\lambda < \lambda_0$ ($\sim 1/10$, using the analysis of [18]), with the key property that $E[W_{ave}] = O(\log n)$, after an initial setup time of $n^{O(1)}$ steps (note the significant reduction in $E[W_{ave}]$). Moreover, it is shown in [18] that for each element $\mathcal{P}$ of a large set of protocols that includes all known backoff protocols, there exists a threshold $\lambda_{\mathcal{P}} < 1$ such that if $\lambda > \lambda_{\mathcal{P}}$, then $E[W_{ave}] = \Omega(n)$ must hold.

1.3 Our results

In the finite senders case, we take a further step in the direction of [18], with the view that $E[W_{ave}]$ must be kept low. Recall that the known results use the fact that the senders' clocks all tick at the same rate. Under the additional assumption that the clocks of the n senders differ by at most a known bound of B time steps, we present a protocol that has $E[W_{ave}] = O(1)$ for $\lambda < \lambda_1 \in (0, 1)$, *independent of* n, after a setup time of $O(B \log B + n \log B \log(n \log B))$ steps. Our result and that in [18] have a stability property stronger than that defined in [9],

in that for *every packet* P, the expected waiting time for P is $O(1)$ (resp., $O(\log n)$ in [18]).

In our view, this assumption on the clock differences is reasonable, since clocks–especially those within a local enough area to be able to share a common resource–usually agree to within, say 15 minutes. (With the same motivation, Hui & Humblet consider a somewhat different problem [11].) Another way of looking at this result is that since the expected waiting time for packets is a crucial parameter, yet another payoff is seen for building accurate clocks.

Our above result is actually shown quite easily from the main result we prove, which is a stable protocol for the infinite case as long as $\lambda < \lambda_1$, assuming that all senders agree on the time. Thus, this additional assumption on accurate clocks presents the first stable acknowledgement-based protocol for the infinite case. The infinite case is of interest too, since it models situations where senders may enter and leave the system, with no known reasonable bound on the number n of competing processes. An interesting point here is that our results are complementary to those of [9]–while the work of [9] shows that (negative) results for the infinite case may have no bearing on the finite case, our results suggest that better intuition and positive results for the finite case may be obtained by passing to the infinite case.

Our protocols are simple. We show an explicit, easily computable collection $\{S_{i,t} : i, t = 0, 1, 2, \ldots\}$ of finite sets of nonnegative integers $S_{i,t}$; for all i and t, every element of $S_{i,t}$ is smaller than every element of $S_{i+1,t}$. A packet born at time t and which has made i (unsuccessful) attempts at the channel so far, picks a time r uniformly at random from $S_{i,t}$, and tries using the channel at time r. If it succeeds, it leaves the system; else if it fails, it increments i and repeats this process. We also show good upper tail bounds on $W(P)$ for every packet P: $\forall a > 0$, $Pr(W(P) \geq a) = O(a^{-c_1})$, where $c_1 > 1$ is a constant. Thus for our protocol, the expected number of packets (and hence the total storage size) in the system at any given time is $O(1)$, improving on the $\Omega(\log n)$ bound of [18]. Finally, we extend our results to various input distributions to show that our protocol is robust to fairly "non-random" distributions with weak tail properties.

Thus we show that the expected waiting times of packets can be reduced to just $O(1)$, if reliable clocks are available. In the infinite senders case, we ask for all clocks to agree; this gives us the first stable acknowledgement-based protocol. In the case of finite senders, it suffices if there is a known upper bound on the time differences between the clocks.

2 Notation and Preliminaries

For any positive integer ℓ, we denote the set $\{1, 2, \ldots, \ell\}$ by $[\ell]$. Theorem 1 presents the Chernoff-Hoeffding bounds [5, 10]; see, *e.g.*, Appendix A of [3] for details.

Theorem 1 *Let R be a random variable with $E[R] = \mu \geq 0$ such that either: (a) R is a sum of a finite number of* independent *random variables $X_1, X_2, \ldots$ with each X_i taking values in $[0, 1]$, or (b) R is Poisson. Then for any $\nu \geq 1$, $Pr(R \geq \mu\nu) \leq H(\mu, \nu) \doteq (e^{\nu-1}/\nu^\nu)^\mu$.*

Fact 1 is easily verified. Lemma 1 is "folklore"; its proof is shown in the appendix.

Fact 1 *If $\nu > 1$ then $H(\mu, \nu) \leq e^{-\nu\mu/M_\nu}$, where M_ν is positive and monotone decreasing for $\nu > 1$.*

Lemma 1 *Suppose $X_1, X_2, \ldots, X_\ell$ and $Y_1, Y_2, \ldots, Y_\ell$ are sequences of $\{0, 1\}$ random variables, such that: (a) $\forall i\ \forall b_1, b_2, \ldots, b_{i-1} \in \{0, 1\}$, $Pr(X_i = 1 | \bigwedge_{j \in [i-1]}(X_j = b_j)) \leq Pr(Y_i = 1)$, and (b) $Y_1, Y_2, \ldots, Y_\ell$ are* independent. *Then for any $c \geq 0$, $Pr(\sum_{i \in [\ell]} X_i \geq c) \leq Pr(\sum_{i \in [\ell]} Y_i \geq c)$. Thus if $c \geq E[\sum_{i \in [\ell]} X_i]$, then $Pr(\sum_{i \in [\ell]} X_i \geq c) \leq H(E[\sum_{i \in [\ell]} X_i], c/E[\sum_{i \in [\ell]} X_i])$.*

Remark. If, for a pair of real-valued random variables A and B, it is true that for all c, $Pr(A \geq c) \leq Pr(B \geq c)$, then A is said to be *stochastically dominated* by B.

Suppose (at most) s packets are present in a static system, and that we have ℓ time units within which we would like to send out a "large" number of them to the channel, with high probability (*w.h.p.*). Then a natural scheme is for each packet to attempt using the channel at a randomly chosen time from $[\ell]$. (There are schemes which provide better constants than this, and we have not yet attempted to optimize the constants.) In fact, we will also need a version of such a scenario where some number z of such experiments are run *independently*, as considered by Lemma 2. Since a packet is successful if and only if no other packet chose the same time slot as it did, the "collision" of packets is a dominant concern. The proof of Lemma 2 is shown in the appendix.

Lemma 2 *Suppose we run z* independent *experiments $E_1, E_2, \ldots, E_z$ where in E_i, a set U_i of at most s balls are thrown uniformly and independently at random into a set V_i of ℓ bins; if $i \neq j$, then $U_i \cap U_j = \phi$ and $V_i \cap V_j = \phi$. Let us say that a ball "collides" if and only if it is not the only ball in its bin. Then, (i) For any given ball B, $Pr(B \text{ collides}) \leq 1 - (1 - 1/\ell)^{s-1} < s/\ell$. (ii) If C denotes the total number of balls that collided, then $\forall \nu > 1$, $Pr(C \geq zs^2\nu/\ell) \leq H(zs^2\nu/(2\ell))$.*

3 Tree model for contention resolution: the infinite case

We assume a time-slotted system. At every time slot $t = 0, 1, 2, \ldots$, a random number of packets whose distribution is Poisson with mean $\lambda < 1$, is injected into the system; the arrivals are independent for different time slots. We assume that all the packets agree on a common global time; there is no common knowledge (or inter-packet communication) apart from this. At every time slot, each packet in the system will decide autonomously, based on its current time, its time of entry into the system, its history of unsuccessful attempts in the past, and on the outcome of its internal coin flips, to try using the channel or not. If it succeeds, then the packet leaves the system; if not, the only information it has gained is that it tried at this current time but failed due to a collision.

We present the ideas parametrized by several constants. Later on we will choose values for the parameters to maximize the throughput. There will be a trade-off between the maximum throughput and the expected waiting time for a packet; a different choice of parameters could take this into consideration. The constants we have chosen for ease of presentation, guarantee that our protocol is stable for $\lambda < 1/32$. In the final version, we will present a more complicated choice of constants for which $\lambda < 1/16$ will suffice for stability.

3.1 The tree protocol

Three important positive constants, b, r and k, where $b \geq 1$ and $r, k > 1$, shape the protocol. At any time during its lifetime in the protocol, a packet is regarded as residing at some node of a tree T that has an infinite number of leaves. Each non-leaf node of T has exactly k children, where

$$k > r\,. \tag{1}$$

T is not actually constructed–it is just for exposition. The nodes of T at the same height i for any $i \geq 0$, are ordered left-to-right. We associate a finite nonempty set of non-negative integers *Trial*(v) with each node v. Define $L(v) \doteq \min\{\textit{Trial}(v)\}$, $R(v) \doteq \max\{\textit{Trial}(v)\}$, and the *capacity* $cap(v)$ of v, to be $|\textit{Trial}(v)|$. A required set of properties of the *Trial*$(\cdot)$ sets is the following:

P1. If u and v is any pair of *distinct* nodes of T, then $\mathit{Trial}(u) \cap \mathit{Trial}(v) = \phi$;

P2. If u is either a proper descendant of v, or if u and v are at the same height with u to the left of v, then $R(u) < L(v)$.

P3. The capacity of all nodes at the same height is the same. Let u_i be a generic node at height i. Then, $cap(u_0) = b$ and $cap(u_i) = \lceil r\ cap(u_{i-1}) \rceil$, for $i \geq 1$. (Thus, $cap(u_i) = br^i$ if r is integral; otherwise, $br^i \leq cap(u_i) \leq br^i + (r^i - 1)/(r - 1)$.)

Suppose we have such a construction of the $\mathit{Trial}(\cdot)$ sets. Each packet P injected into the system at time slot t_P will initially enter the leaf node $u_0(P)$ where $u_0(P)$ is the leftmost leaf such that $L(u_0(P)) > t_P$. Then P will move up the tree if necessary to successor (*parent*) nodes of increasing height, in the following way. In general, suppose P enters a node $u_i(P)$ at height i, at time t_i; we will be guaranteed the invariant "**Q:** $u_i(P)$ is an ancestor of $u_0(P)$, and $t_i < L(u_i(P))$." P will then pick an element $r_i \in \mathit{Trial}(u_i(P))$ uniformly at random, and try using the channel at time r_i. If it was successful, P will (of course) leave the system, otherwise it will enter the parent $u_{i+1}(P)$ of $u_i(P)$, at time r_i. (**Q**) is established by an easy induction on i, using property (P2). Note that the packets entering any given node v conduct the same experiment as is considered by Lemma 2, with $z = 1$. More importantly, if v is any non-leaf node, then the trials at its k children correspond to $z = k$ in Lemma 2, by (P1).

Thus, each node receives all the unsuccessful packets from each of its k children; an unsuccessful packet is imagined to enter the parent of a node u, immediately after it found itself unsuccessful at u. Informally, if the proportion of the time dedicated to height 0 is $1/s$, where $s > 1$, then the proportion for height i will be approximately $(r/k)^i/s$. Since the sum of these proportions for all i can be at most 1, we have $s \geq k/(k - r)$; we will take

$$s = k/(k - r)\ . \tag{2}$$

More precisely, the $\mathit{Trial}(\cdot)$ sets are constructed as follows; it will be immediate that they satisfy (P1,P2,P3). First define

$$k = 16,\ s = 2,\ \text{and}\ r = 8. \tag{3}$$

We remark that though we have fixed these constants, we will use the symbols k, s and r (rather than their numerical values) wherever possible, to retain generality.

For $i = 0, 1, \ldots$, let $F_i = \{j \geq 0 : j \equiv 2^i \pmod{2^{i+1}}\}$; the sets F_i form a partition of Z^+, the set of non-negative integers. Let v_i be a generic node at height i; if it is not the leftmost node in its level, let u_i denote the node at height i that is immediately to the left of v_i. We will ensure that all elements of $\mathit{Trial}(v_i)$ lie in F_i. (For any large enough interval I in Z^+, the fraction of F_i lying in I is roughly $1/2^{i+1} = (r/k)^i/s$; this was what we meant informally above.)

We now define $\mathit{Trial}(v_i)$ by induction on i and from left-to-right within the same level, as follows. If $i = 0$, then if v_0 is the leftmost leaf, we set $\mathit{Trial}(v_0)$ to be the smallest $cap(v_0)$ elements of F_0; else we set $\mathit{Trial}(v_0)$ to be the $cap(v_0)$ smallest elements of F_0 larger than $R(u_0)$. If $i \geq 1$, let w be the rightmost child of v_i. If v_i is the leftmost node at height i, we let $\mathit{Trial}(v_i)$ be the $cap(v_i)$ smallest elements of F_i that are larger than $R(w)$; else define $\mathit{Trial}(v_i)$ to be the $cap(v_i)$ smallest elements of F_i that are larger than $\max\{R(u_i), R(w)\}$. In fact, it is easy to show by the same inductive process that, if u_i is defined, then $R(w) > R(u_i)$; hence for every node v_i with $i \geq 1$,

$$L(v_i) \leq R(w) + 2^{i+1} = R(w) + s(k/r)^i. \tag{4}$$

3.2 Waiting times of packets

We now come to our main random variable of interest: the time that a generic packet P will spend in the system, from its arrival. We need some definitions, where a, d are constants greater than 1.

Definition 1 *For any node $v \in T$, the random variable load(v), the* load of v, *is defined to be* the number of packets that enter v; *for any positive integer t, v is defined to be* t-bad *if and only if load(v) $> br^i d^{t-1}/a$. Node v is said to be* t-loaded *if it is t-bad but not $(t+1)$-bad. It is called* bad *if it is 1-bad, and* good *otherwise.*

It is not hard to verify that for any given $t \geq 1$, the probability of being t-bad is the same for any pair of nodes at the same level in T; this brings us to the next definition.

Definition 2 *For any (generic) node u_i at height i in T and any positive integer t, $p_i(t)$ denotes the probability that u_i is t-bad.*

Definition 3 *(i) The* failure probability q *is the maximum probability that a packet entering a* good *node will* not *succeed during the functioning of that node. (ii) For any packet P, let $u_0(P), u_1(P), u_2(P), \ldots$ be the nodes of T that u_i is allowed to pass through, where the height of $u_i(P)$ is i. Let $E_i(P)$ be the event that P enters $u_i(P)$.*

If a node u at height i is good, then in the notation of Lemma 2, $s \leq \ell/a$, where $\ell = cap(u)$; hence, Lemma 2(i) shows that

$$q < 1/a\,. \tag{5}$$

Note that *the distribution of $E_i(P)$ is independent of its argument.* Hence, for any $i \geq 1$, we may define $e_i \doteq Pr(E_i(P))$ for a generic packet P, with $e_0 \equiv 1$. Suppose P was unsuccessful at nodes $u_0(P), u_1(P), \ldots, u_i(P)$. Let $A(i)$ denote the maximum total amount of time P could have spent in these $(i+1)$ nodes. Then, it is not hard to see that $A(0) \leq s\,cap(u_0) + s\,cap(u_0) = 2sb$ and that for $i \geq 1$, $A(i) \leq kA(i-1) + (k/r)^i sbr^i$, using (4). Hence,

$$A(i) \leq (i+1)sbk^i \text{ for all } i. \tag{6}$$

Lemma 3 is about the distribution of the crucial random variable $W(P)$ the time that P spends in the system. See the appendix for its proof (which is simple, but crucial).

Lemma 3 *(i) For any packet P and for all $i \geq 0$, $Pr(W(P) > A(i)) \leq e_{i+1}$; also, $E[W(P)] \leq \sum_{i=0}^{\infty} A(i)e_i$. (ii) For all $i \geq 1$, $e_i \leq qe_{i-1} + p_{i-1}(1)$.*

3.3 The improbability of high nodes being heavily loaded

As is apparent from Lemma 3, our main interest is in getting a good upper bound on $p_i(1)$. However, to do this we will also need some information about $p_i(t)$ for $t \geq 2$, and hence Definition 2. The basic intuition is that if a node is good, then *w.h.p.*, it will successfully schedule "most" of its packets; this is formalized by Lemma 2, by setting $z = 1$. In fact, Lemma 2 shows that for any node u in the tree, the *good* children of u will, *w.h.p.*, pass on a total of "not many" packets to u, since the functioning of each of these children is independent of the other children.

To estimate $p_i(t)$, we first handle the easy case of $i = 0$. Recall that if X_1 and X_2 are independent Poisson random variables with means λ_1 and λ_2 respectively, then $X_1 + X_2$ is Poisson with mean $\lambda_1 + \lambda_2$. Thus, u_0 being t-bad is a simple large-deviation event for a

Poisson random variable with mean $sb\lambda$. If, for every $t \geq 1$, we define $\nu_t \doteq d^{t-1}/(sa\lambda)$ and ensure that $\nu_t > 1$ by setting

$$sa\lambda < 1, \tag{7}$$

then Theorem 1 shows that

$$p_0(t) = Pr(u_0 \text{ is } t\text{-bad}) \leq H(sb\lambda, \nu_t). \tag{8}$$

We now consider how a generic node u_i at height $i \geq 1$ could have become t-bad, for any given t. The resulting recurrence yields a proof of an upper bound for $p_i(t)$ by induction on i. The two cases, $t \geq 2$ and $t = 1$, are covered by Lemmas 4 and 5 respectively. We now require

$$d^2 + k - 1 \leq dr\ . \tag{9}$$

Remark. Lemma 4 can be strengthened, but we present this version for simplicity.

Lemma 4 *Suppose $d^2 + k - 1 \leq dr$. Then for $i \geq 1$ and $t \geq 2$, if a node u_i at height i in T is t-bad, then at least one of the following two conditions holds, for u_i's set of children. (i) At least one child is $(t+1)$-bad, or (ii) at least 2 children are $(t-1)$-bad. Thus,*

$$p_i(t) \leq kp_{i-1}(t+1) + \binom{k}{2} (p_{i-1}(t-1))^2\ .$$

The proofs of Lemmas 4 and 5 are shown in the appendix. We now consider the case that $t = 1$ where the intuition, that the good children of u_i can be expected to have successfully transmitted much of their load, plays a key role. We now require

$$a(r-d) > k - 1\ . \tag{10}$$

Lemma 5 *If $a(r-d) > (k-1)$, then for any $i \geq 1$,*

$$p_i(1) \leq kp_{i-1}(2) + \binom{k}{2} (p_{i-1}(1))^2 + kp_{i-1}(1) H\left(\frac{(k-1)br^{i-1}}{2a^2}, \frac{a(r-d)}{k-1}\right) + H\left(\frac{kbr^{i-1}}{2a^2}, \frac{ar}{k}\right).$$

We now present a key theorem that proves an upper bound for $p_i(t)$, by induction on i. We assume that our constants satisfy the conditions (1, 2, 7, 9, 10).

Theorem 2 *There exists a constant $\lambda_1 > 0$ such that for a sufficiently large value of b the following holds for $\lambda \leq \lambda_1$. There are positive constants α, β and γ, with $\alpha, \beta > 1$, such that*

$$\forall i \geq 0\ \forall t \geq 1,\ p_i(t) \leq e^{-\gamma \alpha^i \beta^{t-1}}.$$

Before proceeding to the proof of Theorem 2, let us see why it shows the required property that $E[W(P)]$, the expected waiting time of a generic packet P in the system, is finite. Theorem 2 shows that for large i, $p_{i-1}(1)$ is negligible compared to q^i and hence, by Lemma 3(ii), $e_i = q^i(1 + o(1))$, where the $o(1)$ term goes to zero as i tends to infinity. Hence, Lemma 3(i) combined with bound (6) shows that as long as we can ensure that $q < 1/k$, then $E[W(P)]$ is finite (and, in fact, that good upper tail bounds can be proven for the distribution of $W(P)$). Combining this with (5), all we need is to pick a large enough so that

$$a > k\ . \tag{11}$$

Proof of Theorem 2 Induction on i. If $i = 0$, we use inequality (8) and require that

$$H(sb\lambda, \nu_t) \leq \epsilon^{-\gamma\beta^{t-1}}. \tag{12}$$

From (7), we see that $\nu_t > 1$; thus by Fact 1, there is some $M > 0$, $M \leq M_{\nu_1}$, such that $H(sb\lambda, \nu_t) \leq \epsilon^{-\nu_t sb\lambda/M}$. Therefore to satisfy inequality (12), it suffices to ensure that $d^{t-1}b/(aM) \geq \gamma\beta^{t-1}$. We will do this by choosing our constants so as to satisfy:

$$d \geq \beta \text{ and } b \geq \gamma aM\ . \tag{13}$$

We will choose α and β to be fairly close to (but larger than) 1, and so the first inequality will be satisfied. Although γ will have to be quite large, we will be free to choose b sufficiently large to satisfy the second inequality.

We proceed to the induction for $i \geq 1$. We first handle the case $t \geq 2$, and then the case $t = 1$.

Case I: $t \geq 2$. By Lemma 4, it suffices to show that $ke^{-\gamma\alpha^{i-1}\beta^t} + \binom{k}{2}e^{-2\gamma\alpha^{i-1}\beta^{t-2}} \leq \epsilon^{-\gamma\alpha^i\beta^{t-1}}$. It is easy to verify that this holds for some sufficiently large γ, provided

$$\beta > \alpha \text{ and } 2 > \alpha\beta\ . \tag{14}$$

We can pick $\alpha = 1 + \epsilon$ and $\beta = 1 + 2\epsilon$ for some small positive ϵ, $\epsilon < 1$, to satisfy (14).

Case II: $t = 1$. The first term in the inequality for $p_i(1)$ given by Lemma 5, is the same as for Case I with $t = 1$; thus it can be assumed to be much smaller than $e^{-\gamma\alpha^i}$ by an appropriate choice of constants, as seen above. Similarly, the second term in the inequality for $p_i(1)$ can be handled by assuming that $\alpha < 2$ and that γ is large enough, which again has been handled above. The final two terms given by Lemma 5 sum to

$$ke^{-\gamma\alpha^{i-1}} H\left(\frac{(k-1)br^{i-1}}{2a^2}, \frac{a(r-d)}{k-1}\right) + H\left(\frac{kbr^{i-1}}{2a^2}, \frac{ar}{k}\right). \tag{15}$$

We wish to make each summand in (15) at most, say, $e^{-\gamma\alpha^i}/4$, using Fact 1; we now examine some sufficient conditions for these to hold. Let M' be a sufficiently large constant for both applications of Fact 1. Then by Fact 1, we just need to ensure that

$$\frac{br^{i-1}(r-d)}{2aM'} \geq \gamma\alpha^i + \ln(4k) \text{ and } \frac{br^i}{2aM'} \geq \gamma\alpha^i + \ln 4\ . \tag{16}$$

Both of these are true for sufficiently large i, since $r > \alpha$. To satisfy these inequalities for small i, we choose b sufficiently large to satisfy (16,13), completing the proof of Theorem 2. □

Finally, we can choose

$$d = 4.$$

It is now easily verified that conditions (1,2,9,13,14) are all satisfied. Inequality (10) is satisfied in view of (11). Inequalities (7,11) show that ours is a stable protocol if $\lambda < \lambda_1 = 1/(ks) = 1/32$.

Theorem 3 *In a MAC problem with infinitely many senders, suppose the senders' clocks all agree on the time. Then there is a fixed $\lambda_1 \in (0,1)$ such that for any $\lambda < \lambda_1$, our protocol guarantees an expected waiting time of $O(1)$ for every packet.*

4 The finite case

The model now is the one studied in [9, 18]. There are n senders, with a packet arriving with probability λ_i at sender i at every time step, independently of the other senders; arrivals at different time steps are independent of each other. We assume $\sum_{i=1}^{n} \lambda_i \leq \lambda < 1/32$, as in the infinite case. The further assumption we make is that in addition to synchrony, there is a known bound on the time difference between any pair of sender clocks, *i.e.*, that only the last W bits of the time will have to be agreed upon by the senders, for some known W. Note that once we have this agreement, we can simply run our "infinite senders" protocol; so we focus on this clock agreement problem now. One obvious solution to this is for the senders to communicate with each other to agree on the time. Though this is potentially expensive, this one-shot cost might well be balanced by the good gain in the storage requirements and in the waiting times for all packets, from then on.

Suppose though that such inter-process communication is prohibitively expensive. Then the only means of communication is the shared channel, and we now show how to use it to agree on the time within $O(W2^W + nW\log(nW))$ steps, *w.h.p.*. (We have not attempted to optimize the running time of this protocol.) To this end, the senders will send fake "packets" to the channel; this should not be confused with our actual MAC protocol to be run later on.

The clock agreement protocol would ask all senders to "switch on" when their *local clocks* show some particular time (such as 23:59 EST on April 26, 1995). Let $\ell = a \log n$ for some suitable constant a. Each sender s will independently attempt to use the channel with probability $1/n$ independently at each step, until it succeeds. If s does not succeed within $2^W + \ell$ steps, it will stop attempting to use the channel; else if it does succeed, it will then continuously attempt using the channel, for the next $2 \cdot 2^W + \ell$ steps. Since any pair of senders switch on within 2^W steps of each other, it is clear that at most one sender (the *leader*) is successful.

No leader will be elected only if for the ℓ successive steps beginning 2^W steps after the first sender switched on, either no sender or at least two senders tried using the channel. The probability of this happening is very small—$e^{-\Omega(\ell)}$. Thus we may assume that a leader s_0 was elected. Starting at time step $3 \cdot 2^W + \ell + 1$ since it switched on, s_0 will attempt to make all other senders agree with its local time, in phases $P_1, P_2, \ldots, P_W$. We denote a generic sender that is not s_0, by s henceforth. The sender s will, starting at time step $3 \cdot 2^W + \ell + 1$ since *it* switched on, try to agree with s_0's clock. After phase P_i, all senders will agree with s_0 on the i least significant bits (lsbs) of the time, *w.h.p.*. P_i lasts for $\ell_i = 3 \cdot 2^W + cn\log(nW)$ steps for a suitable constant c; thus, two different senders might differ by at most one in the index of the phase that they think they are in.

Assuming that $P_1, P_2, \ldots, P_i$ have been finished, we describe P_{i+1} now. Let T_{i+1} denote the set of time steps when the clock of s_0 shows a one in the $(i+1)$st lsb. In P_{i+1}, s_0 attempts to use the channel exactly at those time steps that lie in T_{i+1}. Sender s, on the other hand, attempts using the channel independently with probability $1/(3n)$ at each time slot, and infers the the $(i+1)$st lsb by taking the majority result from the time steps (in its version of P_{i+1}) in which it tried (using the channel) and collided. A quick analysis of the correctness of this is as follows, the details will be given in the final version. During the period that was P_{i+1} according to s_0, s would have tried using the channel $\Omega((\ell_i - 2^W)/n)$ times, *w.h.p.*. Since the measure of T_{i+1} during this period is roughly a half and since the expected number of non-leaders that can collide with s at any time step is roughly $1/3$, the majority result chosen by s will be correct, *w.h.p.*. Similarly, the fact that s might have thought that some portions of this period belonged to P_i (or P_{i+2}) has negligible effect, since $\ell_i \gg 2^W$. This protocol takes $O(W2^W + nW\log(nW))$ steps, and hence we get

Theorem 4 *In a MAC problem with n senders, suppose the senders' clocks differ by at most a known number B of steps. Then there is a fixed $\lambda_1 \in (0,1)$ such that for any $\lambda < \lambda_1$, our protocol guarantees, after a setup time of $O(B \log B + n \log B \log(n \log B))$ steps, an expected waiting time of $O(1)$ for every packet.*

5 The effect of the input distribution

Suppose that the distribution of incoming packets to the system has substantially weaker random properties than the independent Poisson distribution (or independent binomial, in the finite case); our protocol will still ensure that the expected waiting time for every packet is $O(1)$. The motivation for studying this is two-fold. First, our contention resolution protocol might be a module in a larger system, with the previous module feeding packets with some possibly very "non-random" distribution. For instance, one of the results of [14] is that for PRAM emulation, memory locations can be hashed in an ℓ-wise independent fashion for some suitably large fixed ℓ rather than in a completely random fashion, to avoid having to store huge hash tables. (Recall that a sequence of random variables $X_1, X_2, \ldots, X_m$ is *ℓ-wise independent* if every ℓ of them are mutually independent; such sequences are well-known to be sampleable using many fewer random bits than do their completely independent counterparts. We will encounter these again, below.) We might be able to guess the packet distribution of their PRAM emulation, for any given PRAM algorithm. The second reason is to show that our protocol does not need crucially the very good large-deviation properties of "well-behaved" distributions like independent Poisson/binomial, to maintain $E[W_{ave}] = O(1)$. In particular for an ℓ-wise independent distribution to be sketched below, direct use of the protocol and analysis of [18] for the finite case, will mandate $E[W_{ave}] = n^{\Omega(1)}$, rather than their $O(\log n)$ bound that holds for independent binomial arrivals. (Of course it is conceivable that a modification of their protocol might do better.) Due to the lack of space, we just sketch the result.

From the paragraph immediately following the statement of Theorem 2, we see that $p_i(t) = O(a^{-i})$ will suffice to maintain the property that $E[W_{ave}] = O(1)$—the strong (doubly exponential) decay of $p_i(t)$ as i increases, is unnecessary. In turn, by analyzing the recurrences presented by Lemmas 4 and 5, we can show that rather than the strong bound of (12), it suffices if

$$H(sb\lambda, \nu_t) \leq \delta \kappa^{-t} \tag{17}$$

for some *constant* κ large enough in comparison with k, and for a sufficiently small constant $\delta > 0$. We can then proceed by induction on i to show that $p_i(t) = O(a^{-i})$ (by showing that $p_i(t) = O(a^{-i}\kappa^{-t})$, which is all we need. Bound (17) can connote a very weak tail behaviour. In particular in the finite senders case, such a bound holds if packets arrive independently at different time steps, but if within each time step, the (at most n) incoming packets have an ℓ-wise independent distribution, for some large enough constant ℓ. It is for this scenario that direct use of the protocol and analysis of [18] will mandate $E[W_{ave}] = n^{\Omega(1)}$.

In fact, such a requirement can be weakened further, to not have independent arrivals at each time slot. We would only need that for any finite sequence of distinct time slots $t_1 < t_2 < \cdots < t_m$, the probability that the total arrival in each of these time slots was more than some value beyond expected, is at most some constant times the corresponding probability, had the arrivals been independent at these time slots with the weak tail distribution of (17). Such situations occur commonly in "negatively correlated" cases. For instance, suppose a total of at most λN packets, for some large N, can arrive into the system, each arriving independently at a time chosen uniformly at random from $[N]$. Note that the arrivals at

different time steps are not independent, but that they do satisfy the above negative correlation property.

Acknowledgement. We wish to thank Michael Kalantar of Cornell University for explaining the practical side of this problem to us. We thank Prabhakar Raghavan and Eli Upfal for sending us an early version of their paper [18], and thank Phil MacKenzie, Greg Plaxton and Rajmohan Rajaraman for allowing us to use part of the LaTeX source of their work [14].

References

[1] N. Abramson. The ALOHA system. In N. Abramson and F. Kuo, editors, *Computer-Communication Networks*. Prentice Hall, Englewood Cliffs, New Jersey, 1973.

[2] D. Aldous. Ultimate instability of exponential backoff protocol for acknowledgement based transmission control of random access communication channels. *IEEE Trans. on Information Theory*, IT-33(2):219–223, 1987.

[3] N. Alon, J. H. Spencer, and P. Erdős. *The Probabilistic Method*. Wiley Interscience Series, John Wiley & Sons, Inc., New York, 1992.

[4] R. J. Anderson and G. L. Miller. Optical communication for pointer based algorithms. Technical Report CRI-88-14, Computer Science Department, University of Southern California, 1988.

[5] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Annals of Mathematical Statistics*, 23:493–509, 1952.

[6] M. Dietzfelbinger and F. Meyer auf der Heide. Simple, efficient shared memory simulations. In *Proc. ACM Symposium on Parallel Algorithms and Architectures*, pages 110–119, 1993.

[7] M. Geréb-Graus and T. Tsantilas. Efficient optical communication in parallel computers. In *Proc. ACM Symposium on Parallel Algorithms and Architectures*, pages 41–48, 1992.

[8] L. A. Goldberg, M. Jerrum, F. T. Leighton, and S. B. Rao. A doubly logarithmic communication algorithm for the completely connected optical communication parallel computer. In *Proc. ACM Symposium on Parallel Algorithms and Architectures*, pages 300–309, 1993.

[9] J. Håstad, F. T. Leighton, and B. Rogoff. Analysis of backoff protocols for multiple access channels. In *Proc. ACM Symposium on Theory of Computing*, pages 241–253, 1987. To appear in *SIAM J. Comput.*

[10] W. Hoeffding. Probability inequalities for sums of bounded random variables. *American Statistical Association Journal*, 58:13–30, 1963.

[11] J. Y. N. Hui and P. A. Humblet. The capacity region of the totally asynchronous multiple-access channel. *IEEE Trans. on Information Theory*, IT-31:207–216, 1985.

[12] *IEEE Trans. on Information Theory*, IT-31, 1985.

[13] F. P. Kelly. Stochastic models of computer communication systems. *J. Royal Statistical Society (B)*, 47:379–395, 1985.

[14] P. D. MacKenzie, C. G. Plaxton, and R. Rajaraman. On contention resolution protocols and associated probabilistic phenomena. In *Proc. ACM Symposium on Theory of Computing*, pages 153–162, 1994.

[15] R. Metcalf and D. Boggs. Ethernet: distributed packet switching for local computer networks. *Communications of the ACM*, 19:395–404, 1976.

[16] V. A. Mikhailov and T. S. Tsybakov. Upper bound for the capacity of a random multiple access system. *Problemy Peredachi Informatsii*, 17:90–95, 1981. Also presented at the IEEE Information Theory Symposium, 1981.

[17] N. Pippenger. Bounds on the performance of protocols for a multiple access broadcast channel. *IEEE Trans. on Information Theory*, IT-27:145–151, 1981.

[18] P. Raghavan and E. Upfal. Stochastic contention resolution with short delays. In *Proc. ACM Symposium on Theory of Computing*, 1995. To appear.

[19] R. Raman. The power of Collision: Randomized parallel algorithms for chaining and integer sorting. In *Proceedings, 10th Annual FST & TCS Conference*, Lecture Notes in Computer Science # 472, pages 161–175. Springer-Verlag, Berlin, December 1990. Also available as University of Rochester CS Dept. TR 336, March 1990 (Revised January 1991).

[20] L. G. Valiant. General purpose parallel architectures. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume A*, pages 943–971. Elsevier, New York, 1990.

[21] N. D. Vvedenskaya and M. S. Pinsker. Non-optimality of the part-and-try algorithm. In *Abstracts of the International Workshop on Convolutional Codes, Multiuser Communication, Sochi, USSR*, pages 141–148, 1983.

Appendix

Proof of Lemma 1 Let $a_i = Pr(Y_i = 1)$. This proof is essentially the same as the one of Raman [19]. However since the scenario of [19] is a little different, with a_i being the same for all i, we present this proof here.

The proof is by induction on ℓ; the base case of $\ell = 1$ is immediate. Assume the lemma for $\ell - 1$. For $i \in [\ell]$, let $X^{(i)} \doteq \sum_{j\in[i]} X_j$ and $Y^{(i)} \doteq \sum_{j\in[i]} Y_j$. We may assume that c is an integer, since the X_i and Y_i are integral. If $c = 0$, then $Pr(X^{(\ell)} \geq c) = Pr(Y^{(\ell)} \geq c) = 1$. For $c \geq 1$,

$$\begin{aligned}
Pr(X^{(\ell)} \geq c) &= Pr(X^{(\ell-1)} \geq c) + Pr(X^{(\ell-1)} = c-1)Pr(X_\ell = 1 | X^{(\ell-1)} = c-1) \\
&\leq Pr(X^{(\ell-1)} \geq c) + a_\ell Pr(X^{(\ell-1)} = c-1) \text{ (by (a))} \\
&= a_\ell Pr(X^{(\ell-1)} \geq c-1) + (1-a_\ell)Pr(X^{(\ell-1)} \geq c) \\
&\leq a_\ell Pr(Y^{(\ell-1)} \geq c-1) + (1-a_\ell)Pr(Y^{(\ell-1)} \geq c) \text{ (by induction hypothesis)} \\
&= Pr(Y^{(\ell)} \geq c).
\end{aligned}$$

□

Proof of Lemma 2 Part (i) is immediate. For part (ii), number the balls in each U_i as $1, 2, \ldots$ arbitrarily and let $X_{i,j}$ be the indicator random variable for the jth ball in U_i colliding with a lower-numbered ball from U_i. Thus, $C \leq 2X$, where $X \doteq \sum_{i,j} X_{i,j}$. Note that, since the balls are thrown independently, the conditional probability that $X_{i,j} = 1$ is at most $(j-1)/\ell$, even given the bins in U_i occupied by all the balls but the jth. Thus by Lemma 1, X is stochastically dominated by a sum Y of independent $\{0,1\}$ random variables, where

$$E[Y] \leq z \sum_{j\in[\ell]} (j-1)/\ell \leq zs^2/(2\ell).$$

Thus Lemma 1, combined with the fact that $C \leq 2X$, concludes the proof. □

Proof of Lemma 3 Part (i) is immediate. For part (ii), note that

$$\begin{aligned}
e_i &= Pr(E_i) = Pr(E_i|E_{i-1})Pr(E_{i-1}) = e_{i-1}Pr(E_i|E_{i-1}) \\
&= e_{i-1}(Pr(E_i | u_{i-1}(P) \text{ was good} \wedge E_{i-1})Pr(u_{i-1}(P) \text{ was good} | E_{i-1}) +
\end{aligned}$$

$$
\begin{aligned}
& Pr(E_i|u_{i-1}(P) \text{ was bad} \wedge E_{i-1})Pr(u_{i-1}(P) \text{ was bad}|E_{i-1})) \\
\leq\ & e_{i-1}(Pr(E_i|u_{i-1}(P) \text{ was good} \wedge E_{i-1}) + Pr(u_{i-1}(P) \text{ was bad}|E_{i-1})) \\
\leq\ & e_{i-1}(Pr(E_i|u_{i-1}(P) \text{ was good} \wedge E_{i-1}) + Pr(u_{i-1}(P) \text{ was bad})/Pr(E_{i-1})) \\
\leq\ & e_{i-1}q + Pr(u_{i-1}(P) \text{ was bad}) = qe_{i-1} + p_{i-1}(1).
\end{aligned}
$$

□

Proof of Lemma 4 Suppose that u_i is t-bad but that neither (i) or (ii) holds. Then, u_i has at most 1 child v that is either t-loaded or $(t-1)$-loaded, and none of the other children of is $(t-1)$-bad. Node v can contribute a load of at most $br^{i-1}d^t/a$ packets to u_i; the other children contribute a total load of at most $(k-1)br^{i-1}d^{t-2}/a$. Thus the children of u_i contribute a total load of at most $br^i d^{t-2}(d^2+k-1)/a$, which contradicts the fact that u_i is t-bad if (9) holds. □

Proof of Lemma 5 Suppose that u_i is t-bad. There are the possibilities that at least one child of u_i is 2-bad or that at least two children are 1-bad. If neither of these conditions holds, then either (A) u_i has exactly one child which is 1-loaded with no other child being bad, or (B) all children are good. To apply Theorem 1 and Fact 1, we require that the corresponding values of ν exceed 1. The hypothesis of the lemma assures $a(r-d)/(k-1) > 1$, and, since $r < kd$, $ar/k > a(r-d)/(k-1) > 1$.

In case (A), the $k-1$ good children contribute a total of at least

$$cap(u_i)/a - cap(u_{i-1})d/a = \frac{br^i(r-d)}{ar}.$$

In the notation of Lemma 2, $z = k-1$, $s = br^{i-1}/a$, and $\ell = br^{i-1}$. Thus by Lemma 2, the probability of occurrence of case (A) is at most

$$kp_{i-1}(1)H\left(\frac{(k-1)br^{i-1}}{2a^2}, \frac{a(r-d)}{k-1}\right)$$

packets to u_i. In case (B), the k good children contribute at least $cap(u_i)/a = br^i/a$. By a similar argument, the probability of occurrence of case (B) is at most

$$H\left(\frac{kbr^{i-1}}{2a^2}, \frac{ar}{k}\right).$$

The inequality in the lemma follows. □